

Spring 1-1-2017

Fast Band-Structure Computation for Phononic and Electronic Waves in Crystals

Dimitri Krattiger

University of Colorado at Boulder, dimitri.krattiger@gmail.com

Follow this and additional works at: https://scholar.colorado.edu/asen_gradetds

 Part of the [Acoustics, Dynamics, and Controls Commons](#), [Aerospace Engineering Commons](#),
and the [Physics Commons](#)

Recommended Citation

Krattiger, Dimitri, "Fast Band-Structure Computation for Phononic and Electronic Waves in Crystals" (2017). *Aerospace Engineering Sciences Graduate Theses & Dissertations*. 173.

https://scholar.colorado.edu/asen_gradetds/173

This Dissertation is brought to you for free and open access by Aerospace Engineering Sciences at CU Scholar. It has been accepted for inclusion in Aerospace Engineering Sciences Graduate Theses & Dissertations by an authorized administrator of CU Scholar. For more information, please contact cuscholaradmin@colorado.edu.

**Fast Band-Structure Computation for Phononic and
Electronic Waves in Crystals**

by

Dimitri Krattiger

B.S., University of Wisconsin Madison, 2010

M.S., University of Wisconsin Madison, 2012

A thesis submitted to the
Faculty of the Graduate School of the
University of Colorado in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
Department of Aerospace Engineering Sciences
2017

This thesis entitled:
Fast Band-Structure Computation for Phononic and Electronic Waves in Crystals
written by Dimitri Krattiger
has been approved for the Department of Aerospace Engineering Sciences

Professor Mahmoud I. Hussein (Chair)

Professor John A. Evans

Professor Carlos A. Felippa

Professor Matthew A. Glaser

Professor Per-Gunnar J. Martinsson

Professor Kurt K. Maute

Professor Alireza Doostan

Date _____

The final copy of this research proposal has been examined by the signatories, and we find that both the content and the form meet acceptable presentation standards of scholarly work in the above mentioned discipline.

Krattiger, Dimitri (Ph.D., Aerospace Engineering)

Fast Band-Structure Computation for Phononic and Electronic Waves in Crystals

Thesis directed by Professor Mahmoud I. Hussein

The band structure is a frequency/energy versus wave vector/momentum relationship that fundamentally describes the nature of wave motion in a periodic medium. It is immensely valuable for predicting and understanding the properties of electronic, photonic, and phononic materials, and is typically computed numerically. For materials with large unit cells, such as nanostructured supercells for example, band-structure computation is very costly. This inhibits the ability to feasibly analyze new material systems with potentially extraordinary properties. This thesis describes a novel unit-cell model-reduction technique for band-structure calculations that is capable of lowering computational costs by one or two orders of magnitude with practically insignificant loss of accuracy.

This new methodology, termed *Bloch mode synthesis*, is based on unit-cell modal analysis. It begins from a free-boundary unit-cell model. Before periodic boundary conditions are applied, this free unit cell behaves as though it has been cut out from its periodic surroundings. A truncated set of normal mode shapes is then used to compactly represent the interior portion of the unit cell while retaining nearly all of the dynamically important information. A Ritz basis for the unit cell is formed by combining the interior modes with a second set of modes that preserves the flexibility needed to enforce a Bloch wave solution in the unit cell. Residual mode enhancement and interface modal reduction improve performance further. With this highly reduced model, Bloch boundary conditions corresponding to waves of any directions and wavelength can be applied to very quickly obtain the band structure.

Bloch mode synthesis is derived in the context of elastic wave propagation in phononic crystals and metamaterials, but the framework is also well suited for other types of waves. It shows particular promise in speeding up electronic structure calculations – a central problem in computa-

tional materials science that lies at the heart of property determination for numerous applications including semiconductors, superconductors, photovoltaics, thermoelectrics, lasers, and light emitting diodes. It also shows promise for predicting thermal properties of nanophononic materials. Thermal conductivity calculations require the full-spectrum band structure, which are obtained by modifying the Bloch mode synthesis formulation to incorporate high-frequency information in the basis.

Acknowledgements

This research project has been among the most challenging and rewarding experiences of my life. There are so many people that contributed without whom this project could never have succeeded. I would like to thank my fiancée for her unwavering moral support and for always being quick to compliment my graphs. Also, my parents and sisters who always knew I could do it and helped me believe I could too. I am grateful to my roommates whose riddles and boardgames often provided a welcome distraction, as well as the rest of my friends in Boulder and Madison for making my graduate school experience so enjoyable. I would also like to thank my research group colleagues, both past and present, as well as the many visiting scholars for the interesting discussions, for teaching me new things, and in many cases providing purpose for my research.

I gratefully acknowledge the financial support for my work from the NSF, as well as fellowships from the CU Boulder graduate school, and for funding through fellowships and teaching positions from the Aerospace Engineering Sciences Department at CU Boulder.

Finally I would like to thank my thesis committee for their time, their advise, and for not being afraid to challenge me. I am particularly indebted to my thesis advisor, Dr. Mahmoud Hussein, for helping me navigate the challenges of academic publishing, for supporting me in pursuing research tangents, for putting up and even welcoming my stubbornness, and for pushing me to get the most out of my Ph.D. experience. Thank you!

Contents

Chapter	
1	Introduction 1
1.1	Periodic Materials 1
1.2	Analysis of Periodic Materials 2
1.2.1	Modeling Periodic Materials 3
1.2.2	Improving Computational Efficiency of Band-Structure Calculations 5
1.3	Summary of Chapters and Contributions 8
2	Fundamentals of Wave Propagation in Periodic Media 10
2.1	Introduction 10
2.2	Band Structure of a 1D Spring-Mass Chain 10
2.3	Lattice Symmetry and Brillouin Zones 12
2.4	Bloch's Theorem 16
2.4.1	Bloch Operator Versus Bloch Boundary Conditions 17
2.4.1.1	1D Spring-Mass Example 17
2.4.1.2	Bloch Boundary Condition Formulation for Finite-Element Models 19
2.4.1.3	Bloch Operator Formulation for Finite-Element Models 23
2.5	Band-Structure Calculations 25
2.5.1	Frequency-as-a-Function-of-Wave-Vector Solution: $\omega(\mathbf{k})$ Method 25
2.5.1.1	Comparison of Bloch Boundary Conditions and Bloch Operator Accuracy for 2D Plane-Strain Finite-Element Models 26

2.5.2	Wave-Vector-as-a-Function-of-Frequency Solution: $\mathbf{k}(\omega)$ Method	28
2.5.2.1	$\mathbf{k}(\omega)$ Solutions Using the Bloch Boundary Condition Formulation	30
2.5.2.2	$\mathbf{k}(\omega)$ Solution Using the Bloch Operator Formulation	33
2.6	Damping in Band-Structure Calculations	34
2.6.0.1	Damped $\omega(\mathbf{k})$ Band-Structure Calculations	36
2.6.0.2	Damped Band-Structure Numerical Example	36
2.7	Computational Complexity of Algorithms	37
2.8	Summary	41
3	Bloch Mode Synthesis: A Modal Reduction method for Band-Structure Calculations	43
3.1	Introduction	43
3.2	Formulation of Interior Reduction	47
3.2.1	Hurty/Craig-Bampton Substructure Representation	47
3.2.2	Approximating Residual-Mode Contribution for Enhanced Accuracy	48
3.2.3	Automated Multi Level Substructuring	52
3.3	Numerical Examples	55
3.3.1	Band-Structure Calculation using $\omega(\mathbf{k})$ Method	56
3.3.2	Discretization Error	59
3.3.3	Performance Comparison: Automated Multi-Level Substructuring Versus Hurty/Craig-Bampton Substructuring	60
3.4	Summary	62
4	Extension of Bloch Mode Synthesis to Incorporate Unit-Cell Boundary Reduction	64
4.1	Introduction	64
4.2	System-Level Characteristic Constraint Mode Reduction	68
4.3	Local-Level Characteristic Constraint Mode Reduction	68
4.4	Numerical Experiments	73
4.4.1	Eight-Petal Inclusion Model	75

4.4.1.1	Band-Structure Calculation using $\omega(\mathbf{k})$ Method	75
4.4.1.2	Complex Band-Structure Calculation using $\mathbf{k}(\omega)$ Method	75
4.4.1.3	Performance Evaluation	78
4.4.2	Acoustic Metamaterial Model	78
4.4.2.1	Band-Structure Calculation using $\omega(\mathbf{k})$ Method	80
4.4.2.2	Performance Evaluation	82
4.5	Summary	82
5	Application of Bloch Mode Synthesis to Electronic Band-Structure Problems	84
5.1	Introduction	84
5.2	Electronic Structure Theory	86
5.2.1	Density Functional Theory	87
5.2.2	Bloch's Theorem	89
5.2.3	Brillouin Zone Integration	90
5.2.4	Finite Element Discretization	94
5.2.5	Bloch Mode Synthesis	95
5.3	Numerical Examples	96
5.4	Summary	102
6	Extension of Bloch Mode Synthesis to Full-Spectrum Phonon Calculations	105
6.1	Introduction	105
6.2	Fast System Assembly for Empirical Interatomic Potentials	106
6.2.1	Quasi-Free Unit Cell	107
6.2.2	Building Block Assembly Approach	108
6.3	Full-Spectrum Eigenvalue Solution by Spectrum Slicing	110
6.4	Windowed Bloch Mode Synthesis for Reduced-Order Spectrum Slicing	114
6.4.1	Numerical Example: Silicon Nanopillared Membrane Full-Spectrum Band Structure	118

6.5 Summary	120
7 Conclusion	122
7.1 Summary	122
7.2 Future Work	124
Bibliography	127
Appendix A. Acronyms	133

Tables

Table

4.1	Abbreviations for different generalized BMS configurations	74
4.2	Comparison of full and BMS reduced-model calculation time and model-size for 8-lobed inclusion material band-structure calculations.	76
4.3	Timing and model-size details for acoustic metamaterial.	79
5.1	Reference and base finite element (FE) model parameters	101
6.1	Scaling analysis results for full-spectrum eigenvalue calculation using spectrum slicing algorithm and direct algorithm.	114

Figures

Figure

1.1	2D periodic material with unit cell highlighted.	1
2.1	1D periodic spring-mass chain with unit cell highlighted.	10
2.2	Band-structure diagram for 1D periodic mass-spring chain. Different BZs are shown in various shades of grey.	11
2.3	Lattice diagrams (left) and corresponding BZ diagrams (right) are shown for oblique (a), rectangular (b), centered-rectangular (c), hexagonal (d), and square (e) Bravais lattices.	14
2.4	Schematics showing a square unit cell with a circular inclusion and corresponding symmetry planes (left), the corresponding BZ and IBZ with high symmetry points labeled (middle), and the corresponding band-structure diagram showing frequencies calculated along a path traced along the edge of the IBZ.	15
2.5	Square unit cell with an elliptical inclusion and corresponding symmetry planes (left), the corresponding BZ and IBZ with high symmetry points labeled (middle), and the corresponding band-structure diagram showing frequencies calculated along a path traced along the edge of the IBZ.	16
2.6	Unit cell for 1D periodic spring-mass chain with 4 DOFs per unit cell	17
2.7	Isolated unit-cell FE mesh for general periodic 2D material with boundary sets labeled. The isolated unit cell is obtained by cutting the unit cell out of the periodic material so that all boundaries are free.	21

2.8	Comparison of Bloch operator and Bloch BC frequency errors versus model size. The maximum frequency error is obtained by comparing the lowest 10 band-structure branches of the 2D circular inclusion unit cell (shown in the inset) to a highly refined mesh and taking the maximum relative frequency error over the BZ.	27
2.9	Comparison of Bloch operator and Bloch BC frequency errors versus model size. The maximum frequency error is obtained by comparing the lowest 10 band-structure branches of the 2D square inclusion unit cell (shown in the inset) to a highly refined mesh and taking the maximum relative frequency error over the BZ.	27
2.10	3D representation of the $\mathbf{k}(\omega)$ band structure for the Γ - X direction of the eight-lobe inclusion unit cell (shown in inset). The purely-real wave-number solutions, purely-imaginary wave-number solutions, and complex wave-number solutions are shown in red, green, and blue respectively.	29
2.11	Conventional representation of the $\mathbf{k}(\omega)$ band structure for the Γ - X direction of the eight-lobe inclusion unit cell (shown in inset). Real wave number solutions are shown on the right (positive x-axis) and imaginary wave number solutions are on the left (negative x-axis).	29
2.12	Plate lattice unit cell meshed with Mindlin elements.	37
2.13	Plate lattice damped-frequency band structure (top left) and damping-ratio curves (bottom left) for free waves, and complex-wave vector band structure (top right) for driven waves. The frequencies are normalized as follows: $\Omega = \omega_d L_x \sqrt{\rho/E}$	38

2.14	Computational complexity of iterative and direct eigenvalue solvers. The figure shows computation time versus matrix size for direct eigenvalue solution using MATLAB's <code>eig</code> function, and for iterative eigenvalue solution using MATLAB's <code>eigs</code> function. The system solution is performed for the hermitian system of equations shown in Eq. (3.9), where the matrices are obtained from an FE discretization of the circular-inclusion unit cell shown in the inset. The system size is varied by using increasingly refined FE models. The number of eigenvalue solutions, k , obtained from the iterative solver is varied to produce the different curves. The iterative solver takes advantage of the sparsity of the FE matrices.	40
2.15	Computational complexity of system solver. The figure shows computation time versus matrix size for direct system solution using MATLAB's backslash operator. The system solution is performed for a hermitian matrix with multiple right-hand side vectors. The system solution being performed is shown in Eq. (3.8), where the matrices are obtained from an FE discretization of the circular-inclusion unit cell shown in the inset. The system size is varied by using increasingly refined FE models. The number of right-hand-side vectors, p , is varied from 1, to the total number of interface DOFs, n_b	42
3.1	Flowchart illustrating the steps involved in using BMS to compute the band structure for a discretized unit-cell model	46
3.2	Tree diagram showing the partitioning of a free unit cell for AMLS analysis	54
3.3	Compact representation of multi-level substructure tree	55
3.4	2D unit-cell meshes with eight-lobed inclusion (resembling the Helicoid-Catenoid logo used for the Phononics 20XX conference series).	57
3.5	(a) Dispersion diagram comparison for eight-lobed inclusion material where $\Omega = \omega L_x \sqrt{\rho/E}$ is the non-dimensional frequency, and (b) selected mode shape comparison for the mode highlighted in the dispersion diagram.	58

3.6	Discretization error and BMS reduced-model error for small, medium, and large meshes with respect to the reference mesh. The BMS models are formed using the HCB, and HCB+ reductions of the interior, and no reduction of the interface.	59
3.7	Eight-lobed inclusion model substructure partitions (left) and AMLS maximum frequency error versus computational time fraction (right) for small (top), medium (middle), and large (bottom) meshes from Fig. 3.4.	61
4.1	Schematic of the different interface reduction methods and their classification as system-, local-, or hybrid-level. The thumbnail illustrations show whether interface modes are coupled (single color) or uncoupled (different colors), the level of substructure assembly when the interface modes are computed, and for the local methods, whether modes match at the interface (indicating exact compatibility) or not (indicating weak compatibility).	66
4.2	Flowchart illustrating the addition of a local interface reduction step to BMS. Compared to Fig. 3.1, the new box (step 4) is darkened. Note that the application of Bloch BCs is slightly modified to apply to boundary modes rather than to physical boundary DOFs.	67
4.3	(a) Full boundary mode, (b) partitioned boundary mode, and compatible boundary modes for (c) left and (d) right boundaries. Although boundary modes only contain boundary DOFs, they define a linear combination of constraint modes which can be used to compute deflection in the unit-cell interior. The modes shown here include this interior deflection in light gray.	71
4.4	(a) Dispersion-diagram comparison for 8-lobed inclusion material where $\Omega = \omega L_x \sqrt{\rho/E}$ is the non-dimensional frequency, and (b) selected mode-shape comparison for the mode highlighted in the dispersion diagram.	77
4.5	Complex dispersion-diagram comparison for 8-lobed inclusion material where $\Omega = \omega L_x \sqrt{\rho/E}$ is the non-dimensional frequency.	77

4.6	Performance plot of computation time fraction versus frequency error, generated by computing $\omega(\mathbf{k})$ band structures with the large mesh of the 8-lobe inclusion material.	79
4.7	FE mesh for acoustic metamaterial unit cell consisting of a soft rubber (white) and heavy lead (dark grey) resonator embedded in an epoxy (light grey) matrix.	80
4.8	(a) Dispersion-diagram comparison for acoustic metamaterial, and (b) selected mode shape computed with BMS.	81
4.9	Performance plot of computation time fraction versus frequency error, generated by computing $\omega(\mathbf{k})$ band structures for the resonator model.	81
5.1	Flowchart depicting self-consistent field iteration	89
5.2	MP grids for a square lattice (top), and for a hexagonal lattice (bottom). Grids are shown with no shift (left) and with a half-grid point shift (right). For each diagram, the IBZ is highlighted, and the symmetry reduced set of grid points is shown in red with the corresponding weight multiplier.	92
5.3	Two \mathbf{k} points related by reflection about the y-axis (left) and examples of corresponding electron densities (center and right) also exhibiting a reflection about the y-axis.	93
5.4	Silicon crystalline unit cells. The subfigure on the left (a) shows the 8-atom conventional cell and the subfigure on the right (b) shows the 2-atom primitive cell	97
5.5	Empirical pseudopotential isocontours for silicon primitive cell	97
5.6	Error in band energy with respect to MP rule for 13,824 DOF meshes consisting of linear, quadratic, and quartic elements.	99
5.7	Error in band energy with respect to computation time for various calculation cases	100
5.8	Error in band energy with respect to computation time for various BMS model reductions of the base cases in Table 5.1.	102
5.9	Band-energy diagram obtained with full FE model and with BMS reduced-order model. The band energy is obtained with 65 \mathbf{k} points	103

5.10	Comparison of charge densities between full model (top row) and BMS model (bottom row). Each charge density is computed for a single band-structure solution. The charge densities correspond to points A (left column), B (middle column), and C (right column) in Fig. 5.9.	104
6.1	Silicon thin film unit cell viewed orthogonal to the thin film plane with a layer of phantom atoms added. Phantom atoms are shown in transparent colors corresponding to their periodic counterparts within the unit cell.	108
6.2	Silicon nano-pillared thin film unit cell.	109
6.3	2D “plus” shaped spring-mass unit cell used to investigate spectrum slicing algorithm	112
6.4	Illustration of spectrum slicing algorithm showing consecutive windows being computed	113
6.5	Timing results for full-spectrum eigenvalue calculations on increasing model sizes. The computational complexity is observed to be approximately $\mathcal{O}(n^3)$ for direct eigenvalue solution, and $\mathcal{O}(n^2)$ for the spectrum slicing algorithm.	115
6.6	Full-spectrum band-structure diagram for silicon nanopillared membrane (left) and corresponding density of states, $\text{DOS} = n_{bin}/(n_{\mathbf{k}} \times \Delta\omega_{bin})$. Results for both full-direct calculation and spectrum-sliced BMS calculation are shown. The BMS band structure solutions are shown as colored dots with different colors illustrating different frequency windows. The insets show more detail for select frequency ranges. Circled points indicate spurious BMS solutions	119

Chapter 1

Introduction

1.1 Periodic Materials

Periodic materials have long been studied due to their ability to precisely influence the waves that travel through them. A periodic material can be defined as an array of unit cells extending infinitely in one, two, or three spatial dimensions. Of course, practically no infinite materials exist, but much insight into the behavior of a finite system can be gained by considering its infinite counterpart. This is especially true when the underlying dynamics take place at length scales much smaller than the finite system size. Figure 1.1 shows an example of a 2D periodic material with a single unit cell highlighted. The unit cell is the building block that makes up the material. Thus, a comprehensive understanding of the material can be obtained by studying the unit cell.

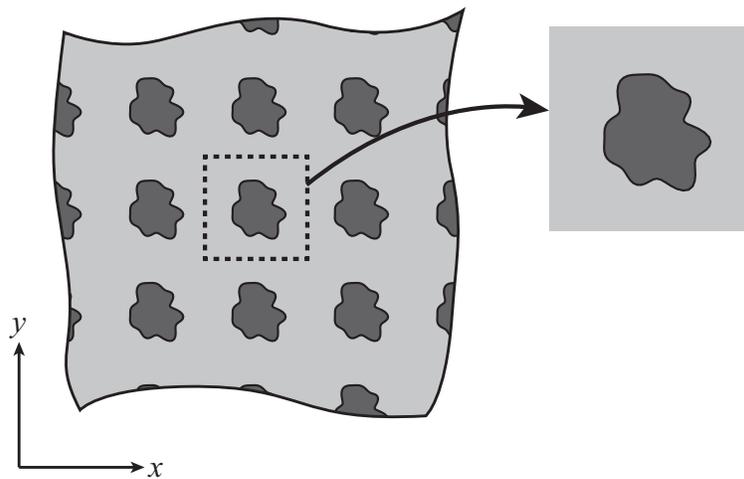


Figure 1.1: 2D periodic material with unit cell highlighted.

Periodic materials can be both naturally occurring and man-made. The most notable examples of naturally occurring periodic materials are crystals, where there is periodicity in the arrangement of the atoms, and the unit cell is only a few nanometers across. This periodicity affects the crystal's electrical and thermal conductivity, as well as optical properties. Man-made periodic materials often arise because strength and weight constraints encourage the use of lattice structures. One example of man-made periodic materials is in the field of aeronautics, where the internal structure of airplane wings makes use of a periodic arrangement of ribs and spars, and the skin can be made of composite materials that exhibit periodicity on a much smaller scale.

1.2 Analysis of Periodic Materials

The abundance of periodic materials and structures in our environment has led to the development of theories that describe how these materials behave. Felix Bloch formalized a theory to describe waves in crystalline solids so that he could solve Schrödinger's equation and predict how electrons behave in crystals [1]. The same theory is applied to the governing equations of elasticity to understand how mechanical waves behave in a periodic material. The term phonon formally describes the particle-like quantum packets of vibrational energy that exist in crystal lattices, but has since been adopted by the broader community studying elastic waves in all periodic media. Thus the term phononic crystal applies to periodic materials of all scales that admit mechanical waves [2]. Bloch's theorem is also applied to the governing equations of electromagnetism to understand how light behaves in periodic materials.

The behavior of waves can be compactly described by the band structure. This diagram provides a relationship between the wave vector and the frequency of the wave, or equivalently, the momentum and energy of the wave. This simple yet powerful relationship is crucial to predicting many material properties. It also elucidates important features like band gaps — frequency or energy levels where no waves can exist [2]. In electronic materials, the presence of a band gap affects whether a material will behave like a conductor, a semiconductor or an insulator [3]. The electronic band gap also dictates whether a material will emit light as excited electrons move down

from higher energy levels, and if so, at what wavelength. In acoustic materials, band gaps represent ranges of sound frequencies that are not allowed to propagate through the medium.

Band gaps may form due to the material periodicity. Waves reflect off of boundaries and interfaces between constituent materials and constructively and destructively interfere. Depending on the interference, the wave may or may not propagate in space. Scattering due to the periodicity of the medium is often referred to as Bragg scattering, or Bragg reflection [3, 4]. This reflection may occur in such a way that the waves decay in space. If this is the case, then the band gap is referred to as a Bragg-scattering induced band gap. Bragg band gaps occur in frequency ranges corresponding to wavelengths that are on the order of the unit-cell size. Wavelengths that are much longer than the unit cell effectively behave as though they are traveling through a homogeneous medium. It is possible to make use of a different phenomenon to affect waves that are much longer than the unit cell. Metamaterials make use of some sort of resonators within a material in order to affect wave propagation. Most notably, these resonators can open band gaps in frequencies well below the Bragg range. This can produce material properties outside of normal ranges such as negative mass or stiffness (in a dynamic sense), and negative refraction. For this reason they are called metamaterials. Although these metamaterials do not rely on periodicity, they are often modeled as periodic to simplify analysis and to gain insight. One of the earliest studies of resonant inclusions in a periodic material was made by Kelvin who modeled optical dispersion in a crystal using a series of Lorentz oscillators. More recently, the concept of metamaterials has been extended to elastic media [5].

1.2.1 Modeling Periodic Materials

Theoretical models for periodic materials are based on the assumption that the periodicity extends infinitely in space. With the help of Bloch's theorem, an infinite material can be exactly modeled by considering just a single unit cell. After modeling the unit cell and enforcing Bloch's theorem, a time harmonic displacement field is assumed to yield the characteristic equation which relates the wave vector and the frequency. For discrete elastic structures like spring-mass systems

or vibrating atomic crystal lattices, the unit cell is modeled by considering the potential energy due to the interactions between the masses/atoms. For spring-mass systems, this potential is commonly described using Hooke's law [6]. For atomic crystal lattices, there are several types of potentials that are obtained empirically [7, 8] or numerically. For unit cells that can be treated with continuum mechanics, the solution field can be approximated over the unit-cell domain using a numerical discretization. The discretization allows the governing partial differential equation to be approximated by a system of ordinary differential equations (ODEs).

Numerous discretization methods exist and have been adapted to band-structure calculation problems. Three of the most popular discretizations for periodic materials are plane-wave (PW) expansion, finite difference (FD), and FE methods. The PW expansion method represents the solution as a superposition of PWs and is very popular due to its simplicity. It struggles with sharp gradients, however, because of the global nature of the PW basis functions. The FD method discretizes the domain into a regular grid of points and expresses the strong form of the governing equations at each point in terms of its neighboring points. The solution at each point in the discretization is very localized so sharp gradients are handled easily and, as a further benefit, the resulting system of equations is very sparse. The FD method struggles, however, with complex geometries and irregular grids, and does not converge as well as methods that make use of basis functions. The FE method makes use of basis functions with very localized support. This, combined with the ability to locally refine the element mesh, allows the FE method to handle sharp gradients in either the solution field or the material distribution. This allows the FE method to produce accurate solutions with relatively few elements. Furthermore the resulting system matrices are sparse allowing for lowered memory and computational costs. For these reasons, this thesis will focus on models obtained with the FE method.

Once the unit cell has been modeled, the band structure is obtained by either looping through all of the wave vectors of interest and solving each resulting eigenvalue problem for the frequencies [$\omega(\mathbf{k})$ method], or by looping through all of the frequencies of interest and solving each resulting eigenvalue problem for the corresponding wave vectors [$\mathbf{k}(\omega)$ method]. Most of the first chapter is

devoted to fundamental topics in the analysis of periodic materials, including modeling the unit cell, application of Bloch's theorem, and computing the band structure.

Depending on the complexity/size of the material unit cell, and the level of accuracy required, the system of equations can be very large. Many algorithms exist for accurately and efficiently obtaining eigen solutions to systems of equations, but even so, the computations can become very time consuming for large systems. This is exacerbated by the necessity to loop through and solve at multiple frequencies or wave vectors to obtain the band structure. In optimization problems, this computational cost is multiplied even further by the number of designs to be evaluated. Perhaps the most computationally demanding band-structure problems occur at the atomic scale. Periodic nanostructures may result in unit cells on the order of several hundred nanometers. Additionally, to understand certain phenomena it may be necessary to consider unit cells as large as several microns. To put this into perspective, a $100\text{nm} \times 100\text{nm} \times 100\text{nm}$ unit cell composed of crystalline silicon already contains around 50 million atoms. Performing a lattice dynamics computation (i.e., computing the phonon band structure) would require 3 degrees of freedom (DOFs) per atom. Requirements for electronic-structure calculations are even more stringent because at least one (and possibly several) DOFs are needed per electron, and the electron count is often much higher than the atom count. This requirement can be relaxed somewhat by approximations that consider just the valence electrons, but even so the computational cost is very high. The self-consistent nature of electronic structure problems further elevate the computational cost because several iterations of the band-structure calculation must be performed in order to first obtain a converged electronic potential [9].

1.2.2 Improving Computational Efficiency of Band-Structure Calculations

Due to the high computational cost, there have been several research efforts dedicated to more efficient band-structure calculation. Axemann and Kuchment noted that direct eigenvalue solution is very costly and does not take advantage of the sparsity of the system matrices [10]. They proposed a simultaneous coordinate over-relaxation (SICOR) [11] approach to obtain the extreme

valued eigenvalues iteratively, so that the sparsity of the system is exploited. Although SICOR has not seen widespread use, other eigenvalue algorithms that use an iterative approach and take advantage of the system sparsity have gained considerable traction [12]. Perhaps the most popular of these iterative eigenvalue solvers is the implicitly restarted Arnoldi method (implemented as `eigs` in Matlab). Dobson developed a Fourier preconditioner that takes advantage of the material periodicity, and can speed up convergence of iterative solvers [13]. He also put forward the idea of using eigenvectors obtained at a previous \mathbf{k} -point as a guess for iterative solution at the next \mathbf{k} -point, thus further improving convergence of the iterative solver. Chern *et al* showed that a multigrid solution method embedded in an inverse eigenvalue solver can speed up convergence of iterative eigenvalue solvers [14].

The previously discussed methods are primarily focused on speeding up convergence of iterative solvers. Another approach to speeding up band-structure calculation is model-order reduction whereby the system of equations is reduced in size so it may be solved more quickly. Depending on the reduction, the solution may be exact or approximate. For example, Guyan reduction [15, 16], also known as static condensation, splits the problem into slave and master DOFs and assumes zero inertial contribution from the slave DOFs. This allows the entire displacement field to be expressed in terms of just the master DOFs, thus reducing the problem size. This reduction is exact for static problems (i.e $\omega = 0$), but is increasingly inaccurate the higher the frequency. The dynamic reduction works very similarly, but does not assume negligible inertia. Rather, it performs the reduction about some test frequency. The reduced model is exact for that test frequency and decreases in accuracy as one moves away from that frequency. A large frequency range can then be accurately spanned by selecting multiple test frequencies. This requires multiple reduced models to be formed, which is in itself a computationally intensive process. The dynamic reduction is very common when solving the $\omega(\mathbf{k})$ problem because the frequency must be specified before solving each eigenvalue problem anyway.

Modal reductions are another common form of model reduction that can be adapted from structural mechanics and applied to periodic materials. The mode shapes of interest for the full

model can be computed at a few key points throughout the Brillouin zone (BZ). The mode shapes are then collected and used as a reduced basis for other points in the BZ [17]. The key points may be selected using high symmetry points, yielding the reduced Bloch mode expansion (RBME) [18], or can be obtained using an adaptive process [19]. This dissertation proposes another form of modal reduction for the unit-cell problem, namely the extension of component mode synthesis (CMS) techniques to band-structure problems. The main idea in CMS is to split a system (for example, an aircraft model) into a set of substructures (fuselage, wings, tail, etc.), describe each substructure with a selected set of Ritz mode shapes, and then couple the mode shapes of the various substructures together in order to form a reduced-order model that can closely approximate the overall system vibrational response. The reduced system modes are thus efficiently synthesized from the component modes. The extension of CMS techniques to band-structure problems is compactly referred to as Bloch mode synthesis (BMS). This approach decomposes the domain into interior and boundary DOFs, and then describes the interior with a truncated modal basis while leaving the boundary DOFs in their original “real space” form. This allows the boundary conditions (BCs) used to enforce Bloch’s theorem to be applied as they normally would for the full model [20, 21, 22]. The most important benefit of BMS is that the same reduced model can be used for any wave vector (or frequency), because application of Bloch BCs occurs *after* the model reduction. The reduced BMS model can be used with both the $\mathbf{k}(\omega)$ and $\omega(\mathbf{k})$ methods. The third chapter of this document discusses the BMS model reduction.

After a reduction of the boundary DOFs, a large majority of the remaining DOFs are typically contained in the boundary set. A secondary reduction of the boundary DOFs is developed using the characteristic constraint (CC) mode reduction in order to address this issue [20]. This boundary modal reduction requires that the boundary be coupled (by Bloch BCs) before modal reduction. Thus the reduction must be applied independently for every \mathbf{k} -point. This limits any efficiency gains from the reduction, and also precludes the reduced model from being solved via the $\mathbf{k}(\omega)$ method. The fourth chapter shows the boundary reduction of Ref. [20], and then presents an improved boundary treatment that makes use of a local interface reduction approach. The local

interface reduction is designed in the context of CMS to reduced substructure interfaces *before* reconnecting substructures. In the context of BMS, this means that the boundary can be reduced prior to enforcing Bloch BCs. This produces significant computational benefits, and also allows the resulting models to be solved via the $\mathbf{k}(\omega)$ method, as well as the $\omega(\mathbf{k})$ method.

1.3 Summary of Chapters and Contributions

This chapter gives a brief introduction to the field of periodic materials and discusses the value of the band-structure diagram. It also provides some idea of the numerical challenges posed by band-structure problems. Finally, a review of the literature dedicated to speeding up band-structure calculation is provided. The second chapter provides a review of fundamental topics involved in band-structure calculations. This includes modeling of unit cells, enforcing Bloch's theorem, and selection of the proper numerical algorithms for obtaining the band structure. The main scientific contribution made in the second chapter is the analysis of Bloch operator versus Bloch BC formulations. These two different methods of enforcing Bloch wave solutions within a unit cell have been developed in the literature, but very little analysis of the differences between the two methods has been published.

The third chapter develops the framework for extending CMS techniques to wave propagation problems in periodic media. This work shows that reduction of the unit-cell model *before* enforcement of the Bloch wave condition is possible and computationally advantageous. Also in the third chapter, a technique for enhancing the accuracy of the reduced order model is presented. This technique approximates the contribution of residual (i.e., truncated) modes, without expanding the size of the reduced-order model.

The fourth chapter describes the extension of boundary reduction techniques to unit-cell problems. These boundary reductions were originally developed as interface reductions for application to CMS problems, where the boundary sets to be reduced are independent, as opposed to a unit cell where the boundary must connect to itself in a relatively complex way. In order to adapt this to wave propagation problems, a formulation is developed wherein the boundary modes are

first partitioned according to certain boundary sets.

The fifth chapter discusses the extension of BMS techniques to electronic-structure problems. The vast majority of electronic band-structure calculations make use of PW expansion techniques because they outperform other types of discretization techniques. The BMS approach could render real-space methods, such as the FE and FD methods, competitive for electronic-structure calculations. The fifth chapter introduces Schrödinger's equation, the governing law describing electrons in materials, and describes how it can be used to obtain the electronic band structure. Then the performance of BMS is studied for the electronic-structure problem.

The sixth chapter describes the extension of BMS to full-spectrum phononic band-structure calculations. The full-spectrum band structure is useful for lattice dynamics applications like the prediction of thermal conductivity which require every branch of the band structure (not just the lowest) to be computed. Full-spectrum eigenvalue solutions become very computationally challenging when the system size is large. Thus a spectrum slicing strategy is employed to split the spectrum into frequency windows which can be treated more easily. The adaptation of BMS for spectrum slicing is also included, and makes use of quasi-static constraint mode sets to achieve accurate results.

The seventh chapter summarizes the conclusions and describes possible extensions of the research for further investigation.

Chapter 2

Fundamentals of Wave Propagation in Periodic Media

2.1 Introduction

Perhaps the most important concept used to describe waves in periodic materials is the band-structure diagram, which can alternatively be referred to as a dispersion diagram. This diagram shows a relationship between frequency and wave vector or equivalently, between energy and momentum. The concept of band structure is best illustrated through an example so a simple mass-spring material is analyzed.

2.2 Band Structure of a 1D Spring-Mass Chain

The 1D spring-mass chain illustrated in Fig. 2.1 shows a 2-mass unit cell that can be modeled using Hooke's law. Skipping over the derivation for the moment, an expression for the frequencies

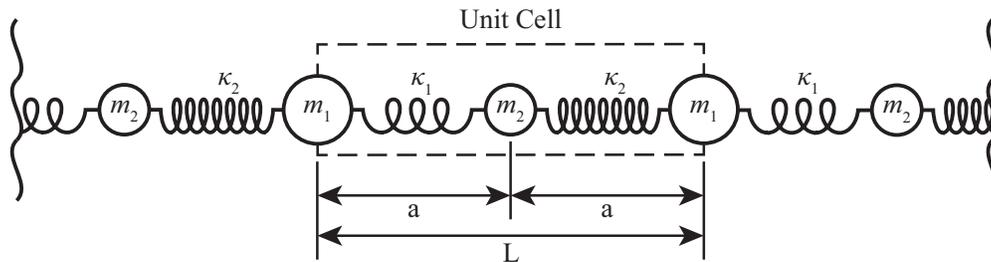


Figure 2.1: 1D periodic spring-mass chain with unit cell highlighted.

of wave propagation is given below,

$$\omega = \sqrt{\frac{\kappa_1 + \kappa_2}{2\bar{m}} \pm \sqrt{\frac{(\kappa_1 + \kappa_2)^2}{4\bar{m}^2} - \frac{\kappa_1\kappa_2}{m_1m_2} (2 - e^{-2ika} - e^{2ika})}}, \quad \bar{m} = \frac{m_1m_2}{m_1 + m_2} \quad (2.1)$$

where k is the wave number and is inversely proportional to the wavelength. The frequency-wave number relationship is plotted in Fig. 2.2. This band-structure diagram shows two branches: one for each DOF in the unit cell. Each point on the band-structure curves corresponds to a wave mode that can exist in the material. Each of these wave modes has a specific frequency and wavelength that can be read from the band-structure diagram. The band structure gives quite a bit of information about the waves. The group velocity of any wave is given by the slope of the band-structure curve, $v_g = d\omega / dk$. From this relationship it can be seen that the curving branches in Fig. 2.2 have a range of group velocities. So if a wave pulse containing a range of frequencies is sent through the material, waves of different frequencies disperse over time with some traveling faster and some traveling slower. Thus, materials whose band-structure branches aren't perfectly straight are dispersive. Band-structure curves make it quite simple to identify how dispersive a material is. This is why band structures are often referred to as dispersion diagrams.

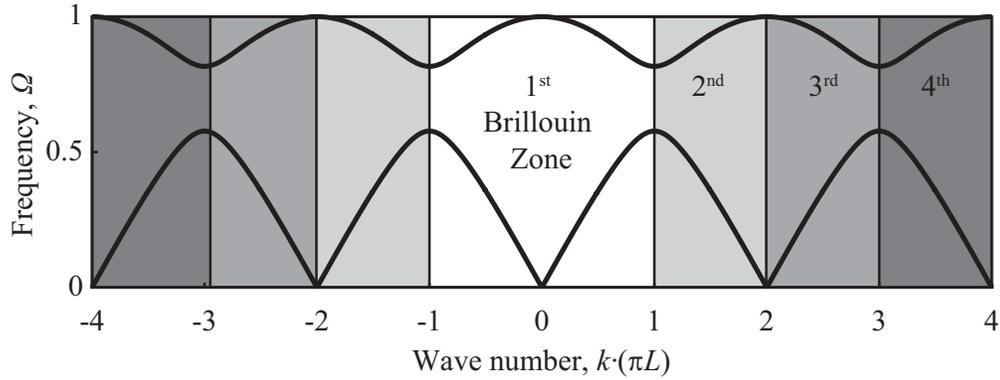


Figure 2.2: Band-structure diagram for 1D periodic mass-spring chain. Different BZs are shown in various shades of grey.

If $\kappa_1 = \kappa_2 = \kappa$, and $m_1 = m_2 = m$, Eq. (2.1) reduces to

$$\omega = \sqrt{\frac{\kappa}{m} \left(2 \pm \sqrt{2 + e^{-2ika} + e^{2ika}} \right)} \quad (2.2)$$

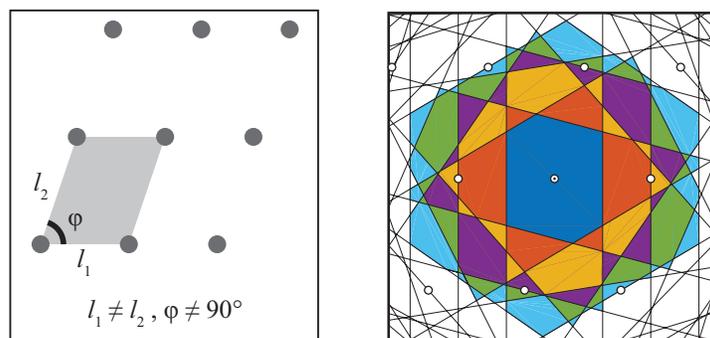
One can quickly verify that for $k = \pi/2a$, the innermost radical goes to zero and the two branches meet. This happens because the two branches are collapsing into a single branch that folds over on itself. For the case where $\kappa_1 \neq \kappa_2$, and/or $m_1 \neq m_2$, the branches are separate and create a band gap

— a frequency region where no wave solutions exist. Band gaps are a very interesting phenomenon that can have a large effect on material properties. For example, the presence of a band gap in the electronic structure can determine whether the material behaves like an electronic conductor, semiconductor, or insulator [3]. Much research has been dedicated to band-gap properties including research quantifying how strongly waves attenuate in band gaps, how to predict the band-gap edge frequencies without explicitly computing the band structure, and how to maximize the size of the band gap by optimizing the unit cell [23, 24].

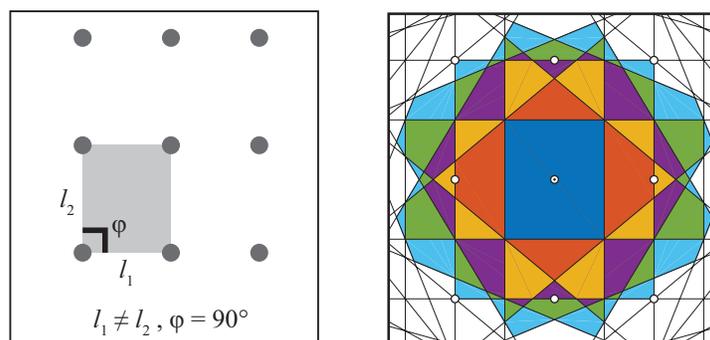
2.3 Lattice Symmetry and Brillouin Zones

The periodicity in real space also produces periodicity in reciprocal space. Thus a reciprocal space “unit cell” called the 1st BZ can be defined. Additional BZs wrap around the first one like layers of an onion. The BZs are shown as shaded regions in Fig. 2.2. Léon Brillouin published works in 1930 describing the process by which reciprocal space zones could be obtained for 2D and 3D lattices. The BZs become much more complex than in 1D, but the process by which they are computed is quite simple. In Brillouin’s words, the “lines [or planes] that limit the zones are perpendicular bisectors upon lines joining the center [point] to some points of the reciprocal lattice” [25]. The BZ depends entirely on the type of lattice. A lattice is an infinite set of points arranged periodically in space. Lattices can be characterized by their lattice vectors, \mathbf{a}_i . Every lattice has a set of symmetry operations, such as translations, reflections, and rotations, that can be applied to the lattice without changing the location of the lattice points. Two lattices that have the same group of symmetry operations are said to have the same Bravais lattice. In 1D there exists just one type of Bravais lattice. In 2D and 3D, there are 5 and 17 respectively. The 5 Bravais lattices in 2D are shown in Fig. 2.3 along with the corresponding BZ diagrams.

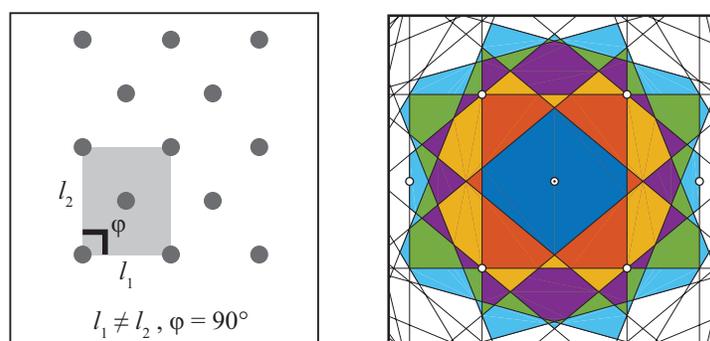
Due to the periodicity in reciprocal space, band-structure solutions need only be computed in the 1st BZ. Most materials exhibit time reversal symmetry. This ensures that waves traveling in one direction behave identically to waves traveling in the opposite direction. The resulting symmetry in the BZ reduces the wave-vector domain that needs to be considered by a factor of 2. In 2D



(a) Oblique



(b) Rectangular



(c) Centered rectangular

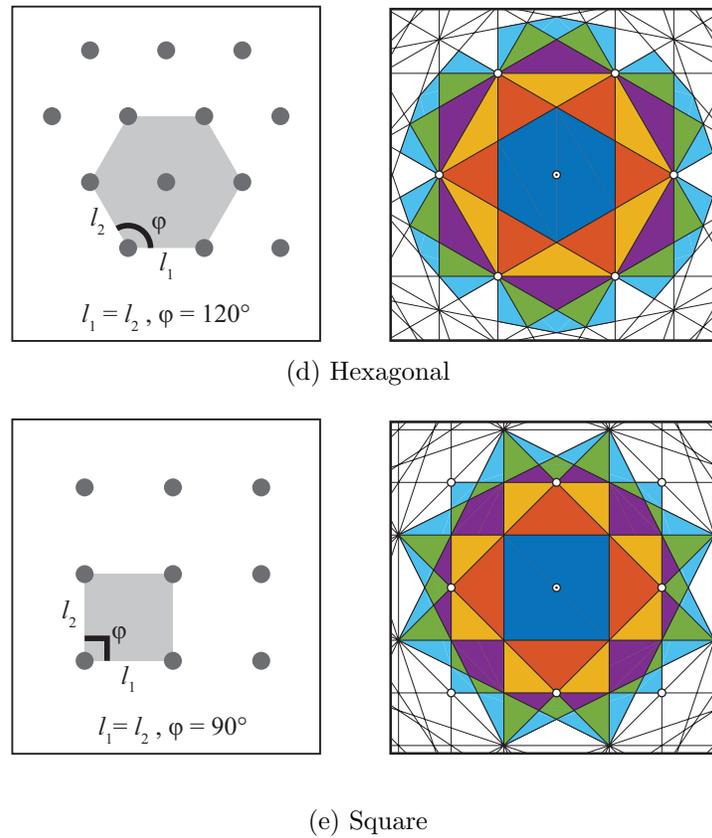


Figure 2.3: Lattice diagrams (left) and corresponding BZ diagrams (right) are shown for oblique (a), rectangular (b), centered-rectangular (c), hexagonal (d), and square (e) Bravais lattices.

and 3D materials, certain unit-cell symmetries can further reduce this wave vector domain. After all these reductions, the wave-vector domain that contains all unique wave solutions is referred to as the irreducible Brillouin zone (IBZ). For a 2D unit cell with symmetry about the x -, y -, and xy -planes the IBZ is one eighth the size of the full BZ. This is illustrated in Fig. 2.4 for a material with a circular inclusion in a square unit cell. If the xy symmetry is removed, the size of the IBZ is doubled. This is illustrated in Fig. 2.5 for a material with an elliptical inclusion in a square unit cell. The corners of the IBZ are high-symmetry points, and are typically labeled with letters. Naming conventions exist for the high-symmetry points of most common IBZs.

Rather than computing band-structure for an entire area in 2D or a volume in 3D, the band-structure is typically represented by tracing a path around the edge of the IBZ. This eases the computational cost and also makes the band-structure easier to represent in a flat plot. Some problems do require the entire area or volume to be considered, but for the most part the “flattened” band structure captures the salient properties of the material.

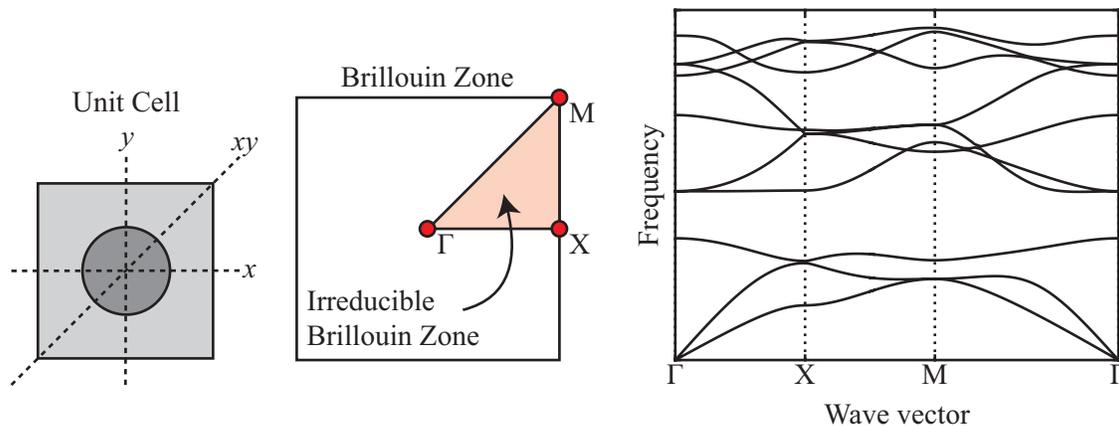


Figure 2.4: Schematics showing a square unit cell with a circular inclusion and corresponding symmetry planes (left), the corresponding BZ and IBZ with high symmetry points labeled (middle), and the corresponding band-structure diagram showing frequencies calculated along a path traced along the edge of the IBZ.

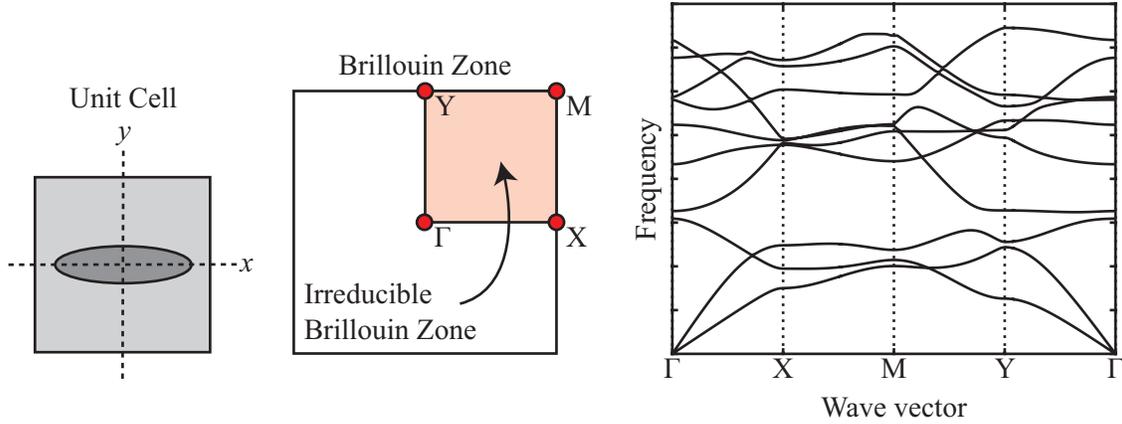


Figure 2.5: Square unit cell with an elliptical inclusion and corresponding symmetry planes (left), the corresponding BZ and IBZ with high symmetry points labeled (middle), and the corresponding band-structure diagram showing frequencies calculated along a path traced along the edge of the IBZ.

2.4 Bloch's Theorem

Felix Bloch formulated a theory to describe waves in periodic media [1]. His theory states that the wave field can be described as a periodic function that is modulated by a PW:

$$\mathbf{u}(\mathbf{x}, \mathbf{k}) = \underbrace{\tilde{\mathbf{u}}(\mathbf{x}, \mathbf{k})}_{\text{periodic function}} \underbrace{e^{i\mathbf{k}^T \mathbf{x}}}_{\text{plane wave}}. \quad (2.3)$$

A periodic function has the following form:

$$\tilde{\mathbf{u}}(\mathbf{x}, \mathbf{k}) = \tilde{\mathbf{u}}(\mathbf{x} + \mathbf{G}, \mathbf{k}), \quad (2.4)$$

where \mathbf{G} can be any lattice vector of the material. Combining Eqs. (2.3) and (2.4), reformulates Bloch's theorem as a relationship between the boundaries of the unit cell,

$$\mathbf{u}(\mathbf{x} + \mathbf{G}, \mathbf{k}) = \mathbf{u}(\mathbf{x}, \mathbf{k}) e^{-i\mathbf{k}^T \mathbf{G}}. \quad (2.5)$$

This theory was first formulated by Bloch, and thus bears his name, but earlier works by Kelvin, and Born [25] showed that the idea had been discovered previously.

2.4.1 Bloch Operator Versus Bloch Boundary Conditions

Bloch theory can be enforced in two different ways. In the first way, the Bloch displacement equation can be substituted into the equations of motion so that the PW terms are lumped into the stiffness matrix, and the vector of unknown terms contains only the periodic function. This is termed the Bloch operator approach. Alternatively, Bloch's theorem can be applied as a set of BCs that relates the DOFs on opposite boundaries of the unit cell via a PW term. This method is referred to as the Bloch BC approach.

2.4.1.1 1D Spring-Mass Example

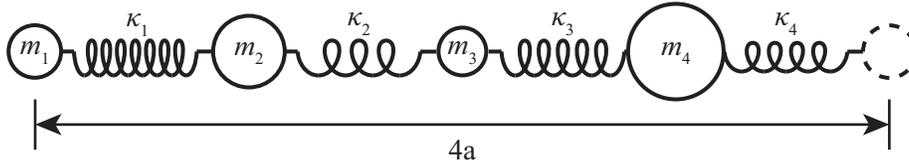


Figure 2.6: Unit cell for 1D periodic spring-mass chain with 4 DOFs per unit cell

To illustrate the differences between these two methods, another 1D mass-spring chain is used. This time the unit cell has 4 DOFs as shown in Fig. 2.6. Using the Bloch operator approach, the Bloch displacement field in Eq. (2.3) is explicitly considered when forming the equations of motion,

$$\left(\begin{bmatrix} \kappa_1 + \kappa_4 & -\kappa_1 e^{ika} & 0 & -\kappa_4 e^{-ika} \\ -\kappa_1 e^{-ika} & \kappa_1 + \kappa_2 & -\kappa_2 e^{ika} & 0 \\ 0 & -\kappa_2 e^{-ika} & \kappa_2 + \kappa_3 & -\kappa_3 e^{ika} \\ -\kappa_4 e^{ika} & 0 & -\kappa_3 e^{-ika} & \kappa_3 + \kappa_4 \end{bmatrix} - \omega^2 \begin{bmatrix} m_1 & 0 & 0 & 0 \\ 0 & m_2 & 0 & 0 \\ 0 & 0 & m_3 & 0 \\ 0 & 0 & 0 & m_4 \end{bmatrix} \right) \begin{Bmatrix} \tilde{u}_1 \\ \tilde{u}_2 \\ \tilde{u}_3 \\ \tilde{u}_4 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{Bmatrix} \quad (2.6)$$

If Bloch's theorem is applied via the Bloch BC approach, the equations of motion are as follows,

$$\left(\begin{bmatrix} \kappa_1 + \kappa_4 & -\kappa_1 & 0 & -\kappa_4 e^{-ik4a} \\ -\kappa_1 & \kappa_1 + \kappa_2 & -\kappa_2 & 0 \\ 0 & -\kappa_2 & \kappa_2 + \kappa_3 & -\kappa_3 \\ -\kappa_4 e^{ik4a} & 0 & -\kappa_3 & \kappa_3 + \kappa_4 \end{bmatrix} - \omega^2 \begin{bmatrix} m_1 & 0 & 0 & 0 \\ 0 & m_2 & 0 & 0 \\ 0 & 0 & m_3 & 0 \\ 0 & 0 & 0 & m_4 \end{bmatrix} \right) \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{Bmatrix} \quad (2.7)$$

The two sets of equations produce an identical band structure. This is because Eq. (2.7) is simply an orthogonal transformation of Eq. (2.6). This can be verified by relating the DOF vectors, $\tilde{\mathbf{u}}$ and \mathbf{u} ,

$$\begin{Bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{Bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & e^{ika} & 0 & 0 \\ 0 & 0 & e^{ik2a} & 0 \\ 0 & 0 & 0 & e^{ik3a} \end{bmatrix}}_{\mathbf{E}} \begin{Bmatrix} \tilde{u}_1 \\ \tilde{u}_2 \\ \tilde{u}_3 \\ \tilde{u}_4 \end{Bmatrix}. \quad (2.8)$$

The matrix \mathbf{E} is an orthogonal matrix that can be used to perform a similarity transform on the stiffness matrix of Eq. (2.7) to obtain the stiffness matrix of Eq. (2.6). Although they are analytically equivalent, the two sets of equations differ in the way that they can be treated numerically. The Bloch BC matrices have very few terms that depend on the wave number, k . Only those DOFs that are on the boundary of the unit cell end up with any wave-number dependence. This means that if the system matrices are partitioned into interior and boundary components, the interior component will have no wave-number dependence and will thus remain constant throughout the BZ. This is an important property that the BMS model reduction takes advantage of.

The spring-mass system is useful due to its simplicity, but it also hides some of the important differences between the Bloch operator approach and the Bloch BC approach. One feature that isn't observed in the previous example is that the mass matrices are not generally the same for the two methods. If the mass matrix had off-diagonal coupling terms (as is the case for FE models), then the Bloch operator mass matrix would have no wave-number dependence, but the Bloch BC mass matrix would indeed have a wave-number dependence. Another thing that the spring mass

model cannot show is the effect that discretization has on the matrices in each case. For this reason another example system is included.

2.4.1.2 Bloch Boundary Condition Formulation for Finite-Element Models

In order to illustrate the differences between the Bloch operator and Bloch BC approaches that are not apparent from the spring-mass model, an FE example is presented below. The Bloch BC approach can be applied to traditional FEs. A brief outline of the traditional FE derivation is included, followed by a discussion of how Bloch BCs can be applied. Then the Bloch operator approach is presented. The Bloch operator approach requires a slight deviation from the traditional FE derivation. This is demonstrated in the following discussion.

Neglecting any external forcing, the Lagrangian of an elastodynamic solid within domain Ω can be expressed in tensorial notation,

$$L = T - V = \frac{1}{2} \int_{\Omega} \rho \dot{\mathbf{u}} \cdot \dot{\mathbf{u}} d\Omega - \frac{1}{2} \int_{\Omega} \boldsymbol{\varepsilon} : \boldsymbol{\sigma} d\Omega \quad (2.9)$$

where $\boldsymbol{\varepsilon}$ is the strain tensor, and $\boldsymbol{\sigma}$ is the stress tensor. The constitutive equation can be used to express the stress in terms of the strain,

$$\boldsymbol{\sigma} = \mathbf{C} : \boldsymbol{\varepsilon} \quad (2.10)$$

where \mathbf{C} is the elasticity tensor and ρ is the density. Both of these material parameters are known functions of space. For small displacements, the strain can be found by taking the symmetric gradient of the displacement field,

$$\boldsymbol{\varepsilon} = \nabla^S \mathbf{u} = \frac{1}{2} (\nabla \mathbf{u} + (\nabla \mathbf{u})^T) \quad (2.11)$$

At this point the displacement field can be expanded in terms of a set of basis functions, or, as they are more commonly referred to in FE literature, shape functions.

$$\mathbf{u} = \sum_{i=1}^p N_i \mathbf{q}_i = \mathbf{N} \mathbf{q} \quad (2.12)$$

The number of shape functions, p , increases with the order of the element. Each expansion coefficient \mathbf{q}_i corresponds to the displacement vector of a node in the element. Note that \mathbf{q} is a discrete displacement vector. For each node in the element, the corresponding shape function is a Lagrange interpolation polynomial that has a value of one at it's corresponding node, and a value of zero at all other nodes. For a 3D displacement field, \mathbf{N} and \mathbf{q} are expressed in matrix form as follows,

$$\mathbf{N} = \left[\begin{array}{ccc|ccc} N_1 & 0 & 0 & \cdots & N_p & 0 & 0 \\ 0 & N_1 & 0 & & 0 & N_p & 0 \\ 0 & 0 & N_1 & & 0 & 0 & N_p \end{array} \right], \quad (2.13)$$

$$\mathbf{q}^T = \left[\begin{array}{ccc|ccc} q_{1x} & q_{1y} & q_{1z} & \cdots & q_{px} & q_{py} & q_{pz} \\ \underbrace{\hspace{10em}}_{\mathbf{q}_1^T} & & & & \underbrace{\hspace{10em}}_{\mathbf{q}_p^T} & & \end{array} \right]. \quad (2.14)$$

At this point, The Lagrangian can be expressed in matrix form:

$$L = \frac{1}{2} \dot{\mathbf{q}}^T \underbrace{\left(\int_{\Omega} \rho \mathbf{N}^T \mathbf{N} d\Omega \right)}_{\mathbf{M}} \dot{\mathbf{q}} - \frac{1}{2} \mathbf{q}^T \underbrace{\left(\int_{\Omega} (\mathbf{D}\mathbf{N})^T \mathbf{C} (\mathbf{D}\mathbf{N}) d\Omega \right)}_{\mathbf{K}} \mathbf{q}, \quad (2.15)$$

where \mathbf{D} is the symmetric-gradient operator expressed in matrix form. For 2D plane strain, and 3D elastic systems, the symmetric-gradient operators are respectively,

$$\mathbf{D}_{2D} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix}, \quad \text{and} \quad \mathbf{D}_{3D} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 & 0 \\ 0 & \frac{\partial}{\partial y} & 0 \\ 0 & 0 & \frac{\partial}{\partial z} \\ 0 & \frac{\partial}{\partial z} & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} & 0 \end{bmatrix}, \quad (2.16)$$

Finally, Lagrange's equations can be applied to the Lagrangian in order to find the discrete equations of motion,

$$0 = \frac{d}{dt} \frac{\partial L}{\partial \dot{\mathbf{q}}} - \frac{\partial L}{\partial \mathbf{q}} = \mathbf{M} \ddot{\mathbf{q}} + \mathbf{K} \mathbf{q}. \quad (2.17)$$

This represents the traditional FE equations of motion. In practice, the global FE equations of motion shown above are not obtained by directly integrating over the entire model domain. Rather,

element mass and stiffness matrices, \mathbf{M}^e and \mathbf{K}^e , are obtained by numerically integrating over the individual element domains using Gauss quadrature. The master mass and stiffness matrices, \mathbf{M} and \mathbf{K} , are then obtained by adding the element matrices into the appropriate locations of the global matrices using the direct stiffness method [26, 27].

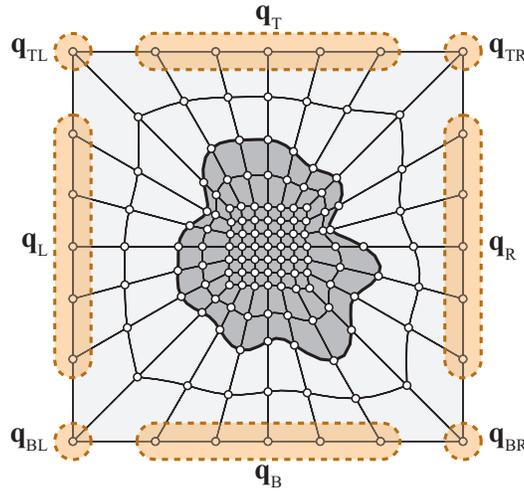


Figure 2.7: Isolated unit-cell FE mesh for general periodic 2D material with boundary sets labeled. The isolated unit cell is obtained by cutting the unit cell out of the periodic material so that all boundaries are free.

In order to apply Bloch's theorem, the DOFs on the boundaries of the unit cell must be related to each other via phase terms that depend on the wave vector. To help illustrate this, an FE mesh of the isolated unit cell for the 2D infinitely periodic material from Fig. 2.7 is shown. The boundary DOF sets that must be related via Bloch BCs are outlined and labeled. For a discrete model, Eq. (2.3) becomes a set of equations relating the DOFs on the boundary of the cell. The DOFs on the right boundary are related to the DOFs on the left boundary, and the DOFs on the top boundary are related to the DOFs on the bottom boundary.

$$\mathbf{q}_R = \mathbf{q}_L \lambda_x, \quad \mathbf{q}_T = \mathbf{q}_B \lambda_y, \quad (2.18)$$

where $\lambda_x = e^{-ik_x L_x}$, and $\lambda_y = e^{-ik_y L_y}$. A similar set of equations is used to link the corner DOFs.

$$\mathbf{q}_{BR} = \mathbf{q}_{BL} \lambda_x, \quad \mathbf{q}_{TR} = \mathbf{q}_{BL} \lambda_x \lambda_y, \quad \mathbf{q}_{TL} = \mathbf{q}_{BL} \lambda_y. \quad (2.19)$$

These Bloch BC equations can be used to describe the free DOF vector, \mathbf{q} , in terms of the periodic DOF vector, $\hat{\mathbf{q}}$,

$$\begin{array}{c} \overbrace{\left[\begin{array}{c} \mathbf{q}_I \\ \mathbf{q}_L \\ \mathbf{q}_R \\ \mathbf{q}_B \\ \mathbf{q}_T \\ \mathbf{q}_{BL} \\ \mathbf{q}_{BR} \\ \mathbf{q}_{TR} \\ \mathbf{q}_{TL} \end{array} \right]}^{\mathbf{q}} = \overbrace{\left[\begin{array}{cccc} \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \lambda_x \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \lambda_y \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \lambda_x \mathbf{I} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \lambda_x \lambda_y \mathbf{I} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \lambda_y \mathbf{I} \end{array} \right]}^{\mathbf{P}} \overbrace{\left[\begin{array}{c} \hat{\mathbf{q}}_I \\ \hat{\mathbf{q}}_L \\ \hat{\mathbf{q}}_B \\ \hat{\mathbf{q}}_{BL} \end{array} \right]}^{\hat{\mathbf{q}}}. \end{array} \quad (2.20)$$

The Bloch periodicity matrix, \mathbf{P} , can be decomposed into components multiplying λ_x and λ_y ,

$$\mathbf{P} = \mathbf{P}_0 + \lambda_x \mathbf{P}_x + \lambda_y \mathbf{P}_y + \lambda_x \lambda_y \mathbf{P}_{xy}. \quad (2.21)$$

The Bloch periodicity sub-matrices, \mathbf{P}_0 , \mathbf{P}_x , \mathbf{P}_y , and \mathbf{P}_{xy} , are formed by placing ones where entries in the free DOF vector should link together with entries in the periodic DOF vector. Note that the DOF vectors will not typically be sorted as shown in Eq. (2.20). In order for the Bloch periodicity matrix to be compatible with the mass and stiffness matrices, they must have the same DOF sorting. This can be accomplished by re-sorting the rows and columns of the mass and stiffness matrices or alternatively by re-sorting the rows of the periodicity transformation.

Substituting Eq. (2.20) into Eq. (2.17) and pre-multiplying by \mathbf{P} gives the Bloch-periodic equations of motion,

$$\left(\mathbf{P}^\dagger \mathbf{K} \mathbf{P} \right) \hat{\mathbf{q}} + \left(\mathbf{P}^\dagger \mathbf{M} \mathbf{P} \right) \ddot{\hat{\mathbf{q}}} = 0. \quad (2.22)$$

where $(\cdot)^\dagger$ represents the Hermitian (complex conjugate transpose). The mass and stiffness matrices are formed using a traditional approach and Bloch BCs are simply applied as a secondary step.

Thus, the Bloch BC approach can just as easily be applied to mass and stiffness matrices that are

obtained from a commercial FE package. This makes available the vast element libraries that have been developed for commercial software.

To obtain the characteristic equation for the structural model, a time-harmonic solution, $\hat{\mathbf{q}} = \bar{\mathbf{q}}e^{i\omega t}$, is assumed. The bar notation indicates that the vector contains amplitudes (with no time dependence). To avoid further complicating the notation however, the bar is not included subsequently. The resulting characteristic equation is an eigenvalue problem that can be solved for the band-structure frequencies:

$$\mathbf{P}^\dagger (\mathbf{K} - \omega^2 \mathbf{M}) \mathbf{P} \hat{\mathbf{q}} = 0. \quad (2.23)$$

2.4.1.3 Bloch Operator Formulation for Finite-Element Models

The second approach to enforcing Bloch's theorem for an FE mesh is referred to as the Bloch operator approach [18]. In order to obtain the Bloch operator equations of motion, Bloch's form for the displacement field must be considered within the element derivation. Using Bloch's displacement field, the gradient operator can be expressed in terms of the periodic displacement field $\tilde{\mathbf{u}}$,

$$\nabla \mathbf{u} = (\nabla \tilde{\mathbf{u}} + i\mathbf{k}^T \times \tilde{\mathbf{u}}) e^{i\mathbf{k}^T \mathbf{x}}. \quad (2.24)$$

Substituting this into the Lagrangian, and applying Lagrange's equation gives the Bloch operator equations of motion,

$$0 = \tilde{\mathbf{M}} \ddot{\tilde{\mathbf{q}}} + \tilde{\mathbf{K}} \tilde{\mathbf{q}}, \quad (2.25)$$

where

$$\tilde{\mathbf{M}} = \int_{\Omega} \rho \mathbf{N}^\dagger \mathbf{N} d\Omega, \quad \tilde{\mathbf{K}} = \int_{\Omega} (\tilde{\mathbf{D}} \mathbf{N})^\dagger \mathbf{C} (\tilde{\mathbf{D}} \mathbf{N}) d\Omega. \quad (2.26)$$

The Bloch symmetric-gradient operator, $\tilde{\mathbf{D}}$ includes additional terms due to the differentiation of the plane wave. For a 3D system, the Bloch symmetric-gradient operator is as follows,

$$\tilde{\mathbf{D}}_{3D} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 & 0 \\ 0 & \frac{\partial}{\partial y} & 0 \\ 0 & 0 & \frac{\partial}{\partial z} \\ 0 & \frac{\partial}{\partial z} & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} & 0 \end{bmatrix} + \begin{bmatrix} i\kappa_x & 0 & 0 \\ 0 & i\kappa_y & 0 \\ 0 & 0 & i\kappa_z \\ 0 & i\kappa_z & i\kappa_y \\ i\kappa_z & 0 & i\kappa_x \\ i\kappa_y & i\kappa_x & 0 \end{bmatrix}. \quad (2.27)$$

Equation (2.26) explicitly contains the PW but the DOF vector is not constrained to be periodic. The periodic displacement vector, $\tilde{\mathbf{q}}$, can be constrained to be periodic by applying a periodicity transformation,

$$\begin{array}{c} \tilde{\mathbf{q}} \\ \left\{ \begin{array}{c} \mathbf{q}_I \\ \mathbf{q}_L \\ \mathbf{q}_R \\ \mathbf{q}_B \\ \mathbf{q}_T \\ \mathbf{q}_{BL} \\ \mathbf{q}_{BR} \\ \mathbf{q}_{TR} \\ \mathbf{q}_{TL} \end{array} \right\} \end{array} = \begin{array}{c} \tilde{\mathbf{P}} \\ \left[\begin{array}{cccc} \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{array} \right] \end{array} \begin{array}{c} \hat{\mathbf{q}} \\ \left\{ \begin{array}{c} \mathbf{q}_I \\ \mathbf{q}_L \\ \mathbf{q}_B \\ \mathbf{q}_{BL} \end{array} \right\}. \end{array} \quad (2.28)$$

This is substituted into Eq. (2.26) to form the Bloch-periodic equations of motion.

$$\left(\tilde{\mathbf{P}}^T \tilde{\mathbf{K}} \tilde{\mathbf{P}} \right) \hat{\mathbf{q}} + \left(\tilde{\mathbf{P}}^T \tilde{\mathbf{M}} \tilde{\mathbf{P}} \right) \ddot{\hat{\mathbf{q}}} = 0. \quad (2.29)$$

Once again, a time-harmonic solution, $\hat{\mathbf{q}} = \bar{\bar{\mathbf{q}}} e^{i\omega t}$, is assumed in order to obtain the characteristic equation for the structural model, but the bar notation is not carried forward to avoid more severely complicating the notation. The resulting eigenvalue problem,

$$\tilde{\mathbf{P}}^T (\tilde{\mathbf{K}} - \omega^2 \tilde{\mathbf{M}}) \tilde{\mathbf{P}} \hat{\mathbf{q}} = 0, \quad (2.30)$$

can be solved for the band-structure frequencies.

2.5 Band-Structure Calculations

The band structure can be calculated in two general ways. The first way involves specifying the wave vector and solving the resulting eigenvalue problem for the frequencies. By stepping through all of the wave vectors of interest, the $\omega(\mathbf{k})$ band structure is formed. The second way involves specifying the frequency and then solving for the wave vector. Then by stepping through all the frequencies of interest, the $\mathbf{k}(\omega)$ band structure is formed.

2.5.1 Frequency-as-a-Function-of-Wave-Vector Solution: $\omega(\mathbf{k})$ Method

The $\omega(\mathbf{k})$ formulation involves stepping through the BZ and solving for the frequencies of wave propagation at each wave vector of interest. If the Bloch-periodic equations of motion are obtained using the Bloch BC approach (Eq. (2.22)), then the corresponding eigenvalue problem is as follows,

$$\left(\hat{\mathbf{K}} - \omega^2 \hat{\mathbf{M}}\right) \hat{\mathbf{q}} = \mathbf{0}, \quad \hat{\mathbf{K}} = \mathbf{P}^\dagger \mathbf{K} \mathbf{P}, \quad \hat{\mathbf{M}} = \mathbf{P}^\dagger \mathbf{M} \mathbf{P}. \quad (2.31)$$

Otherwise, if the Bloch-periodic equations of motion are obtained using the Bloch operator approach (Eq. (2.29)), then the corresponding eigenvalue problem is as follows,

$$\left(\hat{\tilde{\mathbf{K}}} - \omega^2 \hat{\tilde{\mathbf{M}}}\right) \hat{\tilde{\mathbf{q}}} = \mathbf{0}, \quad \hat{\tilde{\mathbf{K}}} = \tilde{\mathbf{P}}^\dagger \tilde{\mathbf{K}} \tilde{\mathbf{P}}, \quad \hat{\tilde{\mathbf{M}}} = \tilde{\mathbf{P}}^\dagger \tilde{\mathbf{M}} \tilde{\mathbf{P}}. \quad (2.32)$$

In the above two sets of equations, the hat accent is used to denote that periodicity has been enforced. The tilde accent is used to denote that the PW has been considered in the formulation so that the solution only represents the periodic component of the displacement field in Eq. (2.3). Thus the eigenvectors resulting from Eq. (2.31) contain the full solution field (PW and periodic component), but the eigenvectors resulting from Eq. (2.32) contain just the periodic component of the displacement field.

Once the matrices have been obtained, there is not much practical difference between the solution approach (one simply uses an eigenvalue solver to solve the generalized eigenproblem of

Eq. (2.32) or Eq. (2.31)). Assuming dense enough meshes, the frequencies obtained with the two eigenvalue problems should match, so the same band structure is obtained either way. The eigenvectors from one eigenproblem can be made to match the eigenvectors from the other problem by simply multiplying or dividing by the PW component. This can be done using an orthogonal transformation matrix similar to the one shown in Eq. (2.8).

2.5.1.1 Comparison of Bloch Boundary Conditions and Bloch Operator Accuracy for 2D Plane-Strain Finite-Element Models

In the case of the spring mass model shown in section 2.4.1.1 the Bloch operator and Bloch BC methods produce different matrices but can be shown to have identical eigenvalues. This will hold true for any spring-mass models, however in the case of FE models, Bloch operator and Bloch BCs are no longer equivalent. They do converge to the same result but not in an identical manner. To demonstrate this, a 2D material with circular inclusions is modeled with the Bloch operator approach and with the Bloch BC approach. For each approach, the mesh is repeatedly refined. The resulting $\omega(\mathbf{k})$ band structures for each mesh are compared to the band structure obtained with a highly converged mesh. The maximum frequency error for each mesh is plotted versus model size in Fig. 2.8. Separate error curves are plotted for linear, quadratic, and cubic elements obtained via both the Bloch operator and the Bloch BC approaches.

The Bloch BC curves for all three element types have lower error than the corresponding Bloch operator curves. This occurs because the Bloch operator mass and stiffness matrices must integrate the PW displacement in addition to any periodic displacement. According to this reasoning, the lower order Bloch operator elements are expected to perform much worse than their Bloch BC counterparts compared to the higher order Bloch operator elements. This however is not the case because with lower order elements formed with either approach, the error is dominated by the inability to exactly capture the circular geometry. This large geometric error affects the Bloch operator and Bloch BC elements equally and effectively washes out the difference caused by the plane-wave integration. To see the effect of the plane-wave integration without any geometric

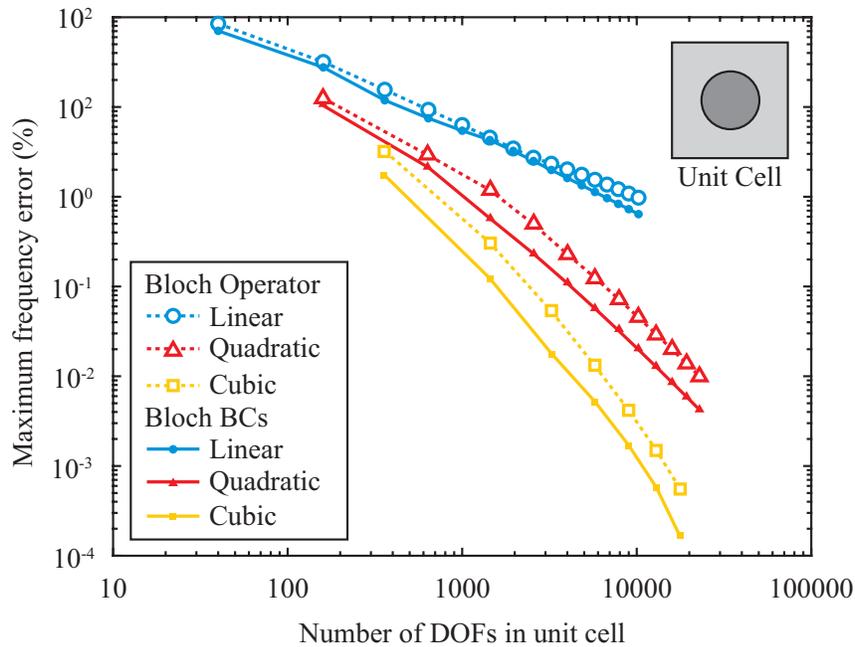


Figure 2.8: Comparison of Bloch operator and Bloch BC frequency errors versus model size. The maximum frequency error is obtained by comparing the lowest 10 band-structure branches of the 2D circular inclusion unit cell (shown in the inset) to a highly refined mesh and taking the maximum relative frequency error over the BZ.

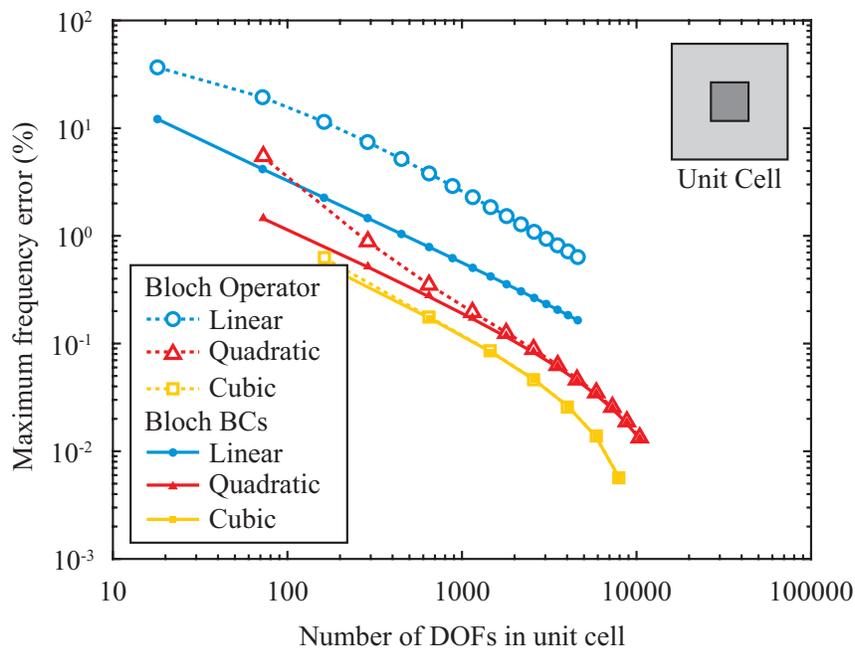


Figure 2.9: Comparison of Bloch operator and Bloch BC frequency errors versus model size. The maximum frequency error is obtained by comparing the lowest 10 band-structure branches of the 2D square inclusion unit cell (shown in the inset) to a highly refined mesh and taking the maximum relative frequency error over the BZ.

modeling error, a second material is analyzed with a square inclusion rather than a circular inclusion. The frequency error plot in Fig. 2.9 shows that indeed when geometric error is eliminated, the lower order Bloch operator elements perform more poorly compared to their counterparts. An additional phenomenon that can be observed in this figure is that the higher order Bloch operator curves coalesce with the Bloch BC curves when the meshes are dense enough. This occurs because the plane-wave component of the displacement field varies relatively smoothly compared to the periodic field.

2.5.2 Wave-Vector-as-a-Function-of-Frequency Solution: $\mathbf{k}(\omega)$ Method

In some situations it may be desirable to solve the wave propagation problem for the wave vector in terms of the frequency. This $\mathbf{k}(\omega)$ problem involves stepping through the frequencies of interest and solving for the wave vectors. The $\mathbf{k}(\omega)$ method is a bit more complicated than the $\omega(\mathbf{k})$ method, but it produces both real and complex wave-vector solutions. Complex wave-vector solutions can be very valuable for certain types of analysis because they contain information about the attenuation of waves in space. The process by which the $\mathbf{k}(\omega)$ band structure is computed depends heavily on the form of the stiffness matrix, and thus differs for Bloch operator and Bloch BC matrices.

Figure 2.10 shows an example of a $\mathbf{k}(\omega)$ band structure plotted in 3D to emphasize that wave number solutions can be real, imaginary, or complex. In the absence of damping, the real portion of the band structure should exactly match the $\omega(\mathbf{k})$ band structure. The most common representation of the $\mathbf{k}(\omega)$ band structure uses a 2D axis and plots the imaginary component of every solution on the negative x-axis and the real component of every solution on the positive x-axis. This is shown in Fig. 2.11. Although the “flattened” representation makes presentation more convenient, some amount of information is lost. For example, the true band gap extends from about 28 to 46 kHz, but the flattened representation suggests that the band gap is much smaller.

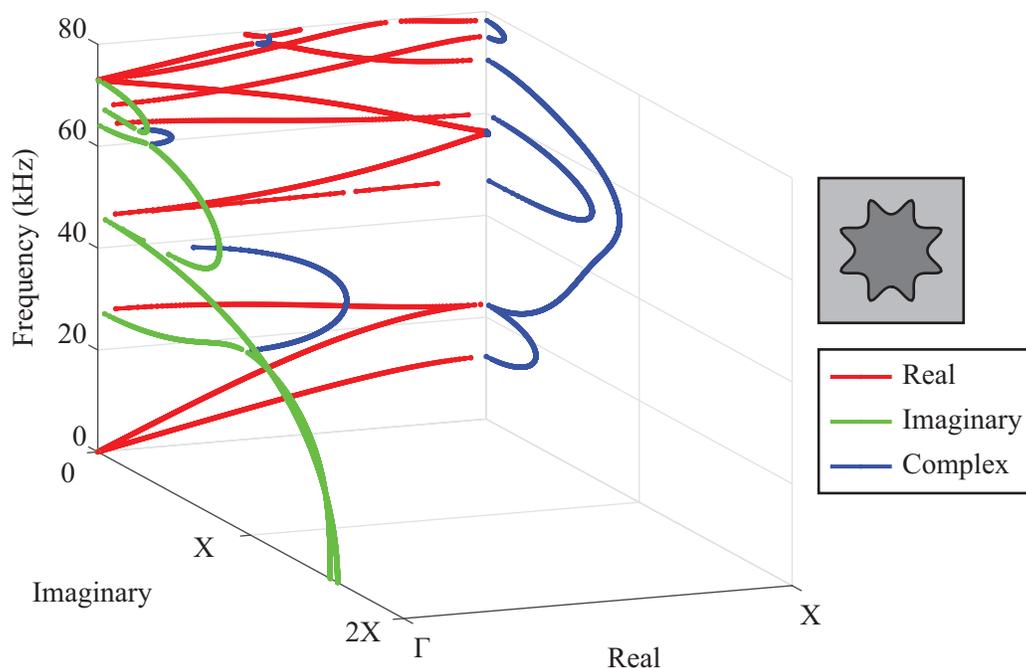


Figure 2.10: 3D representation of the $\mathbf{k}(\omega)$ band structure for the Γ -X direction of the eight-lobe inclusion unit cell (shown in inset). The purely-real wave-number solutions, purely-imaginary wave-number solutions, and complex wave-number solutions are shown in red, green, and blue respectively.

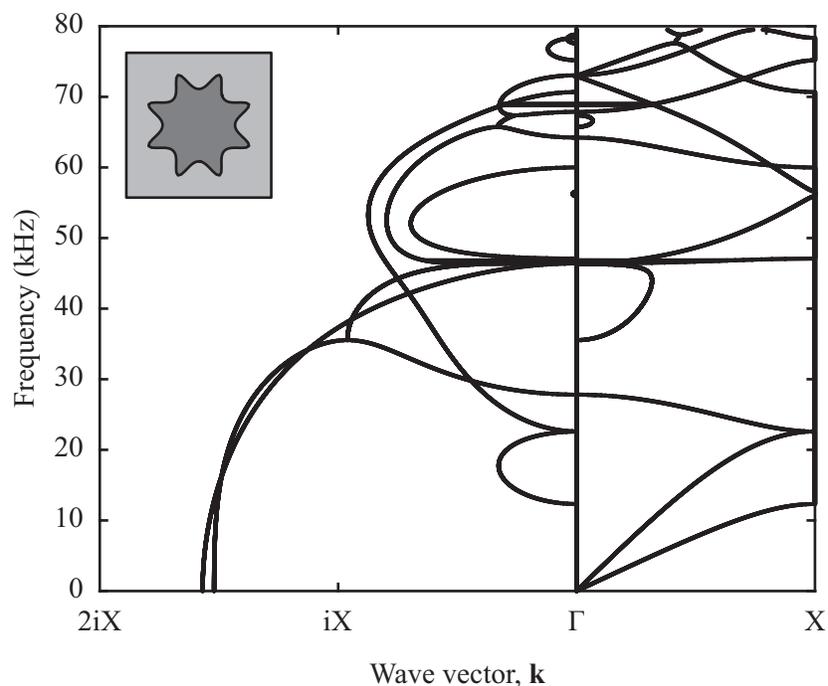


Figure 2.11: Conventional representation of the $\mathbf{k}(\omega)$ band structure for the Γ -X direction of the eight-lobe inclusion unit cell (shown in inset). Real wave number solutions are shown on the right (positive x-axis) and imaginary wave number solutions are on the left (negative x-axis).

2.5.2.1 $\mathbf{k}(\omega)$ Solutions Using the Bloch Boundary Condition Formulation

Computation of the $\mathbf{k}(\omega)$ band structure via the Bloch BC formulation has been thoroughly studied in the context of the wave FE method [28, 29]. A brief overview is included here for the case of 2D wave propagation. To solve the $\mathbf{k}(\omega)$ problem, one can begin by setting $\mathbf{D}(\omega) = \mathbf{K} - \omega^2 \mathbf{M}^\dagger$ in the free equations of motion,

$$\mathbf{D}(\omega)\mathbf{q} = \mathbf{0}. \quad (2.33)$$

After applying Bloch BCs and expanding \mathbf{P} as in Eq. (2.21), this becomes

$$\left[\hat{\mathbf{D}}_1 + \lambda_x \hat{\mathbf{D}}_2 + \lambda_x^* \hat{\mathbf{D}}_2^\top + \lambda_y \hat{\mathbf{D}}_3 + \lambda_y^* \hat{\mathbf{D}}_3^\top + \lambda_x \lambda_y^* \hat{\mathbf{D}}_4 + \lambda_x^* \lambda_y \hat{\mathbf{D}}_4^\top + \lambda_x \lambda_y \hat{\mathbf{D}}_5 + \lambda_x^* \lambda_y^* \hat{\mathbf{D}}_5^\top \right] \hat{\mathbf{q}} = \mathbf{0}, \quad (2.34)$$

where

$$\hat{\mathbf{D}}_1 = \mathbf{P}_0^\top \mathbf{D}(\omega) \mathbf{P}_0 + \mathbf{P}_x^\top \mathbf{D}(\omega) \mathbf{P}_x + \mathbf{P}_y^\top \mathbf{D}(\omega) \mathbf{P}_y + \mathbf{P}_{xy}^\top \mathbf{D}(\omega) \mathbf{P}_{xy}, \quad (2.35)$$

$$\hat{\mathbf{D}}_2 = \mathbf{P}_0^\top \mathbf{D}(\omega) \mathbf{P}_x + \mathbf{P}_y^\top \mathbf{D}(\omega) \mathbf{P}_{xy}, \quad (2.36)$$

$$\hat{\mathbf{D}}_3 = \mathbf{P}_0^\top \mathbf{D}(\omega) \mathbf{P}_y + \mathbf{P}_x^\top \mathbf{D}(\omega) \mathbf{P}_{xy}, \quad (2.37)$$

$$\hat{\mathbf{D}}_4 = \mathbf{P}_x^\top \mathbf{D}(\omega) \mathbf{P}_y, \quad (2.38)$$

$$\hat{\mathbf{D}}_5 = \mathbf{P}_0^\top \mathbf{D}(\omega) \mathbf{P}_{xy}. \quad (2.39)$$

The direction of wave propagation is given by the ratio of propagation constants $\mu_x = k_x L_x$ and $\mu_y = k_y L_y$. If the ratio between μ_x and μ_y is rational, then one can set $\mu_x = \sigma a$ and $\mu_y = \sigma b$ where a and b are integers. Setting $\gamma = e^{-i\sigma}$ gives,

$$\lambda_x = \gamma^a, \quad \lambda_x^* = \gamma^{-a}, \quad \lambda_y = \gamma^b, \quad \lambda_y^* = \gamma^{-b}. \quad (2.40)$$

Substituting Eq. (2.40) into (2.34) and multiplying by γ^{a+b} gives the polynomial eigenvalue equation,

$$\left[\gamma^{a+b} \hat{\mathbf{D}}_1 + \gamma^{2a+b} \hat{\mathbf{D}}_2 + \gamma^b \hat{\mathbf{D}}_2^\top + \gamma^{2b+a} \hat{\mathbf{D}}_3 + \gamma^a \hat{\mathbf{D}}_3^\top + \gamma^{2a} \hat{\mathbf{D}}_4 + \gamma^{2b} \hat{\mathbf{D}}_4^\top + \gamma^{2a+2b} \hat{\mathbf{D}}_5 + \hat{\mathbf{D}}_5^\top \right] \hat{\mathbf{q}} = \mathbf{0}. \quad (2.41)$$

[†]Note that \mathbf{D} is repurposed here to denote the dynamical matrix rather than the symmetric gradient operator introduced in the FE derivation.

Given an m^{th} order polynomial eigenvalue problem of the form

$$\left[\sum_{j=0}^m \mathbf{A}_j \gamma^j \right] \mathbf{v} = 0, \quad (2.42)$$

a process called linearization can be used to formulate this as a general linear eigenvalue problem [30],

$$\left(\begin{bmatrix} \mathbf{A}_0 & & & \\ & \mathbf{J}_1 & & \\ & & \ddots & \\ & & & \mathbf{J}_{m-1} \end{bmatrix} - \gamma \begin{bmatrix} -\mathbf{A}_1 & \cdots & -\mathbf{A}_{m-1} & -\mathbf{A}_m \\ & \mathbf{J}_1 & & \\ & & \ddots & \\ & & & \mathbf{J}_{m-1} & \mathbf{0} \end{bmatrix} \right) \begin{Bmatrix} \mathbf{v} \\ \gamma \mathbf{v} \\ \vdots \\ \gamma^{m-1} \mathbf{v} \end{Bmatrix} = 0, \quad (2.43)$$

where \mathbf{J}_i can in principle be any square matrix, but should in practice be chosen to avoid ill conditioning and to retain maximum sparsity of the system. A common selection is $\mathbf{J}_i = \mathbf{I} \alpha_i$, where α_i is a scaling parameter. Usually, $\alpha_i = 1$ is sufficient, but in some cases, better performance can be obtained by selecting α_i based on the matrices \mathbf{A}_1 through \mathbf{A}_m . After the linear eigenvalue problem, the quadratic eigenvalue problem ($m = 2$) has received the most attention because it arises from many naturally occurring problems [31]:

$$\left(\begin{bmatrix} \mathbf{A}_0 & \mathbf{0} \\ \mathbf{0} & \mathbf{J} \end{bmatrix} - \gamma \begin{bmatrix} -\mathbf{A}_1 & -\mathbf{A}_2 \\ \mathbf{J} & \mathbf{0} \end{bmatrix} \right) \begin{Bmatrix} \mathbf{q} \\ \gamma \mathbf{q} \end{Bmatrix} = \begin{Bmatrix} \mathbf{0} \\ \mathbf{0} \end{Bmatrix}. \quad (2.44)$$

The linearized quadratic eigenvalue problem can be made symmetric by setting $\mathbf{J} = -\mathbf{A}_2^{\text{T}}$.

The solution of Eq. (2.42) via the linearization approach of Eq. (2.43) is difficult for a few reasons. First, the matrix sizes in the linearized eigenproblem are $(nm \times nm)$ which can be very large. Furthermore, the matrices $\hat{\mathbf{D}}_2 - \hat{\mathbf{D}}_5$ are generally singular. This causes iterative solvers to fail and thus requires the use of direct eigenvalue solvers which are typically much slower. These difficulties severely limit the size of the unit-cell models that can be considered.

If the ratio of propagation constants is not rational, then it is still possible to proceed to Eq. (2.34), but a and b are no longer integers, so it is not possible to use a polynomial eigenvalue approach. Rather, a root finding algorithm such as newton's method needs to be used to solve for

γ . This type of solution approach can be very time consuming for large models, requires a guess for the solutions, and does not produce the eigenvectors without additional steps being taken.

The computational cost of performing a direct solution of Eq. (2.43) can be very large, so it is very common to perform a dynamic reduction on $\mathbf{D}(\omega)$ before applying Bloch BCs. The dynamic reduction removes all interior DOFs from the system without any loss in accuracy. To perform a dynamic reduction, Eq. (2.33) is first partitioned into interior and boundary equations,

$$\begin{bmatrix} \mathbf{D}_{II} & \mathbf{D}_{IA} \\ \mathbf{D}_{AI} & \mathbf{D}_{AA} \end{bmatrix} \begin{Bmatrix} \mathbf{q}_I \\ \mathbf{q}_A \end{Bmatrix} = \begin{Bmatrix} \mathbf{0} \\ \mathbf{0} \end{Bmatrix}, \quad (2.45)$$

where the subscript I refers to interior DOFs and the subscript A refers to interface DOFs. Thus \mathbf{q}_A is a collection of all of the boundary DOF sets shown in Fig. 2.7. The upper half of this equation is solved to obtain an expression for \mathbf{q}_I ,

$$\mathbf{q}_I = -\mathbf{D}_{II}^{-1}\mathbf{D}_{IA}\mathbf{q}_A. \quad (2.46)$$

This allows for the description of the entire free DOF vector in terms of just the boundary DOFs,

$$\begin{Bmatrix} \mathbf{q}_I \\ \mathbf{q}_A \end{Bmatrix} = \begin{bmatrix} -\mathbf{D}_{II}^{-1}\mathbf{D}_{IA} \\ \mathbf{I} \end{bmatrix} \mathbf{q}_A. \quad (2.47)$$

Finally, $\mathbf{D}(\omega)$ is dynamically reduced using \mathbf{W} .

$$\mathcal{D}(\omega) = \mathbf{W}^T\mathbf{D}(\omega)\mathbf{W}. \quad (2.48)$$

Returning to Eq. (2.35), one can simply use $\mathcal{D}(\omega)$ instead of $\mathbf{D}(\omega)$. Note that since all the interior DOFs have been condensed out to the system, \mathbf{P} needs to be modified slightly by removing rows and columns corresponding to interior DOFs. Otherwise one can simply proceed as before.

The obvious benefit of using a dynamic reduction is that the size of the eigenvalue problem to be solved is greatly reduced without any sacrifice in accuracy. The main cost of the dynamic reduction is the system solution of Eq. (2.46) which must be computed at every frequency. Despite the cost of the dynamic reduction, it typically benefits the overall computational efficiency.

2.5.2.2 $\mathbf{k}(\omega)$ Solution Using the Bloch Operator Formulation

An alternative approach to obtaining the $\mathbf{k}(\omega)$ band structure also exists. If the free FE stiffness matrix is formed using the Bloch operator approach, it can be decomposed as follows,

$$\tilde{\mathbf{K}} = \tilde{\mathbf{K}}_0 + \tilde{\mathbf{K}}_x k_x + \tilde{\mathbf{K}}_y k_y + \tilde{\mathbf{K}}_z k_z + \tilde{\mathbf{K}}_{xx} k_x^2 + \tilde{\mathbf{K}}_{yy} k_y^2 + \tilde{\mathbf{K}}_{zz} k_z^2 + \tilde{\mathbf{K}}_{xy} k_x k_y + \tilde{\mathbf{K}}_{xz} k_x k_z + \tilde{\mathbf{K}}_{yz} k_y k_z. \quad (2.49)$$

The Bloch operator mass matrix on the other hand has no dependence on wave vector. In order to solve for the wave vector, the wave vector direction must be specified. This can be done by describing the wave vector in the following form,

$$\mathbf{k} = \begin{Bmatrix} k_x \\ k_y \\ k_z \end{Bmatrix} = k \begin{Bmatrix} c_x \\ c_y \\ c_z \end{Bmatrix}, \quad (2.50)$$

where c_x , c_y , and c_z are specified in order to define the direction of wave propagation, and satisfy $\sqrt{c_x^2 + c_y^2 + c_z^2} = 1$. Subbing this into the stiffness matrix definition gives

$$\tilde{\mathbf{K}} = \tilde{\mathbf{K}}_0 + k\tilde{\mathbf{K}}_1 + k^2\tilde{\mathbf{K}}_2, \quad (2.51)$$

where

$$\tilde{\mathbf{K}}_1 = \left(c_x \tilde{\mathbf{K}}_x + c_y \tilde{\mathbf{K}}_y + c_z \tilde{\mathbf{K}}_z \right), \quad (2.52)$$

$$\tilde{\mathbf{K}}_2 = \left(c_x^2 \tilde{\mathbf{K}}_{xx} + c_y^2 \tilde{\mathbf{K}}_{yy} + c_z^2 \tilde{\mathbf{K}}_{zz} + c_x c_y \tilde{\mathbf{K}}_{xy} + c_x c_z \tilde{\mathbf{K}}_{xz} + c_y c_z \tilde{\mathbf{K}}_{yz} \right). \quad (2.53)$$

Using this definition for the stiffness matrix, the periodic equations of motion are,

$$\tilde{\mathbf{P}}^\dagger \left(\tilde{\mathbf{K}}_0 - \omega^2 \mathbf{M} + k\tilde{\mathbf{K}}_1 + k^2\tilde{\mathbf{K}}_2 \right) \tilde{\mathbf{P}} \hat{\mathbf{q}} = \mathbf{0}. \quad (2.54)$$

This is a quadratic eigenvalue problem in k that can be solved using the linearization approach of Eq. (2.43).

The advantage of using the Bloch operator formulation for $\mathbf{k}(\omega)$ calculations is that this formulation leads to a quadratic eigenvalue problem for any direction of wave propagation [32]. This is a big advantage over the previously discussed $\mathbf{k}(\omega)$ method based on Bloch BCs that gives higher

order polynomial eigenvalue problems depending on the direction of wave propagation. Unfortunately the FE matrices obtained with the Bloch operator approach do not produce convergence as quickly as traditional FE matrices. This means that much finer meshes and or higher order elements are necessary to obtain equivalent accuracy. Furthermore, it is not possible to make use of a dynamic reduction using this formulation. Nevertheless, the ability to easily consider any direction of wave propagation makes this approach superior for obtaining band-structure surfaces over the entire BZ. The Bloch operator matrices also cannot be reduced using BMS, thus the remainder of this thesis will use models based on the Bloch BC formulation.

2.6 Damping in Band-Structure Calculations

Structures and materials typically have some mechanism for energy dissipation referred to as damping. Damping encompasses many different loss mechanisms including conversion to heat, sound, and/or transfer of kinetic energy to surrounding fluids for example. These are very complex phenomena to model, so damping in structural models is often treated by creating some parameterized model that can be fit to experimental data. Two common ways of modeling damping are to (1) specify a complex elasticity tensor so that damping is incorporated via complex terms in the stiffness matrix, and to (2) specify a damping matrix, \mathbf{C} that generates forces proportional to the nodal velocities[‡]. A brief discussion of the damping matrix approach is included here. The details of obtaining the damping matrix are neglected because there are so many different ways to do so. Rather, this section gives a brief introduction to damped band-structure calculations assuming that a damping matrix is given.

Including the damping matrix, the free equations of motion for a unit cell (once again neglecting any external forcing) are as follows,

$$\mathbf{K}\mathbf{q} + \mathbf{C}\dot{\mathbf{q}} + \mathbf{M}\ddot{\mathbf{q}} = \mathbf{0}. \quad (2.55)$$

[‡]Note that \mathbf{C} is repurposed here to denote the damping matrix rather than the elasticity tensor introduced in the FE derivation.

Applying Bloch BCs and assuming an exponential solution of the form $\mathbf{q} = \bar{\mathbf{q}}e^{\gamma t}$ yields the Bloch eigenvalue equations. Once again, to avoid excessive notation, the bar accent is dropped.

$$\mathbf{P}^\dagger (\mathbf{K} + \gamma\mathbf{C} + \gamma^2\mathbf{M}) \mathbf{P}\hat{\mathbf{q}} = \mathbf{0}. \quad (2.56)$$

As in the undamped case, one can specify the wave vector and solve for the damped frequencies or, one can specify the frequency and solve for the wave vector. The $\mathbf{k}(\omega)$ approach proceeds identically to the undamped case except that $\mathbf{D}(\omega) = \mathbf{K} + i\omega\mathbf{C} - \omega^2\mathbf{M}$ in Eq. (2.33).

Damping can manifest itself in two ways in wave propagation problems. It can result in complex frequencies indicating temporal dissipation, or it can result in complex wave vectors indicating spatial attenuation. Note that complex wave vectors can also arise in undamped materials, but this only occurs in band gaps. In damped materials, generally all waves vectors are complex so no perfect pass-band solutions exist.

In the undamped case, the eigenvalue solutions to the $\omega(\mathbf{k})$ band-structure calculation are real and match up exactly with the $\mathbf{k}(\omega)$ solutions (at least in the real wave number domain). For damped calculations however, the $\omega(\mathbf{k})$ solutions are complex whereas the $\mathbf{k}(\omega)$ solution requires that a real frequency be specified. This somewhat changes the physical meaning of the $\mathbf{k}(\omega)$ solution. Rather than being the true characteristic solution, the waves are now being forced or driven at a specific frequency. Thus in damped band-structures, the $\mathbf{k}(\omega)$ solution is commonly referred to as the driven-wave band structure whereas the $\omega(\mathbf{k})$ solution is referred to as the free-wave band structure [33].

This situation where either the frequency can be complex or the wave number can be complex, but not both, manifests itself due to the solution methods, not the physics of wave propagation. In principle, a full description of wave propagation would allow for both frequency *and* wave number to be simultaneously complex. With both spatial and temporal dissipation occurring simultaneously. Band structures with simultaneously complex frequencies and wave vectors can be computed, but this requires specialized algorithms and has only been demonstrated for 1D materials [34].

2.6.0.1 Damped $\omega(\mathbf{k})$ Band-Structure Calculations

The approach to solving the $\omega(\mathbf{k})$ problem is slightly more involved once damping is introduced. Given a real wave vector, Eq. (2.56) is a quadratic eigenvalue problem in γ which can be linearized as follows:

$$\left(\begin{bmatrix} \hat{\mathbf{K}} & \mathbf{0} \\ \mathbf{0} & \mathbf{J} \end{bmatrix} - \gamma \begin{bmatrix} -\hat{\mathbf{C}} & -\hat{\mathbf{M}} \\ \mathbf{J} & \mathbf{0} \end{bmatrix} \right) \begin{Bmatrix} \hat{\mathbf{q}} \\ \gamma \hat{\mathbf{q}} \end{Bmatrix} = \begin{Bmatrix} \mathbf{0} \\ \mathbf{0} \end{Bmatrix}. \quad (2.57)$$

The eigenvalue solution, γ , is generally complex and can be expressed in terms of the resonant frequency, ω^* and the damping ratio, ζ :

$$\gamma = -\zeta\omega^* \pm i\omega^*\sqrt{1-\zeta^2}. \quad (2.58)$$

From γ , one can obtain the damped frequency, $\omega_d = \omega^*\sqrt{1-\zeta^2} = |\text{imag}(\lambda)|$, and the damping ratio, $\zeta = -\text{real}(\lambda)/|\lambda|$.

2.6.0.2 Damped Band-Structure Numerical Example

The effects of damping on the band structure are best shown through an example calculation. As an example material, a Mindlin plate material is considered. The unit-cell mesh is shown in Fig. 2.12. The unit-cell geometry is described by the following parameters:

$$L_x = L_y = 10 \text{ cm}, \quad h = 1 \text{ mm},$$

where L_x and L_y are the lattice constants in the in-plane x and y directions, respectively, and h is the plate thickness. The undamped material properties are summarized as follows:

$$E = 2.4 \text{ GPa}, \quad \rho = 1040 \text{ kg/m}^3, \quad \nu = 0.33,$$

where E is the elastic modulus and ν is the Poisson's ratio. General viscous damping is introduced to the model by specifying a modified elastic modulus, $E_d = 2.25 \text{ GPa}$, and a modified Poisson ratio, $\nu_d = 0.25$. Using E_d and ν_d , a modified stiffness matrix \mathbf{K}_d is formed. A scaling parameter, q , is then used to form the damping matrix, $\mathbf{C} = q\mathbf{K}_d$. This process creates a general damping matrix (i.e., one that cannot be diagonalized by the undamped eigenvectors). The value of $q = 3 \times 10^{-4}$

used in this example is unrealistically large for most materials, but is used here so that the difference between the damped and undamped band structures is more noticeable.

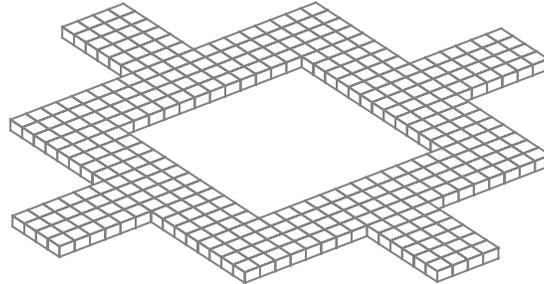


Figure 2.12: Plate lattice unit cell meshed with Mindlin elements.

The driven and free band structures for the damped plate lattice material are shown in Fig. 2.13. Also shown is the damping ratio band-structure obtained from the free-wave calculation. Compared to the undamped band-structure frequencies, the damped free-wave band-structure frequencies are lowered, but retain the same overall shape and pattern, with higher frequency branches being affected most strongly. The effect of damping on the driven-wave band structure is generally to smooth out features. This produces much different curves compared to the undamped case.

The remainder of this thesis will be focused on adapting CMS techniques from structural dynamics to material (i.e., wave propagation) problems. Damping has been incorporated in the context of CMS [35, 36], so incorporating it into BMS should follow a similar approach. For the remainder of this thesis however, damping will not be considered.

2.7 Computational Complexity of Algorithms

Since the focus of this thesis is computational efficiency, it is important to carefully select the eigenvalue algorithms to be used in obtaining the dispersion. A brief discussion of commonly used eigenvalue solution algorithms is included here. Eigenvalue solution algorithms can be categorized into two categories: direct solvers and iterative solvers. Direct solvers involve processing an entire

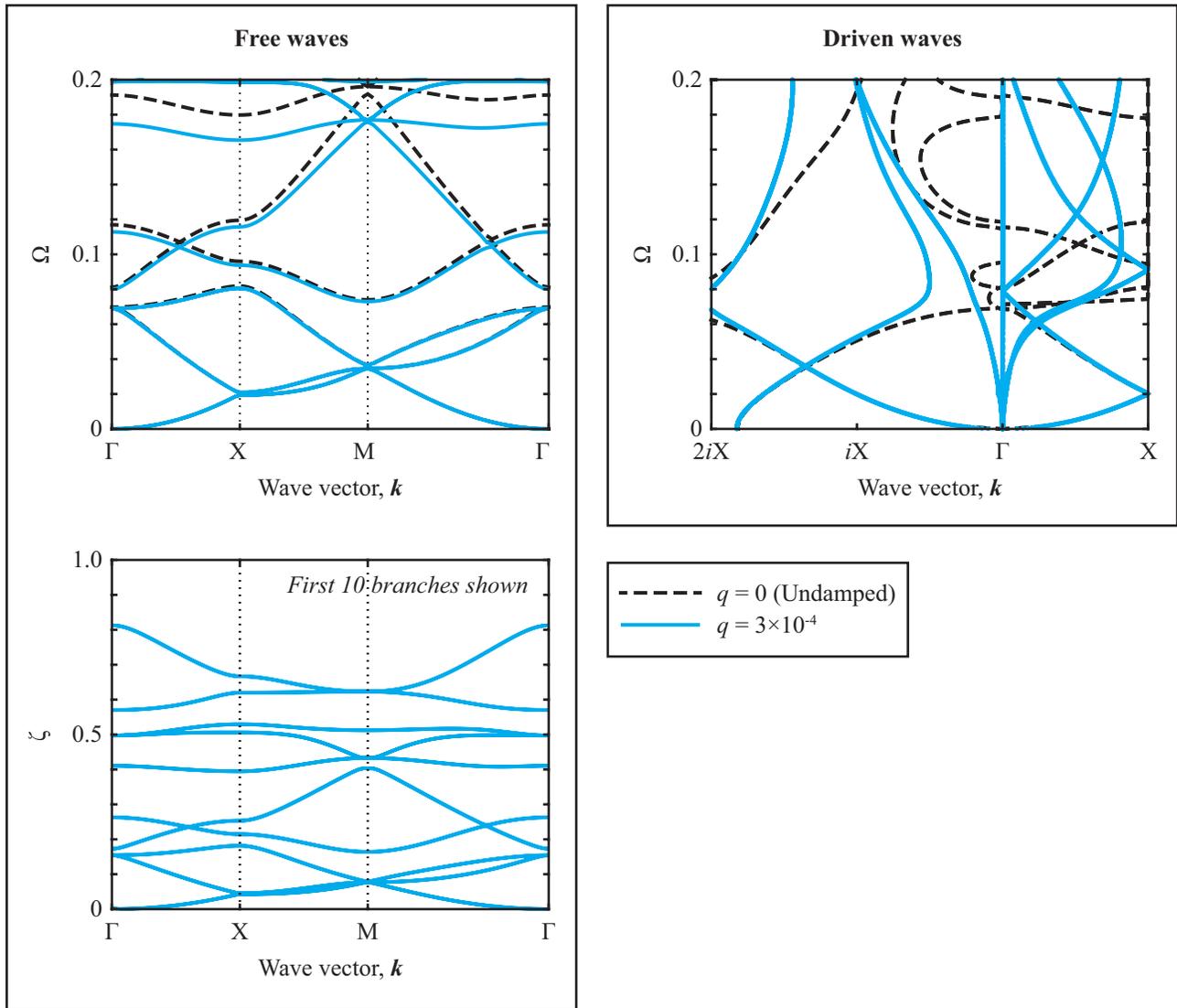


Figure 2.13: Plate lattice damped-frequency band structure (top left) and damping-ratio curves (bottom left) for free waves, and complex-wave vector band structure (top right) for driven waves. The frequencies are normalized as follows: $\Omega = \omega_d L_x \sqrt{\rho/E}$.

matrix to diagonalize it. These solvers are very stable and give all of the eigenvalue-eigenvector pairs of the system. The leading order complexity of direct solvers is typically $\mathcal{O}(n^3)$ as they are often based on matrix factorization. For relatively small matrices (less than about 1000×1000 entries), direct methods are very fast, but as n grows, the cubic-order complexity makes these algorithms computationally impractical.

Iterative solvers start with a small set of random vectors and iteratively refine the vectors until they converge to a set of eigenvectors. The computationally limiting step in every iteration is a small set of matrix vector products. This results in iterative methods having a leading order complexity of $\mathcal{O}(n^2)$ for full matrices and $\mathcal{O}(n)$ for sparse matrices. As mentioned earlier, both FE and FD methods produce sparse matrices. Although iterative methods only give a few eigenvalue eigenvector pairs, most applications only utilize the lowest frequency branches for the band structure anyway. As larger systems are considered (more than $10,000 \times 10,000$ entries) it is imperative to use both an iterative solver and to take advantage of the system sparsity.

In order to observe the complexity trends, eigenvalue calculations are timed for varying matrix sizes, and plotted on a set of log-log axes in Fig. 2.14. For comparison, both direct and iterative eigenvalue solutions are performed. Timing results for iterative solvers also depend on the number of eigen-pairs computed, k , so various values of k are used. The trend lines are added to show the leading order complexity. Thus they ignore initial values that do not appear to be linear.

In addition to eigenvalue solutions, the BMS reduction requires at least one (and maybe more) system solutions. A system solution must be performed to obtain constraint modes (which will be discussed in more detail in chapter 3). In many cases the constraint mode calculation represents the largest computational cost in forming the BMS reduced order model, particularly for very large 3D models. The size of the matrix to be solved is given by the number of DOFs that are in the interior of the unit-cell model domain, n_i which is proportional to the full-model size, n . The number of right-hand-side vectors is given by the number of DOFs on the boundary of the unit cell, n_b , which is proportional to $n^{1/2}$ for 2D models and $n^{2/3}$ for 3D models. The sparse supernodal cholesky algorithm (implemented within MATLAB's backslash operator) is used [37].

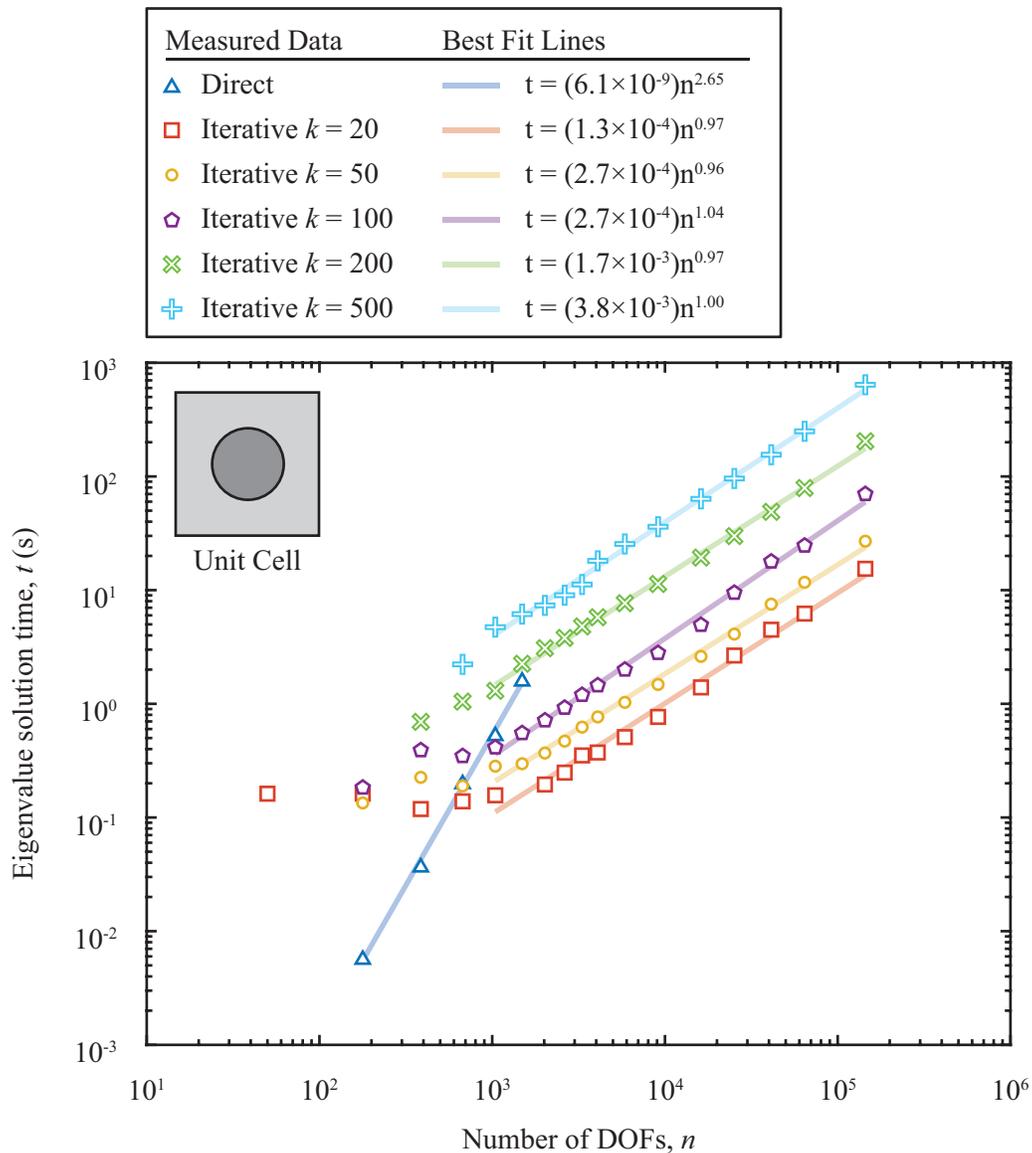


Figure 2.14: Computational complexity of iterative and direct eigenvalue solvers. The figure shows computation time versus matrix size for direct eigenvalue solution using MATLAB's `eig` function, and for iterative eigenvalue solution using MATLAB's `eigs` function. The system solution is performed for the hermitian system of equations shown in Eq. (3.9), where the matrices are obtained from an FE discretization of the circular-inclusion unit cell shown in the inset. The system size is varied by using increasingly refined FE models. The number of eigenvalue solutions, k , obtained from the iterative solver is varied to produce the different curves. The iterative solver takes advantage of the sparsity of the FE matrices.

This is a very efficient direct solver for symmetric positive definite systems that can take advantage of sparsity. For extremely large systems, it may be advantageous to use an iterative system solver,

however for the model sizes that I considered, iterative solvers did not perform as well as the direct approach. One reason for this is that iterative methods are not naturally well suited for system solutions with multiple right-hand-side vectors.

To observe the complexity trends for the direct system solution, the algorithm is timed for varying matrix sizes, and plotted on a set of log-log axes in Fig. 2.15. The number of right-hand-side vectors, p , is varied as well. Once again, trend lines are added to show the leading order complexity. For a fixed number of right-hand-side vectors, the computational order appears to be $\mathcal{O}(n)$. When the number of right-hand-side vectors grows with model size (as is the case for true constraint-mode calculations), the computational order is $\mathcal{O}(np)$ which is equivalent to $\mathcal{O}(n^{3/2})$ for the 2D model used here.

2.8 Summary

This chapter introduced the fundamental principals used in computing band-structure diagrams. The BZ is an important concept that was introduced. This reciprocal-space unit cell contains all unique solutions in wave-vector space and thus contains all information necessary to analyze wave propagation in crystals or periodic media in general. Another important concept described in this chapter, Bloch's theorem, allows for waves in periodic potentials to be represented as the product of a periodic field and a PW. This theorem allows periodic media that are approximated as infinite in space to be modeled by considering just a single unit cell. This chapter also presented the practical implementation of Bloch's theorem either via BCs or directly in the gradient terms of the governing equations. A comparison of these two variations shows that although they converge to the same result, Bloch BCs are usually favorable. Finally, an analysis of the computational complexity of some common linear algebra operations (direct eigenvalue solution, iterative eigenvalue solution, and direct linear system solution) was presented. Since these are common in band-structure calculations and also form the basic building blocks of the BMS reduced-order model, it is important to understand their performance and scaling.

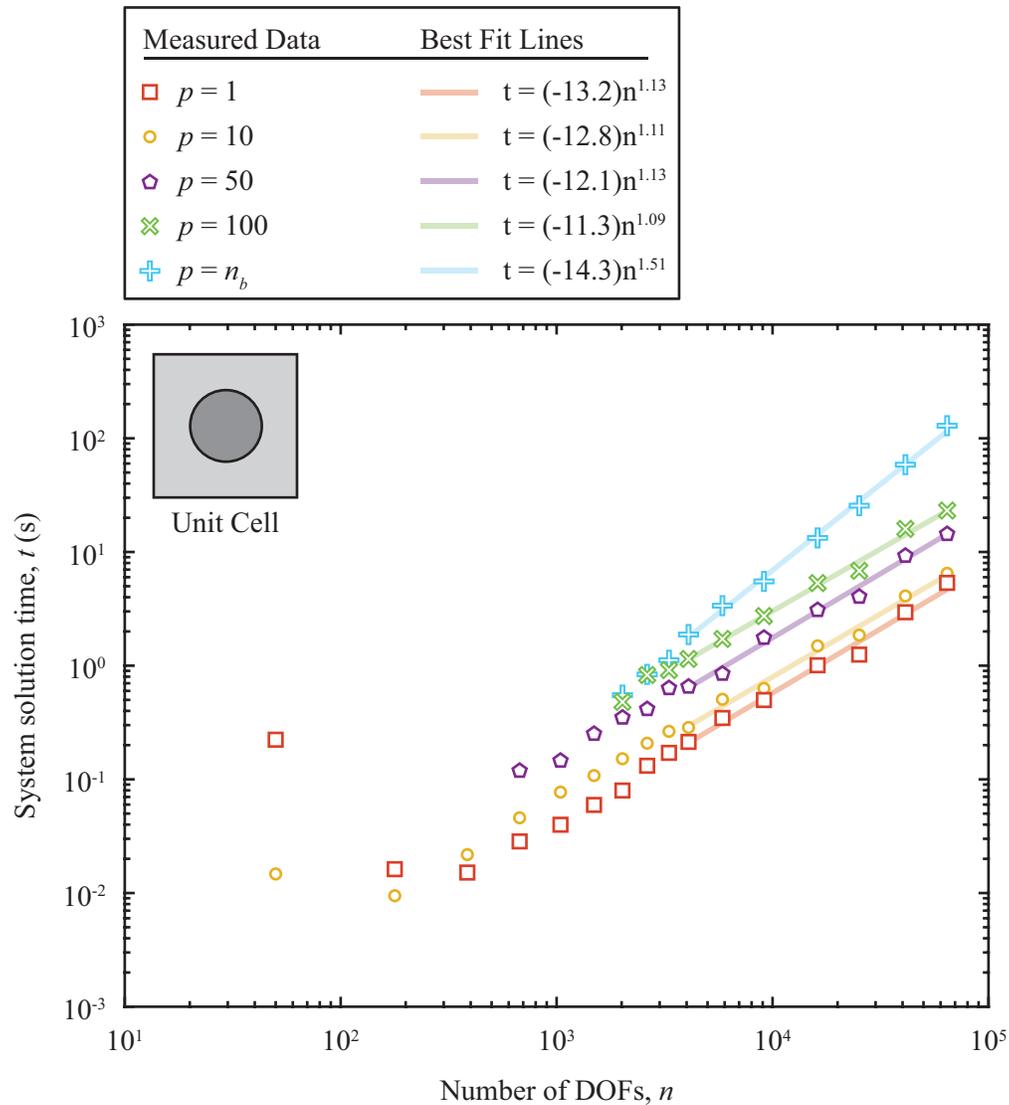


Figure 2.15: Computational complexity of system solver. The figure shows computation time versus matrix size for direct system solution using MATLAB's backslash operator. The system solution is performed for a hermitian matrix with multiple right-hand side vectors. The system solution being performed is shown in Eq. (3.8), where the matrices are obtained from an FE discretization of the circular-inclusion unit cell shown in the inset. The system size is varied by using increasingly refined FE models. The number of right-hand-side vectors, p , is varied from 1, to the total number of interface DOFs, n_b .

Chapter 3

Bloch Mode Synthesis: A Modal Reduction method for Band-Structure Calculations

3.1 Introduction

Modal reductions are powerful because they allow the dynamics of a system to be closely approximated using a few basis functions (mode shapes). Using a few of the lowest frequency modes as the basis functions allows the system behavior to be captured up to a certain frequency. The reduced model is typically accurate within a frequency range proportional to the frequency range of the modes used in the reduction. This reduced model is very useful for obtaining the frequency response function (FRF), which is defined as the response of the system to forcing at a particular frequency. The FRF is obtained by solving the equations of motion given some forcing $\mathbf{f} = \mathbf{F}e^{i\omega t}$,

$$(\mathbf{K} - \omega^2\mathbf{M}) \mathbf{q} = \mathbf{f} \quad (3.1)$$

The FRF, $\mathbf{H}(\omega)$, is the matrix that satisfies $\mathbf{H}(\omega)\mathbf{f} = \mathbf{q}$. Assuming the response has the same form as the forcing, $\mathbf{q} = \mathbf{Q}e^{i\omega t}$, the FRF can be found:

$$\mathbf{H}(\omega) = (\mathbf{K} - \omega^2\mathbf{M})^{-1}. \quad (3.2)$$

Thus, numerically solving for the FRF requires repeated solution of a system of equations. To lessen the computational burden, a modal reduction can be used. The displacements can be expanded in terms of a few mode shapes, $\mathbf{Q} \approx \mathbf{\Phi}\boldsymbol{\eta}$. Substituting into Eq. (3.1) and pre-multiplying by $\mathbf{\Phi}^T$ gives the reduced system of equations,

$$\mathbf{\Phi}^T (\mathbf{K} - \omega^2\mathbf{M}) \mathbf{\Phi}\boldsymbol{\eta} = \mathbf{\Phi}^T\mathbf{F}. \quad (3.3)$$

Assuming the mode shapes are mass normalized, this becomes,

$$(\mathbf{\Lambda} - \omega^2 \mathbf{I}) \boldsymbol{\eta} = \boldsymbol{\Phi}^T \mathbf{F}, \quad (3.4)$$

where $\mathbf{\Lambda}$ is a diagonal matrix containing the eigenvalues (natural frequencies squared) on its diagonal. The FRF is then easily obtained as,

$$\mathbf{H}(\omega) \approx \boldsymbol{\Phi} (\mathbf{\Lambda} - \omega^2 \mathbf{I})^{-1} \boldsymbol{\Phi}^T, \quad (3.5)$$

Since the matrix to be inverted is diagonal, computing the FRF is very simple. Also, recall that $\boldsymbol{\Phi}$ contains just a few of the low frequency mode shapes. Thus $\boldsymbol{\Phi}$ can be computed efficiently using an iterative solver.

The modal reduction's ability to diagonalize the equations of motion makes solution of the transient response to some initial conditions much simpler as well. It is much easier to solve n uncoupled (single DOF) ODEs than a system of n coupled ODEs.

Oftentimes the end goal of structural dynamics analysis is the mode shapes and frequencies themselves. In this case, modal reduction is not useful because obtaining the reduced model requires that the mode shapes be available in the first place. Thus, in structural dynamics, modal analysis is mainly useful for speeding up FRF calculations or forced-response calculations, where the response to a time varying force is computed. In the case of periodic materials however, a modal reduction *can* be useful even when the desired result is the wave mode shapes and frequencies. A modal reduction at some point in the BZ produces a model that can approximate the modes nearby in the BZ. This idea can be extended by combining modal bases from several points in the BZ. By doing so, the modal behavior of all or part of the BZ can be approximated very accurately. This idea was first applied to electronic-structure problems [17]. The approach was then extended to phononic and photonic systems and its performance improved by taking advantage of crystallographic symmetries * [18]. In order to apply these methods, one must select BZ points at which to compute the modal bases. These can be selected to be the high symmetry points [18], or they can be found using adaptive refinement [19].

*This is referred to as RBME

An alternative idea for modal reduction of the unit cell is to begin with the isolated unit-cell model which is free of any BCs, and to use a particular modal representation that does not affect the boundary DOFs. Since the boundary DOFs are unchanged, Bloch BCs can be applied in exactly the same way as they are for the full model. This idea forms the basis of BMS model reduction which constitutes the main contribution of this thesis.

Finding a reduced modal representation that leaves a subset of the DOF unchanged is not trivial. Fortunately this has been addressed in the context of structural dynamics. For structural dynamics analysis it may be computationally beneficial to divide a system into a set of substructures, perform modal reductions of the individual substructures, and finally recombine the reduced substructures to obtain a reduced model of the overall system. This process, called CMS, requires that the interface DOFs between substructures be unaffected by the modal reduction so that the substructures can be re-connected. Several modal representations have been developed in order to meet this requirement. Hurty reported in 1965 on the use of fixed interface (FI) modes paired with rigid-body and flexible constraint modes for substructure coupling [38]. Craig and Bampton in 1968 simplified the representation by noting that the rigid and elastic behavior can be conveniently captured in a single constraint mode set [39]. Following these seminal works, additional methods that use free-interface modes and attachment modes were developed in the 1970s [40, 41, 42].

This chapter demonstrates BMS using the Hurty/Craig-Bampton (HCB) model representation due to its simplicity and good performance. Very large reductions in structural dynamics models are achieved with this technique while still retaining great accuracy within some frequency range (see, e.g., Refs. [43, 44]). When the HCB reduction is utilized for BMS, the dynamics of the interior DOFs are described using some FI modes and coupling of the modes at the unit-cell boundaries is provided by a set of static constraint modes. Though it has not been investigated, in theory any model representation used for CMS should also be applicable for BMS. The flowchart in Fig. 3.1 shows the process by which a unit-cell model is reduced using the HCB representation, and then used to obtain the band structure.

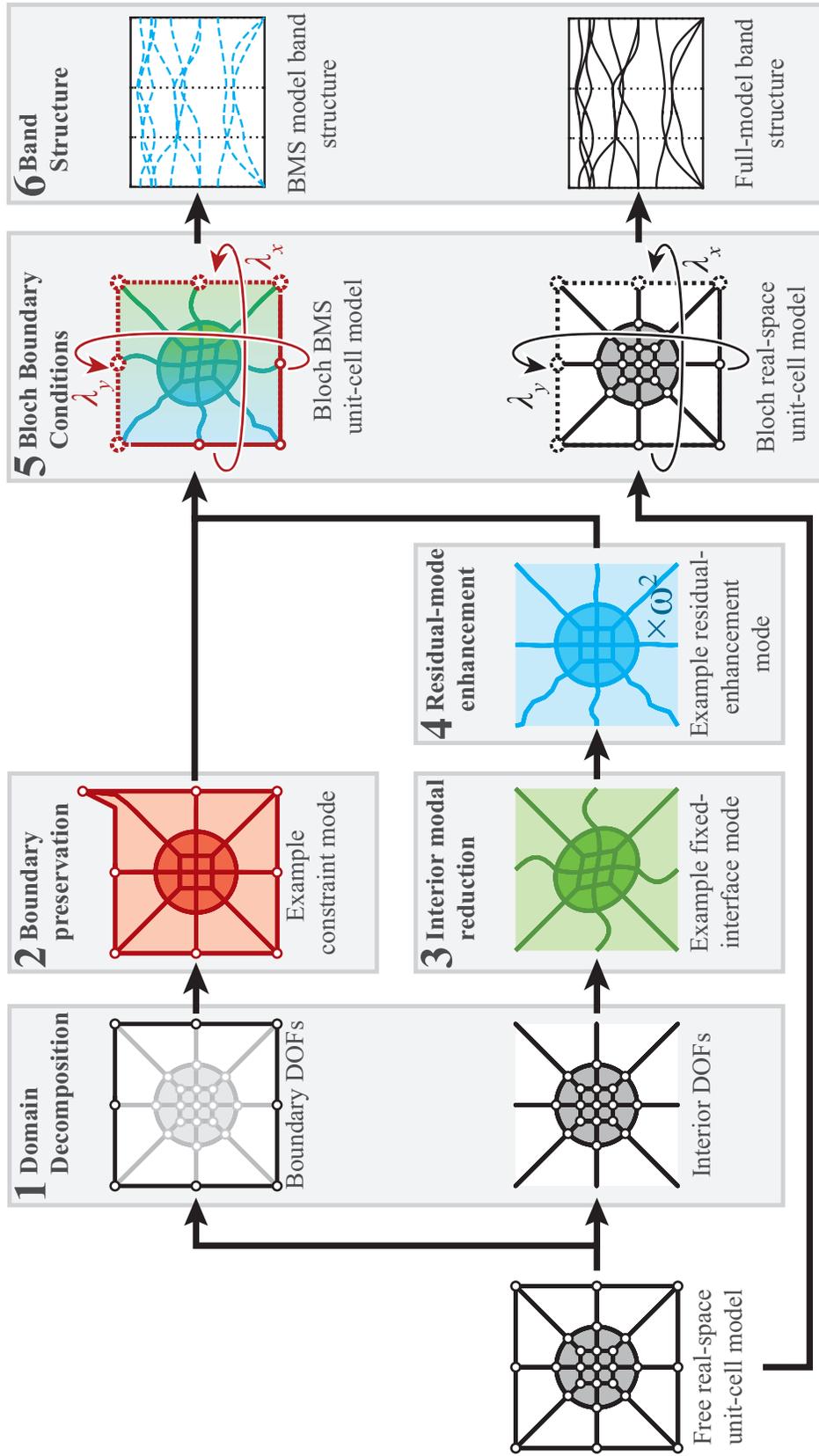


Figure 3.1: Flowchart illustrating the steps involved in using BMS to compute the band structure for a discretized unit-cell model

3.2 Formulation of Interior Reduction

3.2.1 Hurty/Craig-Bampton Substructure Representation

The HCB representation begins with the homogeneous form of the free equations of motion for the unit cell. These equations are referred to as free because no BCs are applied.

$$(\mathbf{K} - \omega^2 \mathbf{M}) \mathbf{q} = 0. \quad (3.6)$$

The free equations of motion are first split into interior and boundary partitions,

$$\left(\begin{bmatrix} \mathbf{K}_{II} & \mathbf{K}_{IA} \\ \mathbf{K}_{AI} & \mathbf{K}_{AA} \end{bmatrix} - \omega^2 \begin{bmatrix} \mathbf{M}_{II} & \mathbf{M}_{IA} \\ \mathbf{M}_{AI} & \mathbf{M}_{AA} \end{bmatrix} \right) \begin{Bmatrix} \mathbf{q}_I \\ \mathbf{q}_A \end{Bmatrix} = 0, \quad (3.7)$$

where the subscript I refers to interior DOFs and the subscript A refers to interface DOFs. Thus \mathbf{q}_A is a collection of all of the boundary DOF sets shown in Fig. 2.7.

Considering the top equation in Eq. (3.7) and setting all dynamic terms to zero (setting $\omega = 0$ and ignoring the mass matrix), it is possible to obtain a relationship between the interior and the interface sets.

$$\mathbf{q}_I = \Psi \mathbf{q}_A, \quad \text{where } \Psi = -\mathbf{K}_{II}^{-1} \mathbf{K}_{IA}. \quad (3.8)$$

Note that by themselves the constraint modes, Ψ , define a static condensation that is exact for static problems ($\omega = 0$), but that quickly loses accuracy as the frequency is increased.

A set of normal mode shapes, Φ_I , can be used to augment the constraint modes by improving the dynamic basis for the interior of the model. Considering the eigenvalue problem for just the interior partition of the equations of motion yields the interior normal modes,

$$(\mathbf{K}_{II} - \omega_I^2 \mathbf{M}_{II}) \phi_I = 0, \quad (3.9)$$

$$\Phi_I = \left[\{\phi_I\}_1 \quad \{\phi_I\}_2 \quad \cdots \quad \{\phi_I\}_{n_{\phi I}} \right], \quad (3.10)$$

where the number of interior mode shapes, $n_{\phi I}$, is much smaller than n_I , the number of interior DOFs. The interior mode shapes are referred to as FI mode shapes because they are the structural mode shapes that would result from clamping the boundaries of the free unit-cell structure.

The HCB transformation matrix, \mathbf{B} , is now formed by collecting the FI modes and the constraint modes together,

$$\underbrace{\begin{Bmatrix} \mathbf{q}_I \\ \mathbf{q}_A \end{Bmatrix}}_{\mathbf{q}} = \underbrace{\begin{bmatrix} \Phi_I & \Psi \\ \mathbf{0} & \mathbf{I} \end{bmatrix}}_{\mathbf{B}} \underbrace{\begin{Bmatrix} \boldsymbol{\eta}_I \\ \mathbf{q}_A \end{Bmatrix}}_{\mathbf{g}}, \quad (3.11)$$

where $\boldsymbol{\eta}_I$ is a set of modal DOFs corresponding to the set of FI modes. The constraint modes, Ψ , capture the influence of interface motion on the interior. The mixed DOF vector, \mathbf{g} , contains both modal and physical DOFs. Substituting Eq. (3.11) into (3.6) and premultiplying by \mathbf{B}^T gives the HCB reduced equations of motion,

$$(\boldsymbol{\kappa} - \omega^2 \boldsymbol{\mathcal{M}})\mathbf{g} = 0, \quad (3.12)$$

where

$$\boldsymbol{\mathcal{M}} = \mathbf{B}^T \mathbf{M} \mathbf{B}, \quad \boldsymbol{\kappa} = \mathbf{B}^T \mathbf{K} \mathbf{B}. \quad (3.13)$$

Note that Eq. (3.12) is a reduced set of equations that still represents the free unit cell because no BCs have been applied yet. In order to complete the BMS computation, Bloch BCs must be applied. This can be done using the Bloch periodicity transformation matrix defined in Eq. (2.20). The reduced model contains modal DOFs rather than physical DOFs in the interior. Thus, the rows and columns of \mathbf{P} corresponding to \mathbf{q}_I must be updated to reflect the size of $\boldsymbol{\eta}_I$.

Although damping has not been considered in the above formulation, A few efforts have been made to extend CMS techniques to systems with general viscous damping [35, 36]. The extension of these methods from the CMS context to BMS is relatively straightforward. Application of BMS to damped systems will not be discussed further here, but may be the subject of future work.

3.2.2 Approximating Residual-Mode Contribution for Enhanced Accuracy

The HCB representation is exact when all modes are kept in the transformation. Of course this is not practical because no reduction in model size is obtained. It has been shown however that an approximation of the residual-mode contributions can lead to improved accuracy [45].

A recent extension of this technique obtains this improved accuracy without any increase in the reduced-model size [46, 47].

Rather than simply truncating the residual-mode set, the contribution of the residual modes can be approximated by modifying the HCB transformation. First, the residual modes are (temporarily) added into Eq. (3.11) so that the transformation represents a change of coordinates but not a model reduction,

$$\begin{Bmatrix} \mathbf{q}_I \\ \mathbf{q}_A \end{Bmatrix} \begin{bmatrix} \Phi_I^d & \Phi_I^r & \Psi \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{Bmatrix} \eta_I^d \\ \eta_I^r \\ \mathbf{q}_A \end{Bmatrix}, \quad (3.14)$$

where the superscript d denotes dominant modes that are kept in the original HCB representation, and the superscript r denotes residual modes that are truncated in the original HCB representation.

Applying this transformation to the Eq. (3.7) gives,

$$\left(\begin{bmatrix} \Lambda^d & \mathbf{0} & \mathbf{0} \\ & \Lambda^r & \mathbf{0} \\ sym & & \mathcal{K}_{AA} \end{bmatrix} - \omega^2 \begin{bmatrix} \mathbf{I} & \mathbf{0} & \{\Phi_I^d\}^T \mathbf{M}_c \\ & \mathbf{I} & \{\Phi_I^r\}^T \mathbf{M}_c \\ sym & & \mathcal{M}_{AA} \end{bmatrix} \right) \begin{Bmatrix} \eta_I^d \\ \eta_I^r \\ \mathbf{q}_A \end{Bmatrix} = \begin{Bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{Bmatrix}, \quad (3.15)$$

where

$$\mathbf{M}_c = (\mathbf{M}_{IA} + \mathbf{M}_{II}\Psi), \quad (3.16)$$

and Λ^d and Λ^r respectively contain the dominant and residual eigenvalues on their diagonals.

Considering just the second row of Eq. (3.15), it is possible to solve for η_I^r ,

$$\eta_I^r = (\Lambda^r - \omega^2 \mathbf{I})^{-1} \{\Phi_I^r\}^T \mathbf{M}_c \eta_A. \quad (3.17)$$

Substituting this back into the first row of Eq. (3.14) gives,

$$\mathbf{q}_I = \Phi_I^d \eta_I^d + (\Psi \mathbf{L} + \omega^2 \mathbf{F}_r \mathbf{M}_c) \eta_A, \quad (3.18)$$

with

$$\mathbf{F}_r = \Phi_I^r (\Lambda^r - \omega^2 \mathbf{I})^{-1} \{\Phi_I^r\}^T, \quad (3.19)$$

where \mathbf{F}_r is the residual-flexibility matrix. Assuming the residual frequencies are much higher than the highest frequency of interest, ω , we can approximate the residual flexibility as,

$$\mathbf{F}_r \approx \underbrace{\Phi_I^r \{\Lambda^r\}^{-1} \{\Phi_I^r\}^T}_{\mathbf{F}_{rs}} + \omega^2 \underbrace{\Phi_I^r \{\Lambda^r\}^{-2} \{\Phi_I^r\}^T}_{\mathbf{F}_{rm}}. \quad (3.20)$$

Substituting back into Eq. (3.18) and keeping only first order terms in ω^2 gives,

$$\mathbf{q}_I = \Phi_I^d \boldsymbol{\eta}_I^d + (\Psi \mathbf{L} + \omega^2 \mathbf{F}_{rs} \mathbf{M}_c) \mathbf{q}_A. \quad (3.21)$$

It is not practical to compute the static residual flexibility, \mathbf{F}_{rs} using the residual modes, because these modes are not computed in the original HCB representation. Computing them would dramatically increase the computational cost of the reduction. Fortunately, the static residual flexibility can be expressed in terms of the dominant modes as follows,

$$\mathbf{F}_{rs} = \mathbf{K}_{II}^{-1} - \Phi_I^d \{\Lambda^d\}^{-1} \{\Phi_I^d\}^T. \quad (3.22)$$

At this point, Eq. (3.14) can be rewritten in terms of $\boldsymbol{\eta}_I^d$ and \mathbf{q}_A ,

$$\underbrace{\begin{Bmatrix} \mathbf{q} \\ \mathbf{q}_I \\ \mathbf{q}_A \end{Bmatrix}}_{\mathbf{B}_e} = \underbrace{\left(\begin{array}{c|c} \mathbf{B} & \mathbf{B}_r \\ \hline \begin{bmatrix} \Phi_I^d & \Psi \\ \mathbf{0} & \mathbf{I} \end{bmatrix} & \begin{bmatrix} \mathbf{0} & \mathbf{F}_{rs} \mathbf{M}_c \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \end{array} \right)}_{\mathbf{B}_e} + \omega^2 \begin{Bmatrix} \mathbf{g} \\ \boldsymbol{\eta}_I^d \\ \mathbf{q}_A \end{Bmatrix}. \quad (3.23)$$

If the goal is to calculate the $\mathbf{k}(\omega)$ band structure, then forming \mathbf{B}_e is straightforward because ω is known. For the $\omega(\mathbf{k})$ formulation however, ω^2 must be approximated. Since ω^2 changes as a function of the wave vector, a slightly different transformation must be used for each wave vector. The modification performed at every wave vector is computationally cheap however so it does not affect the performance. First, \mathbf{B} is replaced with \mathbf{B}_e in Eq. (3.13),

$$\mathcal{M}_e = \mathbf{B}_e^T \mathbf{M} \mathbf{B}_e, \quad \mathcal{K}_e = \mathbf{B}_e^T \mathbf{K} \mathbf{B}_e. \quad (3.24)$$

Expanding this product gives,

$$\mathcal{K}_e = \underbrace{\mathbf{B}^T \mathbf{K} \mathbf{B}}_{\mathcal{K}_0} + \omega^2 \left(\underbrace{\mathbf{B}^T \mathbf{K} \mathbf{B}_r}_{\mathcal{K}_2} + \underbrace{\mathbf{B}_r^T \mathbf{K} \mathbf{B}}_{\mathcal{K}_2^T} \right) + \omega^4 \underbrace{\mathbf{B}_r^T \mathbf{K} \mathbf{B}_r}_{\mathcal{K}_4}, \quad (3.25)$$

$$\mathcal{M}_e = \underbrace{\mathbf{B}^T \mathbf{M} \mathbf{B}}_{\mathcal{M}_0} + \omega^2 \left(\underbrace{\mathbf{B}^T \mathbf{M} \mathbf{B}_r}_{\mathcal{M}_2} + \underbrace{\mathbf{B}_r^T \mathbf{M} \mathbf{B}}_{\mathcal{M}_2^T} \right) + \omega^4 \underbrace{\mathbf{B}_r^T \mathbf{M} \mathbf{B}_r}_{\mathcal{M}_4}. \quad (3.26)$$

Pre- and post-multiplying \mathcal{K}_e and \mathcal{M}_e by \mathbf{P} to enforce Bloch BCs gives

$$\hat{\mathcal{K}}_e = \hat{\mathcal{K}}_0 + \omega^2 \left(\hat{\mathcal{K}}_2 + \hat{\mathcal{K}}_2^T \right) + \omega^4 \hat{\mathcal{K}}_4, \quad (3.27)$$

$$\hat{\mathcal{M}}_e = \hat{\mathcal{M}}_0 + \omega^2 \left(\hat{\mathcal{M}}_2 + \hat{\mathcal{M}}_2^T \right) + \omega^4 \hat{\mathcal{M}}_4. \quad (3.28)$$

To remove the dependence of $\hat{\mathcal{K}}_e$ and $\hat{\mathcal{M}}_e$ on ω^2 , consider briefly the reduced-order characteristic equation with no enhancement terms, after enforcement of periodic BCs,

$$\left(\hat{\mathcal{K}}_0 - \omega^2 \hat{\mathcal{M}}_0 \right) \hat{\eta} = \mathbf{0}. \quad (3.29)$$

This can be rearranged and solved to give,

$$\omega^2 \hat{\eta} = \underbrace{(\hat{\mathcal{M}}_0)^{-1} \hat{\mathcal{K}}_0}_{\mathbf{Y}} \hat{\eta}. \quad (3.30)$$

This substitution is referred to in [46, 48] as O'Callahan's approach and further details may be found there. Note that obtaining \mathbf{Y} is computationally cheap because $\hat{\mathcal{K}}_0$ and $\hat{\mathcal{M}}_0$ are small relative to the full model. Substituting the previous result and the Bloch periodicity equation into Eq. (3.23) gives,

$$\mathbf{q} = (\mathbf{B} + \omega^2 \mathbf{B}_e) \mathbf{P} \hat{\eta} = \underbrace{(\mathbf{B} \mathbf{P} + \mathbf{B}_e \mathbf{P} \mathbf{Y})}_{\mathbf{Q}} \hat{\eta}. \quad (3.31)$$

Finally, the residual-mode enhanced matrices are obtained by pre- and post-multiplying the free mass and stiffness matrices by the composite transformation \mathbf{Z} ,

$$\hat{\mathcal{K}}_e = \mathbf{Z}^T \mathbf{K} \mathbf{Z} = \hat{\mathcal{K}}_0 + \left(\hat{\mathcal{K}}_2 \mathbf{Y} + \mathbf{Y}^T \hat{\mathcal{K}}_2^T \right) + \mathbf{Y}^T \hat{\mathcal{K}}_4 \mathbf{Y}, \quad (3.32)$$

$$\hat{\mathcal{M}}_e = \mathbf{Z}^T \mathbf{M} \mathbf{Z} = \hat{\mathcal{M}}_0 + \left(\hat{\mathcal{M}}_2 \mathbf{Y} + \mathbf{Y}^T \hat{\mathcal{M}}_2^T \right) + \mathbf{Y}^T \hat{\mathcal{M}}_4 \mathbf{Y}. \quad (3.33)$$

Finally, the band-structure frequencies are obtained by solving the residual-enhanced BMS eigenvalue problem,

$$\left(\hat{\mathcal{K}}_e - \omega^2 \hat{\mathcal{M}}_e \right) \hat{\mathbf{g}} = \mathbf{0}. \quad (3.34)$$

The approximation of the residual-mode contribution allows for significant improvements in accuracy. This allows band-structure error to be decreased with just a small trade off in additional computation time. Note that the residual-mode correction provides an enhancement to the interior modes but not to the interface modes. A similar enhancement for local interface modal reduction may be possible but has not currently been developed. In the present work, when the residual-enhanced technique is used to improve the accuracy of the HCB reduction of the interior, then a larger number of boundary modes most likely needs to be used so that the error in the boundary-mode reduction does not dominate the overall error.

The next section describes automated multi-level substructuring (AMLS). The residual-enhanced approach was first developed in the context of HCB reductions, but was shown to be useful for AMLS models as well [48]. To provide a shorthand notation, residual-enhancement is indicated by appending a “+” to the interior reduction type, i.e.: residual-enhanced HCB (HCB+), and residual-enhanced AMLS (AMLS+).

3.2.3 Automated Multi Level Substructuring

For very large unit-cell models, the up-front cost of the HCB reduction can become very large. This is mainly due to the eigenvalue solution of Eq. (3.9), the system solution required by Eq. (3.8), and the matrix-matrix products required by Eq. (3.13). AMLS was developed in order to speed up some of these calculations [49, 50].

The main idea is to split the unit cell up into smaller sub-structures, perform separate reductions on all of the subcomponents, and then combine the sub-components into a single reduced model. This idea is illustrated in Fig. 3.2 which shows the unit cell of Fig. 1.1 being subdivided into several levels of substructures. Note that in a typical substructuring framework, the boundaries are included in each substructure. This results in redundant boundary DOFs between substructures that will be reconnected. This approach is straightforward, but is not as efficient (especially when the number of substructures is very high). The AMLS method formalizes a procedure by which the substructure boundaries are treated as their own independent substructures. Thus the boundaries

must only be considered once, rather than once as a boundary and again as the interior of a higher level substructure.

All of the substructures that will be considered in the structure can be shown compactly in Fig. 3.3a. Note that intermediate substructures are not included because they are never explicitly considered in the AMLS procedure. The process begins with the lowest level substructures (level d in Fig. 3.3b) and computes their interior modes, as well as their contribution to the substructure mass and stiffness matrices in the levels above them. This repeats until the top level substructure has been reached. A detailed description of the implementation can be found in [49]. For a unit-cell analysis, the top level substructure must always contain all of the boundary DOFs. This allows Bloch BCs to be applied as usual.

In order to make AMLS truly automated, the partitioning into substructures must be automated as well. There is a vast body of literature on graph partitioning that can be adapted to this purpose. For this analysis, spectral partitioning [51] is used. This technique relies on the fact that the second lowest eigenvector (the Fiedler vector) of the adjacency matrix provides a natural splitting of a graph or mesh. This technique was chosen due to its elegant simplicity, but it does have a few drawbacks compared to more advanced partitioning schemes.

- (1) For very large graphs, it can be computationally burdensome to obtain the Fiedler vector, especially when the graph has a very high degree of connectivity leading to a relatively dense adjacency matrix. This is not a problem for FE meshes as the adjacency matrix is typically very sparse.
- (2) A good quality partition typically minimizes the number of boundary vertices (nodes) by minimizing the number of edges (connections between nodes) that must be cut, but also produces partitions that are roughly equal in size. The Fiedler vector tends to produce two equal sized partitions with no consideration given to the number of edge cuts made, so this consideration must be built into the algorithm separately.
- (3) Symmetric graphs/meshes can lead to degeneracy of the Fiedler vector. This leads to

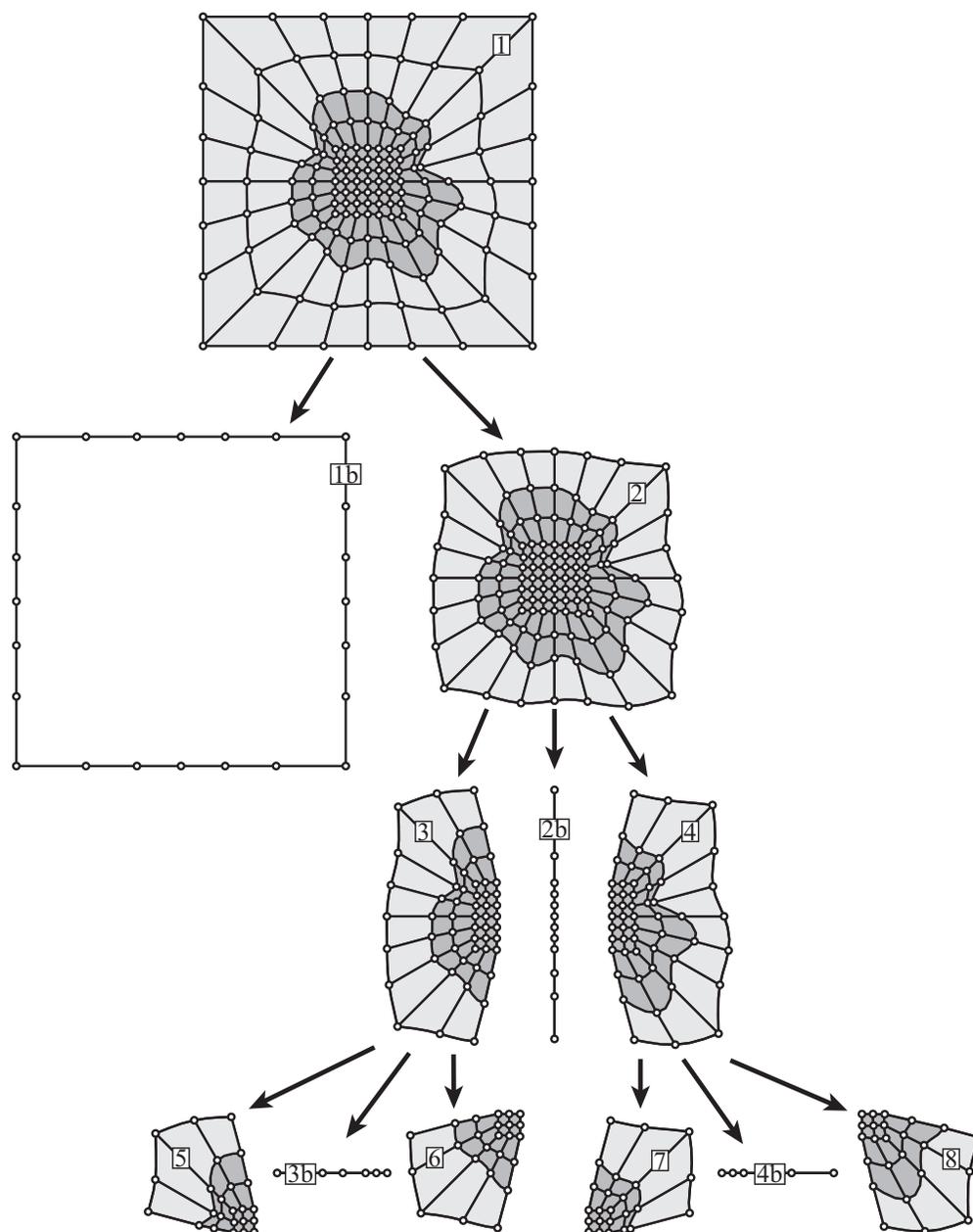


Figure 3.2: Tree diagram showing the partitioning of a free unit cell for AMLS analysis

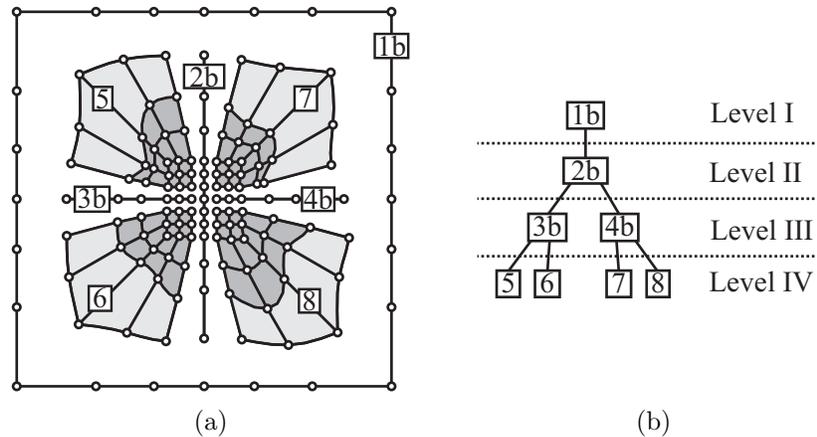


Figure 3.3: Compact representation of AMLS Substructures (a) where intermediate substructures have been removed, and substructure tree (b)

partitions that are not deterministic, and that tend to ignore the symmetry of the graph. For unit cells, symmetry axes often align with the lattice vectors, so the partitions can be encouraged to align with lattice vectors in order to produce better partitions.

- (4) Some partitioning algorithms can split a graph into any number of partitions (rather than just two). This may be more natural depending on the graph.

One additional consideration when partitioning FE models regardless of the graph partitioning algorithm is that the mesh should be split along element boundaries (i.e. elements should not be split up). Despite the drawbacks listed above, the spectral partition works reasonably well for all unit cells considered thus far.

3.3 Numerical Examples

In this section some results are demonstrated using a 2D material with an eight-lobed inclusion in every unit cell. The performance of the BMS reduced model is measured by the error relative to the full model, and the computational time relative to the full model. The frequency error is defined as the deviation from the nominal frequency divided by the nominal frequency. This definition does not work well when the nominal frequency is zero, so zero-frequency points are

ignored in frequency error discussions. The error between two mode vectors, ϕ_A and ϕ_B is defined as,

$$e_\phi = 1 - \frac{|\langle \phi_A | \phi_B \rangle|}{\|\phi_B\| \|\phi_B\|}, \quad (3.35)$$

where $|\cdot|$ denotes the complex modulus, $\langle \cdot | \cdot \rangle$ denotes the dot product operator, and $\|\cdot\|$ denotes the 2 norm of a vector.

The computational speedup is described by the reduced-model computation time normalized by the full-model computation time. The full-model computation has a computation time per point (per \mathbf{k} point or per frequency point depending on the method of band-structure calculation). When multiplied by the number of points used in the computation the total computation time is obtained for the full model,

$$t_{full} = n_p \times t_{full}^p. \quad (3.36)$$

Reduced-model computations have a per-point computation time, but additionally they have an up-front computation time associated with the time necessary to perform the reduction itself,

$$t_{red} = n_p \times t_{red}^p + t_{red}^u. \quad (3.37)$$

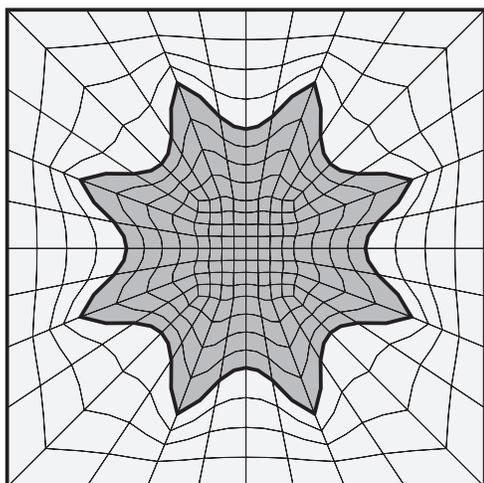
Since the up-front time for the reduced-order model is amortized over all of the \mathbf{k} points, BMS is more worthwhile if a large number of \mathbf{k} points must be considered in the band-structure calculation.

The material considered for the examples in this section has a 2D unit cell with an eight-lobed inclusion. The inclusion is shaped to resemble the Helicoid-Catenoid logo of the Phononics conference. The unit cell is modeled using 2D plane-strain elements. The varying mesh densities shown in Fig. 3.4 are used to understand the impact of model size on the effectiveness of the reduction.

3.3.1 Band-Structure Calculation using $\omega(\mathbf{k})$ Method

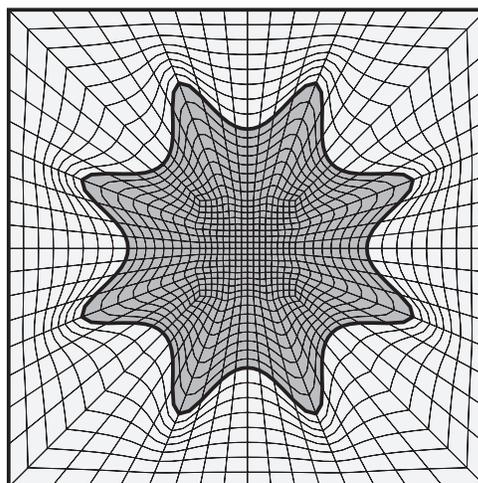
Using the $\omega(\mathbf{k})$ approach, the first 10 dispersion branches of the medium sized model are computed for 97 \mathbf{k} points. The model is then reduced from 10240 to 226 DOFs (30 interior

Small Mesh
320 Elements, 2560 DOF



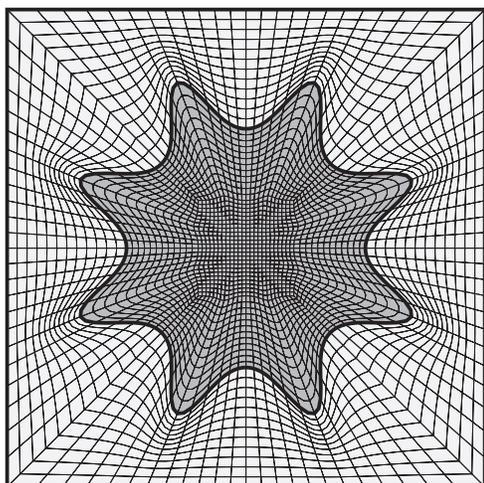
(a)

Medium Mesh
1280 Elements, 10240 DOF



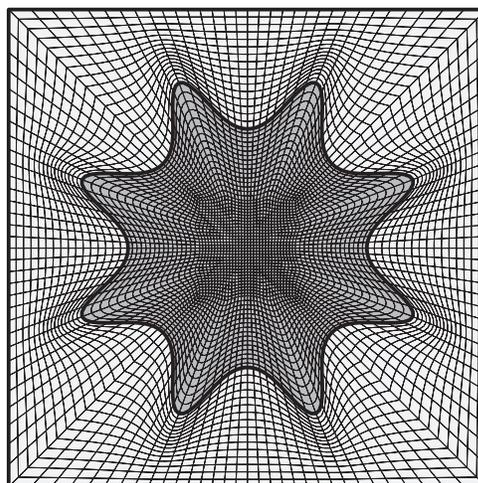
(b)

Large Mesh
2880 Elements, 23040 DOF



(c)

Reference Mesh
5120 Elements, 40960 DOF



(d)

Figure 3.4: 2D unit-cell meshes with eight-lobed inclusion (resembling the Helicoid-Catenoid logo used for the Phonics 20XX conference series).

modes and 190 boundary DOFs), and again the band structure is computed. Figure 3.5a shows a comparison of the full- and reduced-model dispersion diagrams, and indicates very good visual agreement in the dispersion frequencies. The maximum error over the first 10 branches is less than 0.015%. The reduced-model computation time is just 5.6 seconds compared to 170.1 seconds for the

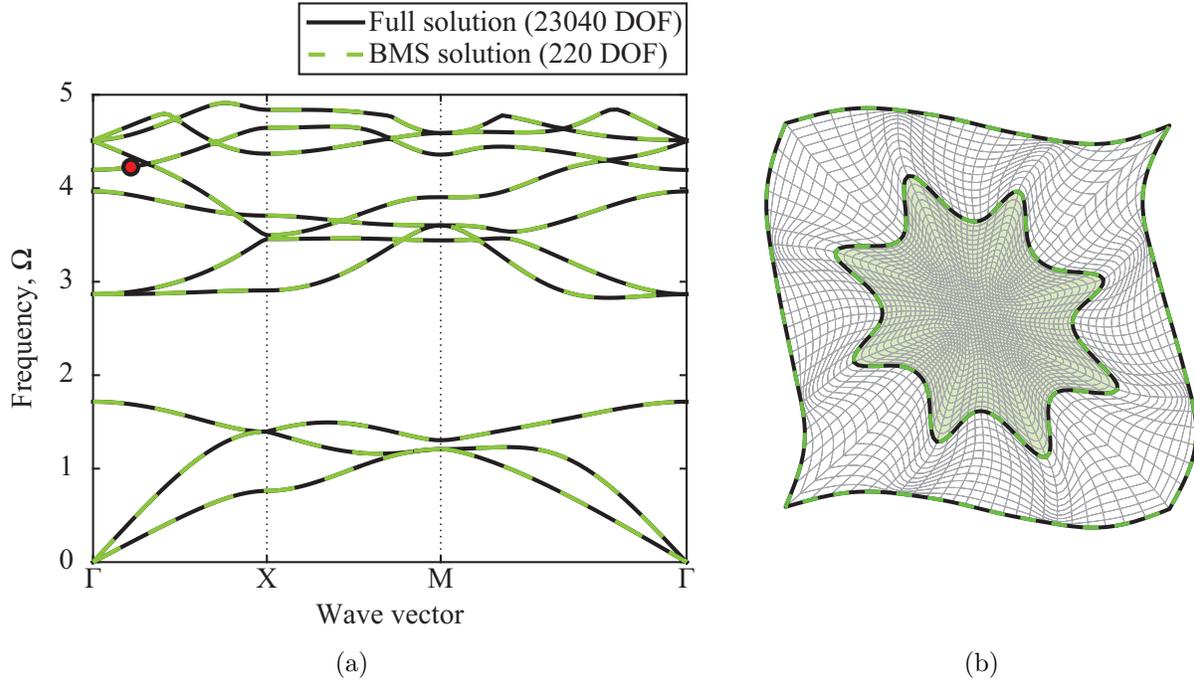


Figure 3.5: (a) Dispersion diagram comparison for eight-lobed inclusion material where $\Omega = \omega L_x \sqrt{\rho/E}$ is the non-dimensional frequency, and (b) selected mode shape comparison for the mode highlighted in the dispersion diagram.

full model. This represents a speedup by a factor of nearly 30 times. The reduced computational time includes approximately 4 seconds of up-front calculations, so per- \mathbf{k} -point computations are over 100 times faster than the full model. Figure 3.5b shows the reduced-model mode shape plotted on top of the full model mode shape for the point highlighted on the dispersion plot. The mode shapes are visually indistinguishable and the error between them is $e_\phi = 1.1 \times 10^{-6}$. The maximum mode error for the band structure shown in Fig. 3.5a is $e_\phi = 7.3 \times 10^{-5}$. This maximum mode error occurs at points where the band structure is degenerate or nearly degenerate.

Note that the eigenvectors computed with reduced models do not represent physical coordinates. Rather the eigenvectors computed with the reduced model are in a reduced modal space. To convert the reduced-space mode shape, ϕ_{Red} , into a physical space mode, ϕ_{Phys} , requires a transformation back to physical coordinates. This is easily accomplished using the HCB transformation

matrix.

$$\phi_{\text{Phys}} = \mathbf{B}\phi_{\text{Red}} \quad (3.38)$$

3.3.2 Discretization Error

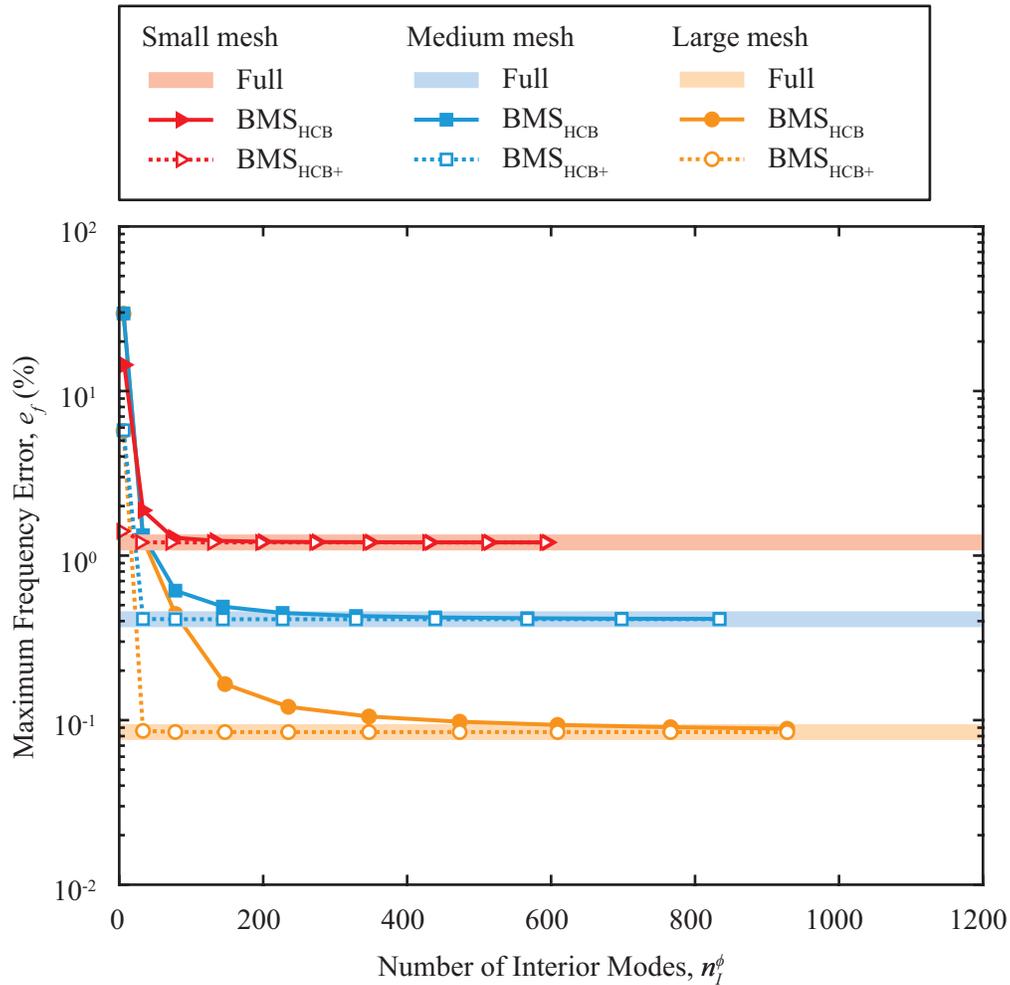


Figure 3.6: Discretization error and BMS reduced-model error for small, medium, and large meshes with respect to the reference mesh. The BMS models are formed using the HCB, and HCB+ reductions of the interior, and no reduction of the interface.

When studying the error due to model reduction, it is important to also consider discretization error. Discretization error arises due to the inability of a discrete model to capture the behavior (and sometimes geometry) of a continuous structure. Mesh refinement can be used to decrease the discretization error and get closer and closer to the continuous model solution. Reduction error

occurs due to the inability of a reduced modal space to capture the exact dynamics of the full mesh. Reduction error decreases as more modes are used in the model. For BMS models, this error decreases as the number of FI modes is increased.

The overall error in a BMS reduced-order model consists of discretization error and reduction error. One can go to great lengths to make sure that the error due to the model reduction is very small, but there isn't much benefit if the discretization error is comparatively large. This is illustrated in Fig. 3.6. The dashed lines show the maximum frequency error for the small, medium, and large meshes with respect to the reference mesh. This is a good approximation to the discretization error in the three meshes because the reference mesh is highly converged compared to the smaller meshes. The solid lines show the overall frequency error for several BMS reductions of the small, medium, and large meshes, with increasing number of FI modes. The error due to the reduction quickly becomes small compared to the discretization error, at which point the benefit of adding modes to the reduction is marginal. It is also easy to see that reduced versions of denser meshes can have much lower error than the smaller full models.

3.3.3 Performance Comparison: Automated Multi-Level Substructuring Versus Hurty/Craig-Bampton Substructuring

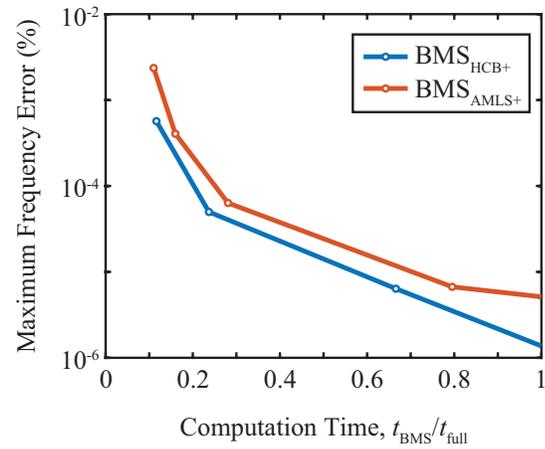
The AMLS framework introduces some additional error over traditional HCB representations. The eight-lobed inclusion material is used to evaluate this additional error. The small, medium, and large unit-cell meshes from Fig. 3.4 are partitioned into substructures to prepare for AMLS. Each mesh is repeatedly subdivided until every partition has fewer than four-hundred DOFs (133 nodes). The resulting partitioned meshes are shown in Fig. 3.7.

For each mesh, several AMLS+ models with varying number of FI modes are created and the resulting band structures are computed. The band-structure calculation computes the 10 lowest branches at 97 discrete \mathbf{k} points along the IBZ edge. The number of FI modes kept in each AMLS+ substructure is varied by changing a cutoff frequency (all modes under the cutoff frequency are kept) from 3 to 10 times the highest band-structure frequency. As the cutoff frequency and hence the



2,560 DOF

(a)

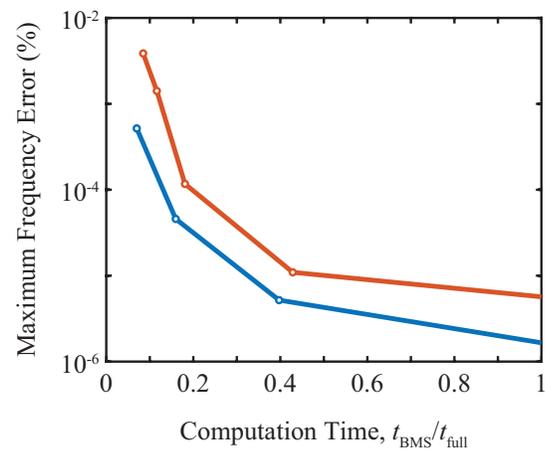


(b)



10,240 DOF

(c)

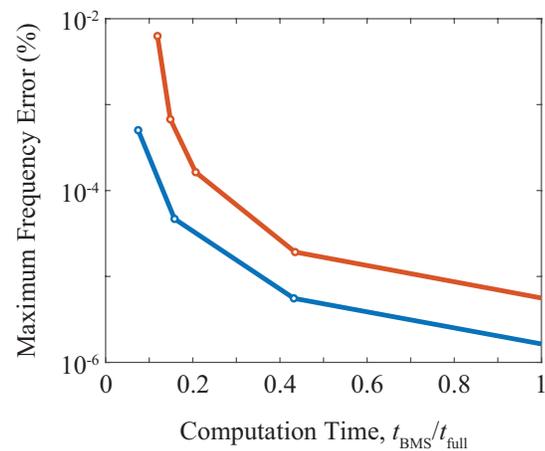


(d)



23,040 DOF

(e)



(f)

Figure 3.7: Eight-lobed inclusion model substructure partitions (left) and AMLS maximum frequency error versus computational time fraction (right) for small (top), medium (middle), and large (bottom) meshes from Fig. 3.4.

number of modes is increased, the AMLS+ computation time increases and the error decreases. The resulting performance curves are plotted in Fig. 3.7 for the small, medium, and large meshes. The performance curves show maximum frequency error over the calculated band structure versus computation time (normalized by the full-model computation time). The error is measured relative to each reduced-order model's respective full FE model. Thus, discretization error is not included.

For comparison, each performance plot also includes a performance curve for the standard (single-level) HCB+ method. The number of FI modes in the HCB+ models is determined using the same frequency cutoffs as the AMLS+ models. Note that using the same cutoff frequencies for AMLS+ and HCB+ will not generally give the same final model sizes. In this case the model sizes are within 10% in almost all cases, with the HCB+ models typically being smaller.

From Fig. 3.7 it is clear that AMLS+ does not improve performance. Compared to HCB+, the computational speedup (if any) is very small but there is a significant loss in accuracy. Though not shown here, the same trends can also be observed when comparing AMLS and HCB without residual enhancement. It should also be noted that AMLS tends to perform worse for 3D models because the number of interface DOFs between substructures tends to be much more significant.

These results are quite unexpected because AMLS is reported to produce dramatic speedups over the HCB representation. An important thing to recognize here is that implementation plays a large role in the performance of any algorithm. Although with any algorithm I try to get the best possible performance, AMLS may be more susceptible to inefficient implementation due to its increased complexity compared to the HCB representation which relies mainly on built-in MATLAB functions like `eigs` and `mldivide`. Either way, AMLS will not be used in any further examples because it cannot match the performance of HCB reductions in my tests.

3.4 Summary

This chapter demonstrated the extension of CMS techniques to unit-cell problems, thus forming the BMS method. The HCB reduced-order model represents the interior DOFs of the model with a small set of normal modes and preserves the boundary DOFs. This reduced-order model

is obtained with an up-front calculation and can then be used to solve the eigenvalue problem for any wave vector. This chapter also applied a residual-mode enhancement technique to the reduced-order model that improves accuracy by approximating the contribution of the interior modes that were truncated in the HCB representation.

Although the reduced-order representation does not preserve the sparsity of the full FE model, the reduced-model size is typically small enough that this does not matter. Dramatic reductions in computation time are thus possible, especially for 2D problems. For 3D problems, the reduced-order model size may still be very large due to the un-reduced boundary. Thus, boundary reduction strategies for the unit-cell problem are investigated in the next chapter.

Chapter 4

Extension of Bloch Mode Synthesis to Incorporate Unit-Cell Boundary Reduction

4.1 Introduction

The BMS reduction is powerful because it replaces all of the interior DOFs in the unit-cell model with just a handful of modes that capture most of the dynamics of the interior. For very large models, especially in 3D, the remaining set of boundary DOFs may still be prohibitively large. In this case, a secondary reduction is useful. The secondary reduction must represent the boundary DOFs with a reduced set of modes. This topic has received some attention in the CMS literature because it is often beneficial to reduce the boundary DOF set between a pair of substructures.

The first and most common method for interface reduction is the CC mode reduction [52]. This takes the boundary partitions of the HCB reduced mass and stiffness matrices *from the coupled system* and uses them to compute a set of CC modes. These modes are then used to reduce the boundary of each substructure. The modes are computed and applied at the system-level, i.e., after the system has been assembled. For this reason, this is sometimes referred to as system-level CC (S-CC) reduction. The application of S-CC reduction to unit-cell models is presented in [20] and is reviewed in the first section of this chapter. The main shortcoming of this method is that the boundary modes must be computed for the coupled model. In the context of BMS, this means that Bloch BCs must be applied before computing the modes, and thus the reduction must be re-computed for every wave vector. This prevents the S-CC reduction from having any significant effect on the computational efficiency of band-structure computations. Additionally, this type of

reduction cannot be used with the $\mathbf{k}(\omega)$ solution method, and is thus restricted to $\omega(\mathbf{k})$ solutions.

The shortcomings of S-CC reduction can be addressed by performing the interface reduction locally, i.e., before substructures are assembled or Bloch BCs are applied. Many variations of local interface reduction exist [53]. The schematic diagram in Fig. 4.1 shows an overview of several types of interface reduction for a very simple structure model that has been divided into five substructures. The thumbnail figures show the level of assembly involved when the modes are computed. Note that in addition to the system- and local-level interface reductions, there is also hybrid-level CC (H-CC) which computes interface modes for partially assembled substructures. The schematic shows four types of local interface reduction. From left to right they are the exact-compatibility (EC) [54], weak-compatibility (WC), and uncoupled weak-compatibility (UWC) local-level CC (L-CC) methods, and the virtual node (VN) method. These techniques all have advantages and disadvantages, and more details about their performance can be found in [53]. For use with BMS, it was found that EC L-CC performs best, therefore from here on all local interface reductions are performed with the EC L-CC. To simplify notation, EC L-CC is shortened to simply L-CC.

For BMS models, local interface reduction makes it possible to reduce the unit-cell boundary before applying Bloch BCs. Thus, a single reduced model can be used to describe any wave vector in the BZ. This gives it a computational advantage over S-CC because the reduced model must only be computed once. Further, this type of model can be used to obtain either $\mathbf{k}(\omega)$ or $\omega(\mathbf{k})$ solutions.

Figure 4.2 is a flowchart depicting the steps involved in BMS, including the interface reduction. The light boxes were covered in chapter 3, and the darkened box shows L-CC reduction.

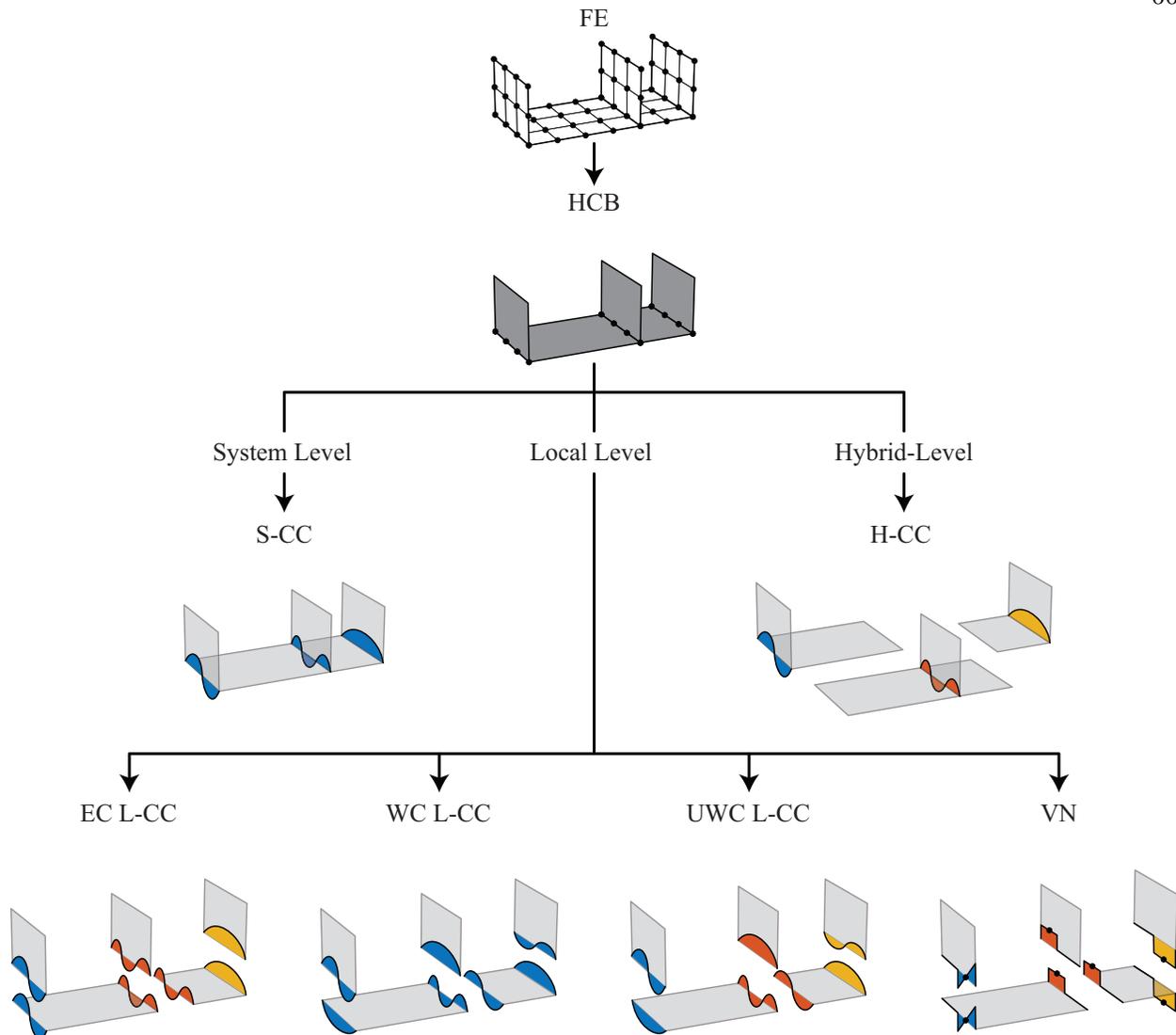


Figure 4.1: Schematic of the different interface reduction methods and their classification as system-, local-, or hybrid-level. The thumbnail illustrations show whether interface modes are coupled (single color) or uncoupled (different colors), the level of substructure assembly when the interface modes are computed, and for the local methods, whether modes match at the interface (indicating exact compatibility) or not (indicating weak compatibility).

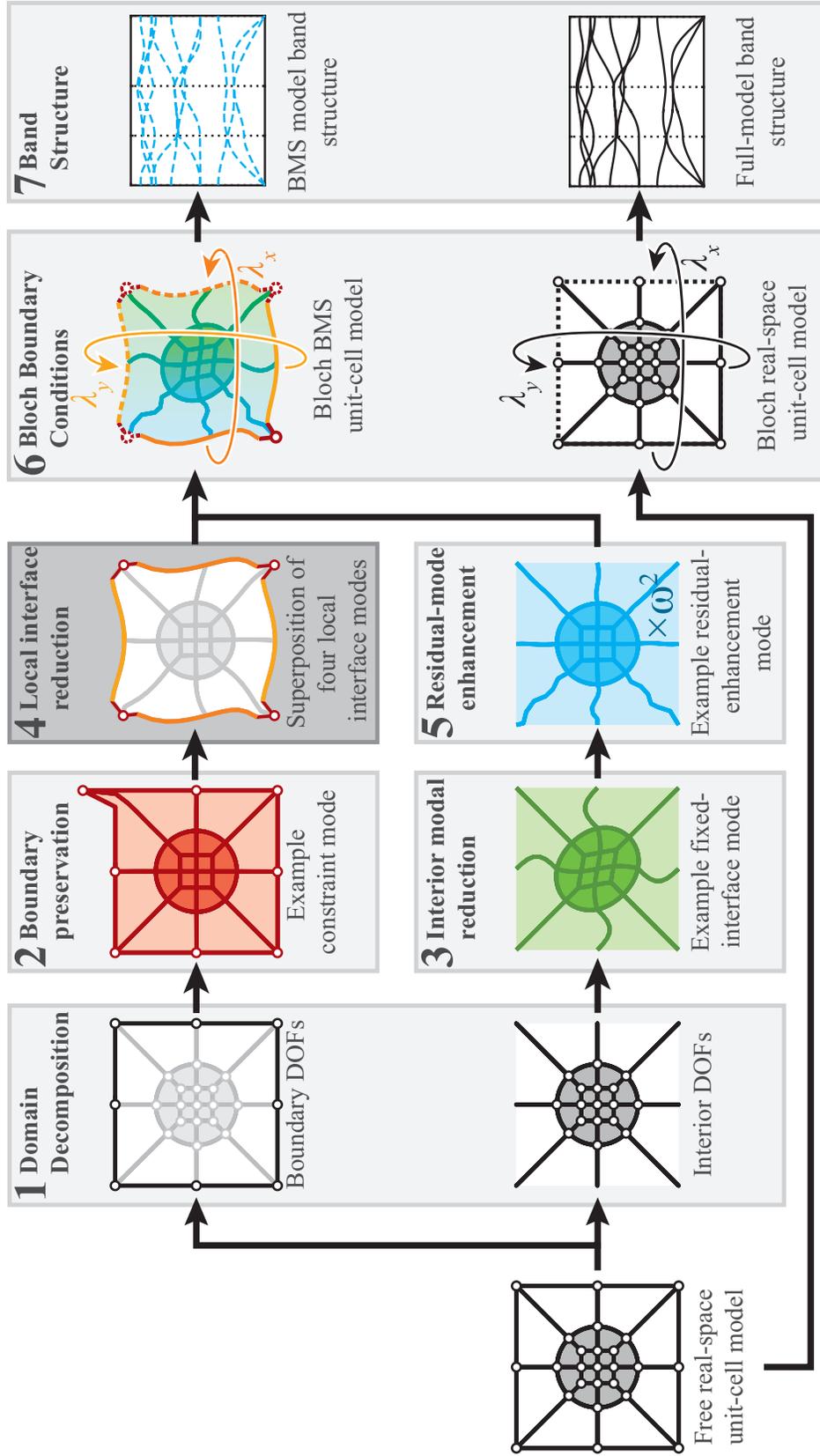


Figure 4.2: Flowchart illustrating the addition of a local interface reduction step to BMS. Compared to Fig. 3.1, the new box (step 4) is darkened. Note that the application of Bloch BCs is slightly modified to apply to boundary modes rather than to physical boundary DOFs.

4.2 System-Level Characteristic Constraint Mode Reduction

The S-CC reduction begins by partitioning the periodic (i.e. coupled) BMS mass and stiffness equations into interior and boundary components,

$$\left(\begin{bmatrix} \mathcal{K}_{II} & \hat{\mathcal{K}}_{IA} \\ \hat{\mathcal{K}}_{AI} & \hat{\mathcal{K}}_{AA} \end{bmatrix} - \omega^2 \begin{bmatrix} \mathcal{M}_{II} & \hat{\mathcal{M}}_{IA} \\ \hat{\mathcal{M}}_{AI} & \hat{\mathcal{M}}_{AA} \end{bmatrix} \right) \begin{Bmatrix} \eta_I \\ \hat{\mathbf{q}}_A \end{Bmatrix} = \mathbf{0}. \quad (4.1)$$

Note that a few sub-matrices and sub-vectors do not have the hat accent indicating Bloch periodicity. This is because those terms are not affected by the Bloch BC periodicity transformation. Using the boundary partitions, a set of normal mode shapes, $\hat{\Phi}_A$, is computed to describe the boundary motion,

$$\left(\hat{\mathcal{K}}_{AA} - \omega_A^2 \hat{\mathcal{M}}_{AA} \right) \hat{\phi}_A = 0, \quad (4.2)$$

$$\hat{\Phi}_A = \left[\left\{ \hat{\phi}_A \right\}_1 \quad \left\{ \hat{\phi}_A \right\}_2 \quad \cdots \quad \left\{ \hat{\phi}_A \right\}_{n_A} \right]. \quad (4.3)$$

This set of boundary modes is then used to form a transformation matrix to reduce the boundary terms,

$$\underbrace{\begin{Bmatrix} \eta_I \\ \hat{\mathbf{q}}_A \end{Bmatrix}}_{\hat{\mathbf{g}}} = \underbrace{\begin{Bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \hat{\Phi}_A \end{Bmatrix}}_{\hat{\mathbf{L}}} \underbrace{\begin{Bmatrix} \eta_I \\ \hat{\mathbf{q}}_A \end{Bmatrix}}_{\hat{\eta}}. \quad (4.4)$$

Using the S-CC transformation, $\hat{\mathbf{L}}$, the Bloch-periodic reduced-boundary BMS mass and stiffness matrices are formed,

$$\hat{\mathcal{M}} = \hat{\mathbf{L}}^\dagger \hat{\mathcal{M}} \hat{\mathbf{L}} = \hat{\mathbf{L}}^\dagger \left(\mathbf{P}^\dagger (\mathbf{B}^\top \mathbf{M} \mathbf{B}) \mathbf{P} \right) \hat{\mathbf{L}}, \quad (4.5)$$

$$\hat{\mathcal{K}} = \hat{\mathbf{L}}^\dagger \hat{\mathcal{K}} \hat{\mathbf{L}} = \hat{\mathbf{L}}^\dagger \left(\mathbf{P}^\dagger (\mathbf{B}^\top \mathbf{K} \mathbf{B}) \mathbf{P} \right) \hat{\mathbf{L}}. \quad (4.6)$$

These matrices are then used to obtain the band structure using the $\omega(\mathbf{k})$ method of section 2.5.1.

4.3 Local-Level Characteristic Constraint Mode Reduction

The S-CC reduction is effective for reducing the size of the model boundary while still retaining high accuracy, but it does not improve computational efficiency because a new set of reduction

modes must be computed for each wave vector. It would be much more efficient to perform just a single reduction of the boundary that is independent of the BZ. In order to do so, the boundary DOFs must be reduced before coupling the interface DOFs. This is possible using the L-CC reduction [54].

First, the free HCB reduced equations of motion are partitioned into interface and boundary sets,

$$\left(\begin{bmatrix} \mathcal{K}_{II} & \mathcal{K}_{IA} \\ \mathcal{K}_{AI} & \mathcal{K}_{AA} \end{bmatrix} - \omega^2 \begin{bmatrix} \mathcal{M}_{II} & \mathcal{M}_{IA} \\ \mathcal{M}_{AI} & \mathcal{M}_{AA} \end{bmatrix} \right) \begin{Bmatrix} \boldsymbol{\eta}_I \\ \mathbf{q}_A \end{Bmatrix} = 0. \quad (4.7)$$

Next, a set of normal mode shapes, $\boldsymbol{\Phi}_A$, is computed to describe the free boundary motion,

$$(\mathcal{K}_{AA} - \omega_A^2 \mathcal{M}_{AA}) \phi_A = \mathbf{0}, \quad (4.8)$$

$$\boldsymbol{\Phi}_A = \left[\{\phi_A\}_1 \quad \{\phi_A\}_2 \quad \cdots \quad \{\phi_A\}_{n_\phi^A} \right]. \quad (4.9)$$

An example of such a boundary mode shape is shown in Fig 4.3a. The goal is to perform a modal reduction using the boundary modes and then apply Bloch BCs. The mode shape can be partitioned into sets corresponding to each boundary set as shown in 4.3b,

$$\boldsymbol{\Phi}_A = \begin{Bmatrix} \boldsymbol{\Phi}_L \\ \boldsymbol{\Phi}_R \\ \boldsymbol{\Phi}_B \\ \boldsymbol{\Phi}_T \\ \boldsymbol{\Phi}_{BL} \\ \boldsymbol{\Phi}_{BR} \\ \boldsymbol{\Phi}_{TR} \\ \boldsymbol{\Phi}_{TL} \end{Bmatrix}. \quad (4.10)$$

Suppose that each boundary segment was simply reduced using the corresponding mode partition. To illustrate the problem with this approach consider just the right and left boundary sets. The modal reduction would take the following form:

$$\mathbf{q}_L = \boldsymbol{\Phi}_L \boldsymbol{\eta}_L, \quad \mathbf{q}_R = \boldsymbol{\Phi}_R \boldsymbol{\eta}_R. \quad (4.11)$$

In order to apply Bloch BCs, Eq. (4.11) is substituted into Eq. (2.18),

$$\Phi_L \eta_L = \Phi_R \eta_R \lambda x. \quad (4.12)$$

This must be solved for η_R , but this is not possible because Φ_R generally does not span the same column space as Φ_L . To circumvent this issue a single modal basis is used to reduce both the right and left boundary sets,

$$\mathbf{q}_L = \Phi_{LR} \eta_L, \quad \mathbf{q}_R = \Phi_{LR} \eta_R. \quad (4.13)$$

Then, Eq. (4.12) becomes,

$$\Phi_{LR} \eta_R = \Phi_{LR} \eta_L \lambda x, \quad (4.14)$$

which allows Bloch BCs to be enforced by directly relating the modal DOFs,

$$\eta_R = \eta_L \lambda x. \quad (4.15)$$

The only remaining question is how to obtain the single modal basis, Φ_{LR} so that it contains the important modal information from both the left and right boundary sets. This can be accomplished by simply combining the original left and right mode sets, $\Phi_{LR} = \begin{bmatrix} \Phi_L & \Phi_R \end{bmatrix}$. To ensure that the mode vectors in Φ_{LR} are not linearly dependent, the combined set of mode shapes is orthogonalized, using either the QR decomposition or singular value decomposition (SVD) (as is done in this case). Figure 4.3c shows an example of a resulting mode shape in Φ_{LR} used to reduce the left and right boundaries. After reduction by Φ_{LR} , the left and right boundaries are still compatible for connection by Bloch BCs.

Using the combined mode sets from each boundary, a block-diagonal transformation matrix is formed to reduce the entire boundary at once,

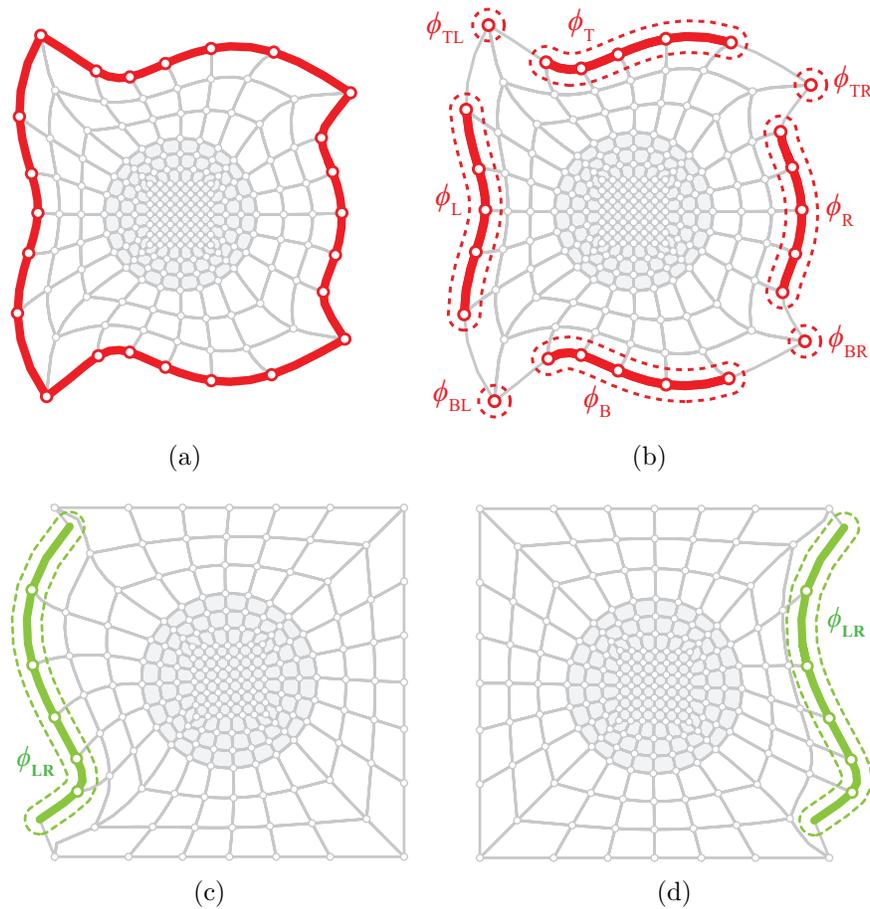


Figure 4.3: (a) Full boundary mode, (b) partitioned boundary modes for (c) left and (d) right boundaries. Although boundary modes only contain boundary DOFs, they define a linear combination of constraint modes which can be used to compute deflection in the unit-cell interior. The modes shown here include this interior deflection in light gray.

$$\begin{array}{c} \underbrace{\mathbf{g}} \\ \left. \begin{array}{c} \eta_I \\ \mathbf{q}_L \\ \mathbf{q}_R \\ \mathbf{q}_B \\ \mathbf{q}_T \\ \mathbf{q}_{BL} \\ \mathbf{q}_{BR} \\ \mathbf{q}_{TR} \\ \mathbf{q}_{TL} \end{array} \right\} = \end{array} \begin{array}{c} \underbrace{\mathbf{L}} \\ \left[\begin{array}{ccc} \mathbf{I} & & \\ \Phi_{LR} & & \mathbf{0} \\ & \Phi_{LR} & \\ & & \Phi_{BT} \\ & & & \Phi_{BT} \\ & & & & \Phi_C \\ & & & & & \Phi_C \\ \mathbf{0} & & & & & & \Phi_C \\ & & & & & & & \Phi_C \end{array} \right] \end{array} \begin{array}{c} \underbrace{\boldsymbol{\eta}} \\ \left. \begin{array}{c} \eta_I \\ \eta_L \\ \eta_R \\ \eta_B \\ \eta_T \\ \eta_{BL} \\ \eta_{BR} \\ \eta_{TR} \\ \eta_{TL} \end{array} \right\}, \end{array} \quad (4.16)$$

where $\Phi_{BT} = \begin{bmatrix} \Phi_B & \Phi_T \end{bmatrix}$ is the compatible mode set for the bottom-top boundary DOFs, and $\Phi_C = \begin{bmatrix} \Phi_{BL} & \Phi_{BR} & \Phi_{TR} & \Phi_{TL} \end{bmatrix}$ is the compatible mode set for the corner boundary DOFs. Substituting into Eq. (2.45), gives the free boundary-reduced BMS equations of motion.

$$(\mathcal{K} - \omega^2 \mathcal{M})\boldsymbol{\eta} = 0, \quad (4.17)$$

where

$$\mathcal{M} = \mathbf{L}^T \mathcal{M} \mathbf{L}, \quad \mathcal{K} = \mathbf{L}^T \mathcal{K} \mathbf{L}. \quad (4.18)$$

To enforce Bloch periodicity, the modal DOF vectors are linked just as the physical DOF vectors

are linked in Eq. (2.20),

$$\underbrace{\begin{Bmatrix} \eta_I \\ \eta_L \\ \eta_R \\ \eta_B \\ \eta_T \\ \eta_{BL} \\ \eta_{BR} \\ \eta_{TR} \\ \eta_{TL} \end{Bmatrix}}_{\eta} = \underbrace{\begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \lambda_x \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \lambda_y \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \lambda_x \mathbf{I} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \lambda_x \lambda_y \mathbf{I} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \lambda_y \mathbf{I} \end{bmatrix}}_{\mathbf{P}} \underbrace{\begin{Bmatrix} \hat{\eta}_I \\ \hat{\eta}_L \\ \hat{\eta}_B \\ \hat{\eta}_{BL} \end{Bmatrix}}_{\hat{\eta}}. \quad (4.19)$$

Substituting this into Eq. (4.7) gives the Bloch-periodic boundary-reduced BMS equations of motion,

$$\mathbf{P}^\dagger (\mathcal{K} - \omega^2 \mathcal{M}) \mathbf{P} \hat{\eta} = 0. \quad (4.20)$$

The band structure can now be obtained using the $\omega(\mathbf{k})$ approach described in section 2.5.1 or the $\mathbf{k}(\omega)$ approach described in section 2.5.2.1.

4.4 Numerical Experiments

In this section some results are demonstrated using two example materials. The first is the 2D material with an 8-lobed inclusion in every unit cell included earlier. The second is a locally resonant acoustic metamaterial [5] consisting of a heavy spherical resonator coated in a soft silicon rubber embedded in an epoxy matrix. The numerical experiments aim to characterize both the speedups obtained relative to the full model and the error in the reduced-model results with respect to the full-model results.

The examples in this section explore different combinations of the interface and interior reduction strategies described earlier. To make it simpler to discuss the various reduction strategies, the abbreviations shown in Table 4.1 are used.

Abbreviation	Interior Reduction Type	Residual-Mode Enhancement	Interface Reduction
BMS_{HCB}	HCB	✗	-
BMS_{HCB+}	HCB	✓	-
BMS_{HCB}^{L-CC}	HCB	✗	EC L-CC
BMS_{HCB+}^{L-CC}	HCB	✓	EC L-CC

Table 4.1: Abbreviations for different generalized BMS configurations

4.4.1 Eight-Petal Inclusion Model

The first material to be considered has a 2D unit cell with an 8-lobed inclusion as seen in chapter 3.

4.4.1.1 Band-Structure Calculation using $\omega(\mathbf{k})$ Method

Using the $\omega(\mathbf{k})$ approach, the first 10 dispersion branches of the 23040 DOF large mesh are computed at 97 \mathbf{k} -points. Then, a $\text{BMS}_{\text{HCB}+}^{\text{L-CC}}$ reduced-order model with 80 DOFs (30 interior modes and 50 boundary modes) is formed, and again the band structure is computed. Figure 4.4a shows both the full- and reduced-model band-structure diagrams, and indicates very good visual agreement. Indeed the maximum frequency error over the first 10 branches is just 0.03% relative to the full model. This ensures that the overall error is dominated by the discretization and not by the reduction, but still produces a very compact reduced-order model. The reduced-model band-structure calculation time is just 4.1 seconds which is over 40 times faster than the full-model calculation time of 170 seconds. Of these 4.1 seconds, 3.9 are spent on the up-front calculations, and just a fraction of a second is needed to visit all of the \mathbf{k} points. The performance details are summarized in table 4.2.

Figure 4.4b shows a mode-shape comparison between the full and reduced models for the point highlighted on the dispersion plot. The error between the full and reduced mode shapes is $e_\phi = 1.71 \times 10^{-6}$. The maximum mode error for the band structure computed in Fig. 4.4a is $e_\phi = 7.15 \times 10^{-5}$.

4.4.1.2 Complex Band-Structure Calculation using $\mathbf{k}(\omega)$ Method

As a second example using the same model, the $\mathbf{k}(\omega)$ approach is used to compute both the propagating and evanescent band structure. Figure 4.5 shows a comparison of the full-model and BMS-model complex band structures computed for the Γ -X direction. For plotting and error comparison, only the 8 branches nearest the gamma point are considered, but the $\text{BMS}_{\text{HCB}+}^{\text{L-CC}}$ model used to compute the complex band structure is the same 80 DOF model used in the previous

	Full Model	BMS _{HCB+} ^{L-CC} Model	
Model Details	Model size (DOF)	23040	
		80 (30 interior, 50 boundary)	
	Up-front time (s)		1.9 (Interior mode)
			1.1 (Constraint mode)
		+ 0.9 (Other)	
		<u>3.9</u>	
$\omega(\mathbf{k})$ Calc.	Per- \mathbf{k} -point time (s)	1.75	
	Total time, 97 \mathbf{k} pts. (s)	170.1	
	Max. frequency error (%)	-	
	Max. mode error	-	
$\mathbf{k}(\omega)$ Calc.	Per- ω -pt. time (s)	51.5	
	Total time, 200 ω pts. (s)	10308	
	Max. wave-vector error (%)	-	
		0.0028	
		4.1	
		0.034	
		7.15×10^{-5}	
		0.0070	
		5.6	
		6.5	

Table 4.2: Comparison of full and BMS reduced-model calculation time and model-size for 8-lobed inclusion material band-structure calculations.

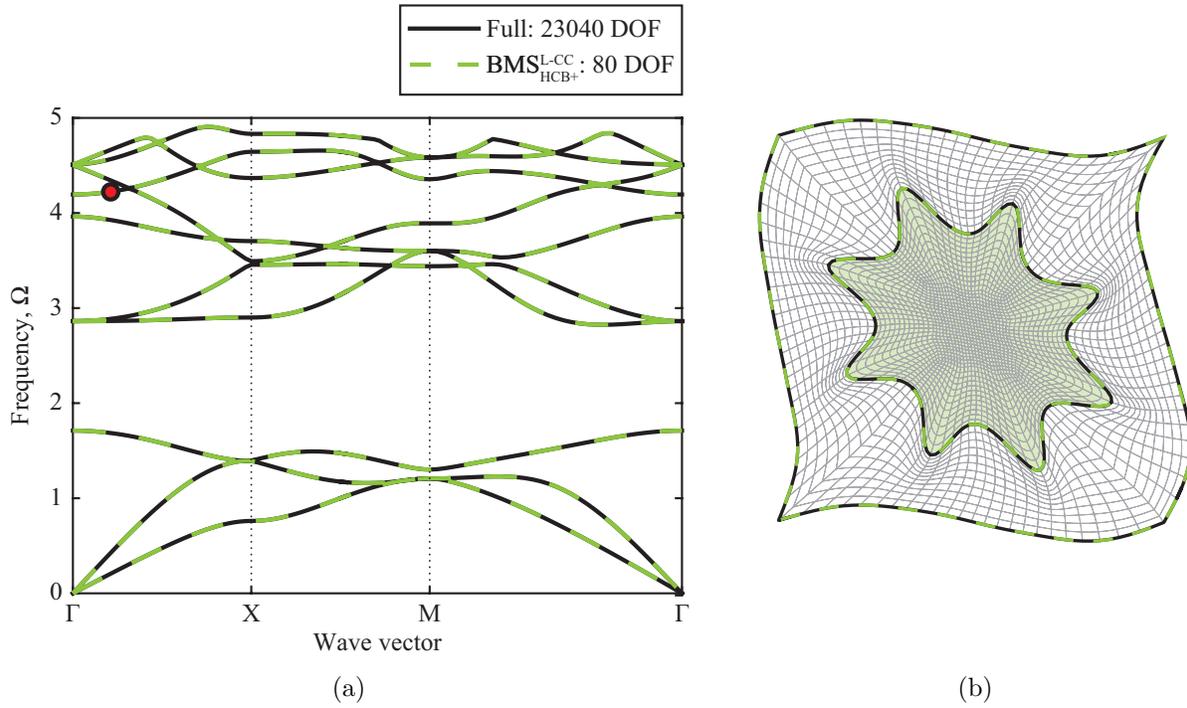


Figure 4.4: (a) Dispersion-diagram comparison for 8-lobed inclusion material where $\Omega = \omega L_x \sqrt{\rho/E}$ is the non-dimensional frequency, and (b) selected mode-shape comparison for the mode highlighted in the dispersion diagram.

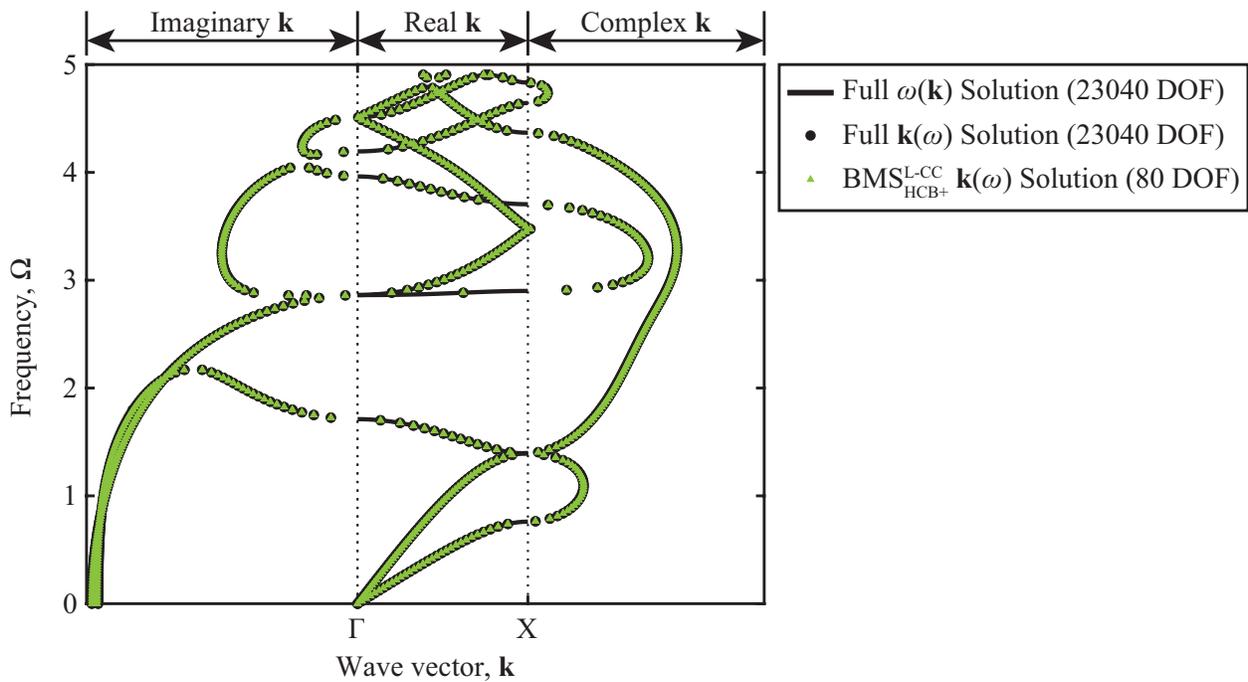


Figure 4.5: Complex dispersion-diagram comparison for 8-lobed inclusion material where $\Omega = \omega L_x \sqrt{\rho/E}$ is the non-dimensional frequency.

section. The reduced-model band-structure calculation time is just 5.6 seconds which is over 1800 times faster than the full-model calculation time of 4016 seconds (approximately 3 hours). Less than 2 seconds are needed to visit all of the ω points with the BMS model. The maximum error in the BMS wave vectors is 6.5% with respect to the full model.

4.4.1.3 Performance Evaluation

The performance of the BMS method is observed by computing the $\omega(\mathbf{k})$ band structure for various BMS reductions of the large mesh, and plotting the maximum frequency error versus computation time in Fig. 4.6. The maximum frequency error is evaluated for the first 10 branches of the band structure across 97 \mathbf{k} points. The curves in Fig. 4.6 are generated by varying the number of FI modes, n_A^ϕ , and performing either interface reduction using $c = 0.25$, or no interface reduction at all.

All of the methods show a sharp initial drop in error with little increase in computational time as the cutoff frequency is raised. Further increase in the cutoff frequencies produce diminishing returns as the curves level off. The residual-mode enhancement is able to reduce error levels by over two orders of magnitude with little additional computational cost. Interface reduction is able to reduce the per- \mathbf{k} -point computation time significantly, but the total computation times are so heavily dominated by the up-front time that the curves do not show much separation. The curves would separate more if the number of \mathbf{k} was larger.

4.4.2 Acoustic Metamaterial Model

The second material to be considered has a 3D cubic unit cell composed of resin with a resonator inside. The resonator consists of a lead sphere coated in silicon rubber. The mesh is plotted in Fig. 4.7 with a section removed to show the spherical resonator. The FE model consists of 352 quintic Lagrangian brick elements and contains 67,656 DOFs (after enforcing Bloch BCs).

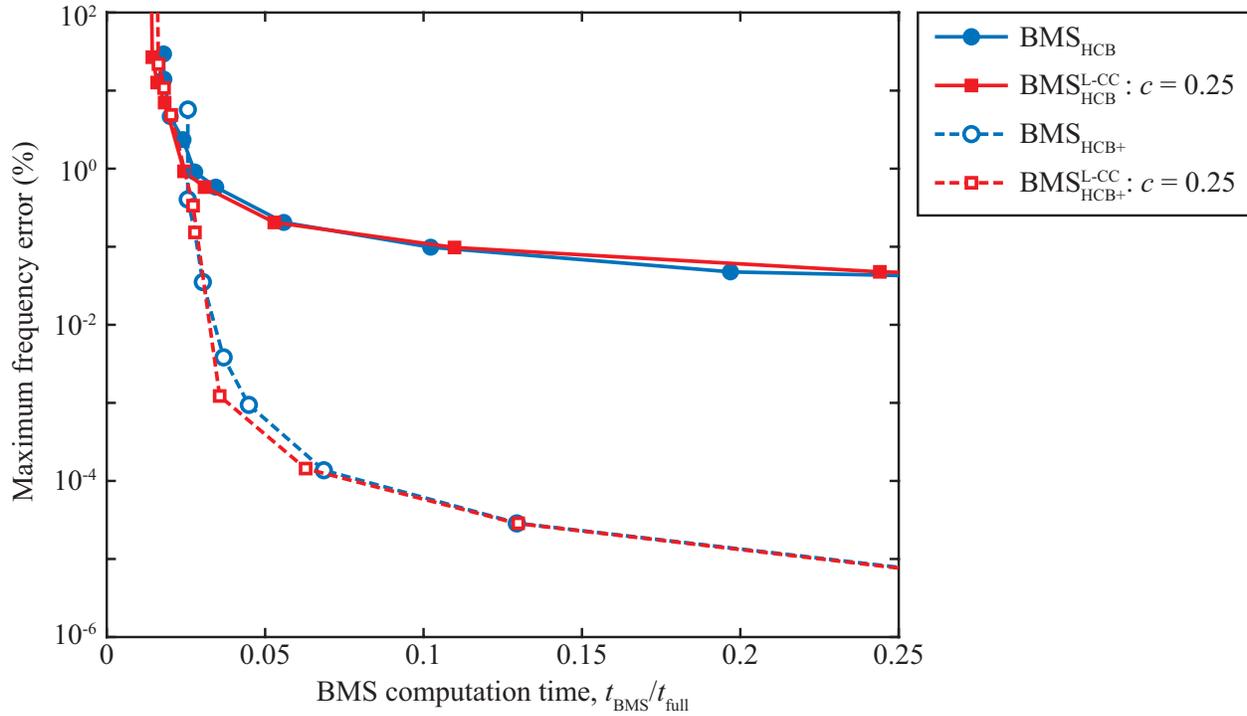


Figure 4.6: Performance plot of computation time fraction versus frequency error, generated by computing $\omega(\mathbf{k})$ band structures with the large mesh of the 8-lobe inclusion material.

		Full Model	BMS _{HCB+} ^{L-CC} Model
Model Details	Model size (DOF)	67656	628 (250 interior, 378 boundary)
	Up-front calculation time (s)	-	1190 (Interior mode)
			950 (Constraint mode)
		+ 310 (Other)	
			2450
$\omega(\mathbf{k})$ Calc.	Per- \mathbf{k} -point time (s)	1398	0.61
	Total time, 65 \mathbf{k} pts. (s)	90870	2490
	Max. frequency error (%)	-	1.89
	Max. mode error	-	0.0039

Table 4.3: Timing and model-size details for acoustic metamaterial.

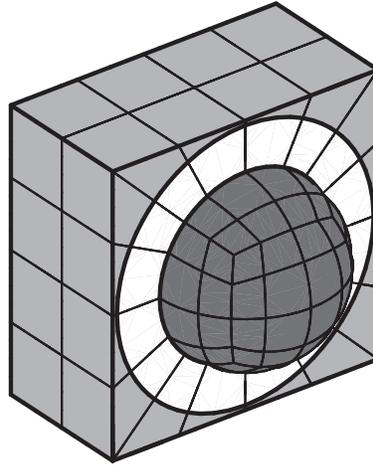


Figure 4.7: FE mesh for acoustic metamaterial unit cell consisting of a soft rubber (white) and heavy lead (dark grey) resonator embedded in an epoxy (light grey) matrix.

4.4.2.1 Band-Structure Calculation using $\omega(\mathbf{k})$ Method

A $\text{BMS}_{\text{HCB}^+}^{\text{L-CC}}$ reduced-order model is formed with 250 interior modes and 378 boundary modes for a total of 628 DOFs. The first 150 curves of the band structure are computed using the full model and the BMS model, and both are plotted in Fig. 4.8a. Note that there are many flat bands whose frequencies do not change throughout the BZ. These represent resonator modes that couple very weakly or not at all with waves moving through the material. Although they do not carry any energy through the system, these modes are physically meaningful. The flat bands are grayed out in the band-structure diagram so that the energy carrying branches can be seen more easily.

The BMS model produces band-structure curves that match the full model very well. The maximum error in band-structure frequencies is: 1.9%. Figure 4.8b also shows a reduced-model mode shape corresponding to the point highlighted in the band-structure diagram. The maximum mode error over the first 148 branches is 0.0039. The 149th and 150th branches are ignored because they contain degeneracies with higher uncomputed branches.

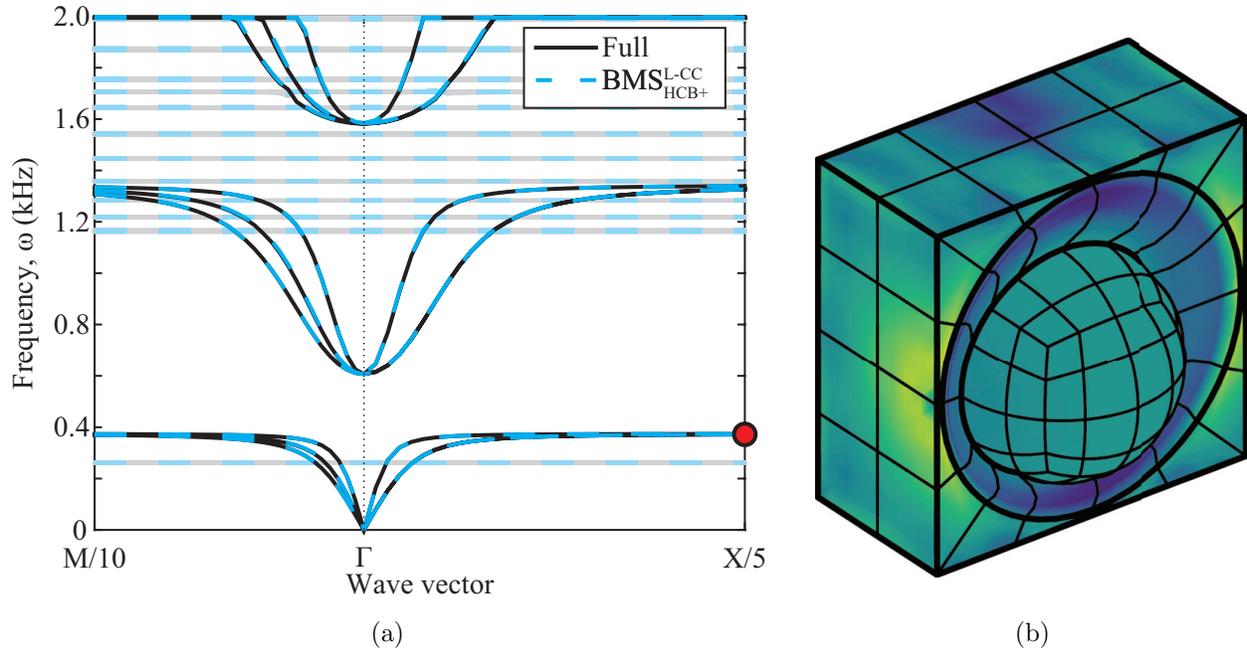


Figure 4.8: (a) Dispersion-diagram comparison for acoustic metamaterial, and (b) selected mode shape computed with BMS.

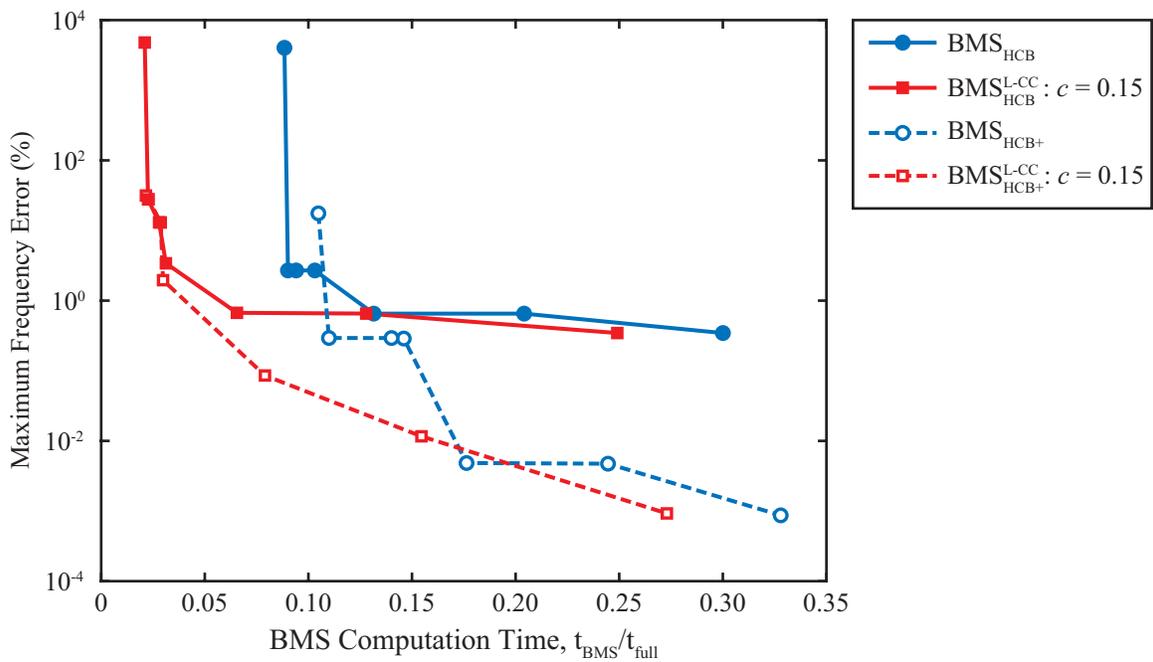


Figure 4.9: Performance plot of computation time fraction versus frequency error, generated by computing $\omega(\mathbf{k})$ band structures for the resonator model.

4.4.2.2 Performance Evaluation

The performance of the BMS method is studied by computing the $\omega(\mathbf{k})$ band structure for various BMS reductions of the large mesh, and plotting the maximum frequency error versus computation time in Fig. 4.6. The maximum frequency error is evaluated for the first 150 branches of the band structure across 65 \mathbf{k} points. The curves in Fig. 4.6 are generated by varying the number of FI modes, n_A^ϕ , and performing either interface reduction using $c = 0.15$, or no interface reduction at all.

As with the previous model, all of the performance curves show a sharp initial drop in error with little increase in computational time as the cutoff frequency is raised. The error levels do not get quite as low as those in the previous example. A possible explanation for this is the presence of the flat branches that do not carry energy. Each of these flat branches corresponds to a single FI mode. The flat branches effectively “use up” a FI mode that does not contribute to the solution at the elastic branches.

4.5 Summary

This chapter described the use of EC L-CC interface reduction to reduce the number of boundary DOFs in the BMS reduced-order model. The boundary reduction is performed locally so that BCs can be applied after the reduction. The reduced model size due to the secondary boundary reduction can significantly improve performance, especially if a large number of \mathbf{k} points or ω points are considered in the band-structure calculation. All of the local interface reduction methods described at the beginning of this chapter can be applied to unit cells. Some perform better than others, and nearly all of them have parameters that can be adjusted to improve performance. A possible direction for future work is to more thoroughly investigate the different interface reduction methods to make sure the best-performing, most robust technique is identified. An additional direction for future work is to extend the residual-mode enhancement — which is used to make the interior reduction more accurate — to the interface reduction as well. Residual-mode enhancement

has been successfully applied to the S-CC interface reduction, and it may be possible to extend the same ideas to one or more of the L-CC techniques as well.

Chapter 5

Application of Bloch Mode Synthesis to Electronic Band-Structure Problems

5.1 Introduction

Schrödinger's equation, published in 1926, governs the wave-mechanics of electrons, and holds the key to many if not all material properties [55]. A few years later in 1929, Paul Dirac [56] wrote "The underlying physical laws necessary for the mathematical theory of a large part of physics and the whole of chemistry are thus completely known, and the difficulty is only that the exact application of these laws leads to equations much too complicated to be soluble." Indeed, electronic structure — obtained by solving Schrödinger's equation — has played a vital role in our understanding of many physical properties in crystalline solids. Conductivity, absorptivity, and photoemission are just a few properties that can be explained simply by looking at a material's band-structure diagram [3]. The band structure provides valuable insight into experimental results and can even be used to explore databases of materials for which experimental data is not available [57]. One can imagine taking this even one step further by using optimization to guide the design of new materials.

Fortunately, since Dirac's comment in 1929, techniques have emerged for obtaining approximate (though in many cases very accurate) solutions to Schrödinger's wave equation. A common framework for doing so is density functional theory (DFT) [58, 59]. Perhaps the most difficult aspect of electronic calculations is to include electron-electron interactions. DFT transforms the problem of many interacting electrons into an equivalent but much more tenable system of noninteracting electrons in an effective potential. This mapping transforms the Schrödinger equations

into the Kohn-Sham equations. Despite this powerful simplification, solution of the Kohn-Sham equations can still be a formidable challenge because the solution requires self consistency. Forming the Kohn-Sham equations requires knowledge of the electron density, which can be obtained from the solution of the Kohn-Sham equations. This circular dependency requires iteration until a self-consistent solution is reached.

There are a few simple electronic systems such as the Hydrogen atom for which electronic structure can be derived analytically. For nearly all practical problems however, the Kohn-Sham equations must be discretized in space — yielding a discrete eigenvalue problem — before they can be solved. PW techniques form a very common discretization methodology for electronic-structure calculations because the solutions converge very rapidly with the number of terms in the Fourier expansion, leading to very small models. Despite this, the method can become computationally unwieldy for systems with more than a few thousand atoms because the global basis functions produce a dense system of equations. Furthermore, the Fourier basis has uniform resolution everywhere so it is not naturally suited to handle sharp gradients in electron density or regions of vacuum where the wavefunction is zero. These situations are common in nano-structured metamaterials [60]. Even in relatively simple crystals, sharp gradients in the wavefunction can occur near ion cores. For this reason most PW electronic-structure calculations rely on pseudopotentials which give a much smoother potential that is nonetheless accurate away from the ion cores. Pseudopotential methods can give very good accuracy, but they treat the behavior of core electrons and valence electrons as uncoupled problems so some uncertainty as to the validity of the results is introduced.

The limitations of PW-based techniques have led to the investigation of alternative spatial discretizations of electronic systems such as the FE method. The FE method produces sparse matrices and can have arbitrarily high local spatial refinement, allowing for sharp gradients and surfaces to be handled much more easily than with PW methods. Over the past four decades several researchers have tested FE methods on electronic systems. Lowney *et al* [61] computed the ground state of an electron in an ellipsoidal Hamiltonian using a 2D mesh by taking advantage of the axisymmetric nature of the problem. Hermansson and Yelnick demonstrated electronic-structure

calculations using a regular 3D mesh, but concluded that PW methods are superior for smooth potentials like the AlSb pseudopotential they considered. White, Wilkins, and Teter [62] took advantage of adaptive refinement to obtain energy levels for H, H₂⁺, He, and H₂.

Despite the benefits of FE techniques, they have not been widely adopted for electronic-structure calculations because FE model sizes are much larger than PW models with similar accuracy. Although FE models yield sparse rather than dense systems, the difference in model size means that PW models still typically have the computational edge. By introducing the BMS model-order reduction to electronic-structure FE calculations, it is possible to narrow the performance gap between these methods, and potentially give FE methods the computational advantage for some problems.

This chapter gives an overview of the steps involved in computing the electronic band structure of a crystalline material using FE based DFT. Then, BMS model-order reduction is applied to the model and the impact on performance is reported.

5.2 Electronic Structure Theory

Electronic band structure defines the states that electrons in an electrostatic field can occupy. For materials, this electrostatic field comes from the atomic nuclei which are much more massive than the electrons and thus by the Born-Oppenheimer approximation are treated as fixed point charges. The electronic states belonging to the band structure are fully defined by the wavefunction (which characterizes each electron's position and spin), and the corresponding energy.

Schrödinger's wave equation provides the fundamental description of electrons in potential fields [55]. For a system with N electrons, the time independent Schrödinger's equation is as follows,

$$\left[\frac{\hbar^2}{2m_e} \sum_{i=1}^{N_e} \nabla_i^2 + \sum_{i=1}^{N_e} V(\mathbf{x}_i) + \sum_{i=1}^{N_e} \sum_{j<i} U(\mathbf{x}_i, \mathbf{x}_j) \right] \Psi = E\Psi, \quad (5.1)$$

where h is Planck's constant, \mathbf{x}_i is the position of the i^{th} electron, $V(\mathbf{x}_i)$ is the potential energy of the i^{th} electron due to the atomic nuclei, $U(\mathbf{x}_i, \mathbf{x}_j)$ is the interaction energy between the i^{th} and j^{th} electrons, Ψ is the wavefunction, and E is the ground state energy [9]. This equation is

very challenging to solve because the wavefunction depends on all $3N_e$ spatial coordinates of the electrons.

5.2.1 Density Functional Theory

DFT represents a simplification of Schrödinger's equation, a many-body equation in $3N_e$ coordinates to a set of single-body equations in 3 coordinates. This simplified set of equations arises from the minimization of the ground state energy, a functional of the electron density. The theoretical basis of DFT can be attributed to the work of Hohenberg and Kohn [58], and of Kohn and Sham [59].

The first step in simplifying Schrödinger's equation, is to introduce the electron density, $\rho(\mathbf{x}) = |\Psi(\mathbf{x})|^2$, which is proportional to the square of the complex amplitude of the wavefunction. Hohenberg and Kohn proved two important theorems that make the electron density very interesting for quantum calculations [58]. They showed that (1) the ground state energy is a unique functional of the electron density, and (2) the true electron density is the one that minimizes the ground state energy. This means that rather than finding the wavefunction in terms of $3N_e$ spatial dimensions, it is sufficient to find the electron density, a function of just 3 spatial dimensions. Kohn and Sham then showed that it is further possible to replace the problem of interacting electrons with a problem of non-interacting electrons that are acted on by an effective potential that encompasses electron-electron interactions [59]. By appropriate selection of the effective potential, the exact same electron density can be obtained with both the interacting and non-interacting electron problems. The non-interacting or single-electron equations are often called the Kohn-Sham equations:

$$\left[-\frac{\hbar^2}{2m_e} \nabla^2 + V(\mathbf{x}) + V_H(\mathbf{x}) + V_{XC}(\mathbf{x}) \right] \Phi_i(\mathbf{x}) = E_i \Phi_i(\mathbf{x}). \quad (5.2)$$

where $\Phi_i(\mathbf{x})$ is the Kohn-Sham orbital for the i^{th} electron. It is also possible to calculate some excited states with the Kohn-Sham equations, so the eigenvalue index, i , is not necessarily limited to the number of electrons. The number of calculated eigenvalues (bands) is denoted by N_b . The

electron density can be calculated by summing over the bands:

$$\rho(\mathbf{x}) = 2 \sum_{i=1}^{N_b} f_{\text{occ}}(E_i) \Phi_i(\mathbf{x})^* \Phi_i(\mathbf{x}), \quad (5.3)$$

where the factor of 2 allows for spin-up and spin-down occupation of orbitals, and $f_{\text{occ}}(E_i)$ gives the statistical likelihood that a band with energy E_i is occupied. In the Kohn-Sham equations, the electron interaction term, U from the Schrödinger equation has been replaced by the Hartree potential, V_H , and the exchange-correlation potential, V_{XC} . These new potential terms will not be discussed in detail here, except to say that they can be defined in such a way that the Kohn-Sham equations produce the exact same electron density as the full interacting-electron equations. In this chapter, the focus is not on how the potential is obtained. Rather, the goal is to show that for a given potential, BMS can be used to quickly and accurately obtain the single-electron wave-functions and energy levels. Thus, for the remainder of this chapter, the potential terms are lumped together into an effective potential, $V_{\text{eff}}(\mathbf{x}) = V(\mathbf{x}) + V_H(\mathbf{x}) + V_{XC}(\mathbf{x})$, so the Kohn-Sham equations become:

$$\left[-\frac{\hbar^2}{2m_e} \nabla^2 + V_{\text{eff}}(\mathbf{x}) \right] \Phi_i(\mathbf{x}) = E_i \Phi_i(\mathbf{x}). \quad (5.4)$$

To reiterate, the non-interacting electron system represented by the Kohn-Sham equations by construction produces the exact same electron density as the interacting electron system. There is no such guarantee about the energy eigenvalues or Kohn-Sham orbitals. In fact, these orbitals and eigenvalues cannot be shown to have any physical meaning (other than to produce the correct charge density) [63]. Thus, it is a bit strange that the Kohn-Sham eigenvalues are used to represent the energy bands. Although the Kohn-Sham eigenvalues and orbitals cannot be shown theoretically to correspond to physical properties, they have been observed to reproduce empirical results with good accuracy.

As mentioned earlier, the Kohn-Sham equations require an iterative solution because the effective potential used to find the Kohn-Sham orbitals is (via the electron density) a function of the Kohn-Sham orbitals themselves. The steps involved in self-consistent field iteration are summarized in Fig. 5.1.

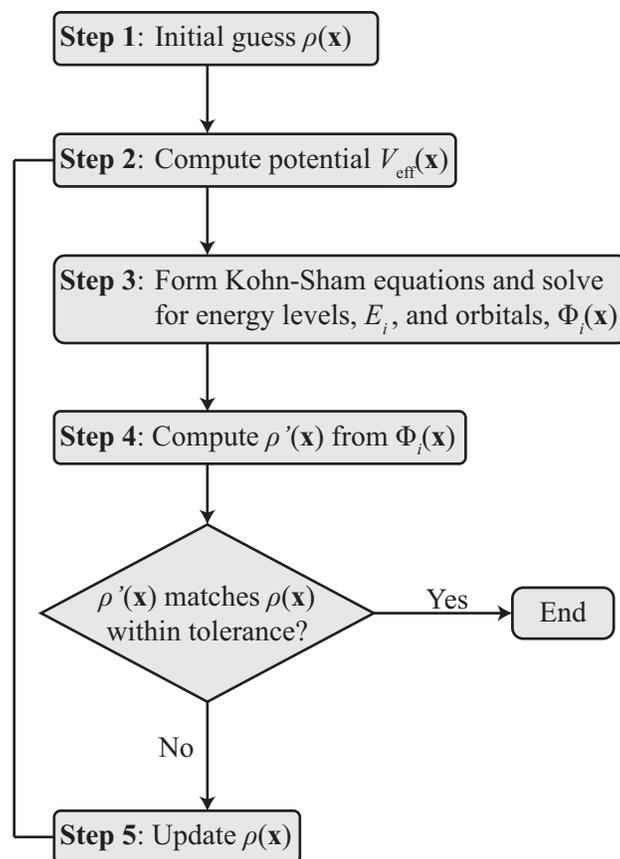


Figure 5.1: Flowchart depicting self-consistent field iteration

The computationally demanding part of this iteration is commonly the eigenvalue solution(s) of step 3. BMS can be applied to step 3 during every iteration. Thus, as long as BMS is sufficiently accurate so as not to negatively impact the convergence of the iterative process, it is reasonable to expect the same performance improvements for the overall DFT calculation as for a single iteration. Thus, for the remainder of this chapter, just a single iteration is considered, beginning with a specified potential and finishing with the calculation of the electron density.

5.2.2 Bloch's Theorem

To this point, the discussion of electronic band-structure is applicable to material (i.e. crystal) calculations, as well as to discrete molecular calculations. Since BMS is only applicable only to the former case however, the discussion is specialized to the material case here. As was the case for

phononic crystals, electronic materials are idealized as extending infinitely in space with a repeating unit cell, and thus admit a Bloch wave solution. This allows for characterization of the material modeling just its unit cell. Similar to the displacement field in Eq. (2.3), Bloch's theorem can be applied to the Kohn-Sham orbitals,

$$\Phi_i(\mathbf{x}, \mathbf{k}) = \tilde{\Phi}_i(\mathbf{x}, \mathbf{k})e^{i\mathbf{k}^T \mathbf{x}}, \quad (5.5)$$

where $\tilde{\Phi}_i$ is a strictly periodic function that satisfies the following relation for any lattice vector, \mathbf{G} ,

$$\tilde{\Phi}_i(\mathbf{x}, \mathbf{k}) = \tilde{\Phi}_i(\mathbf{x} + \mathbf{G}, \mathbf{k}). \quad (5.6)$$

5.2.3 Brillouin Zone Integration

For each wave vector or point in momentum space, the solution of the Kohn-Sham equations may yield a different set of energy levels and corresponding orbitals. Thus, a separate electron density could be found for each κ point. The true electron density is the average over all possible states weighted by the occupation function. There are infinitely many momentum states, but due to the periodicity of electron states in reciprocal space, the average can be found by integrating just over the first BZ.

In practice, these integrals are performed numerically. Numerous techniques for BZ integration have been proposed and investigated. Perhaps the simplest and most ubiquitous approach for BZ integrations is Monkhorst-Pack (MP) integration [64]. The main idea is to create an $(n \times n \times n)$ grid of points that spans the parallelepiped (or parallelogram in 2D) defined by the reciprocal lattice vectors:

$$\mathbf{k}_{pqr}^{\text{MP}} = \frac{2p-n-1}{2n}\mathbf{b}_1 + \frac{2q-n-1}{2n}\mathbf{b}_2 + \frac{2r-n-1}{2n}\mathbf{b}_3, \quad \text{for } p, q, r = 1 \dots n. \quad (5.7)$$

Although the MP grid points do not necessarily fall inside the first BZ, the volume occupied by the grid is equal to that of the BZ. Monkhorst and Pack chose to use a regular grid because for a certain density of points one can exactly integrate Fourier terms up to a certain order.

For any crystal, the states in the BZ are guaranteed to exhibit some symmetry, meaning that the IBZ is smaller than the BZ. Even the lowest symmetry crystals have at least a twofold reduction in size from the BZ to the IBZ because assuming time reversal symmetry, waves traveling in opposite directions are identical (up to a reflection). This allows a reduced set of points to be used in the MP integration. The summation is performed by considering the reduced subset of MP points, \mathbf{k}^{MPR} , that are unique under the symmetry operations that exist in the crystal. The contribution of each of these reduced points is multiplied by a weight, w , that specifies the number of points in the same symmetry group. Figure 5.2 shows (8×8) MP grids for a square lattice and for a hexagonal lattice. For each lattice, the original grid and a shifted grid are shown. The grid shift, \mathbf{s} is always half of a grid step:

$$\mathbf{s} = \frac{1}{2n} (\mathbf{b}_1 + \mathbf{b}_2 + \mathbf{b}_3). \quad (5.8)$$

Note that for an even number of grid points, the Γ point (origin) is not included in the original grid, but after the shift it is included. For an odd number of grid points, this situation is reversed. Any grid that includes the Γ point is referred to as Γ -centered.

The original and shifted grids give the same accuracy because they have the same density of \mathbf{k} in reciprocal space. Typically, one or the other is better because it leads to fewer points in the symmetry-reduced set (and thus fewer solutions of the Kohn-Sham equations). For the square lattice, it is favorable to use the non- Γ -centered grid, whereas for the hexagonal lattice it is favorable to use the Γ -centered grid.

Since the charge density is a function of space, some extra precautions must be taken to accurately represent the full MP integration with a symmetry-reduced set of points. A slightly simpler case will be considered first. The band energy gives the sum of energy over all occupied bands, and is not a function of spatial coordinates so symmetry-reduced MP integration is very straightforward. The band energy can be expressed as a function of wavevector as follows,

$$E_b(\mathbf{k}) = \sum_{i=1}^{N_b} E_i(\mathbf{k}) f_{\text{occ}}(E_i(\mathbf{k})). \quad (5.9)$$

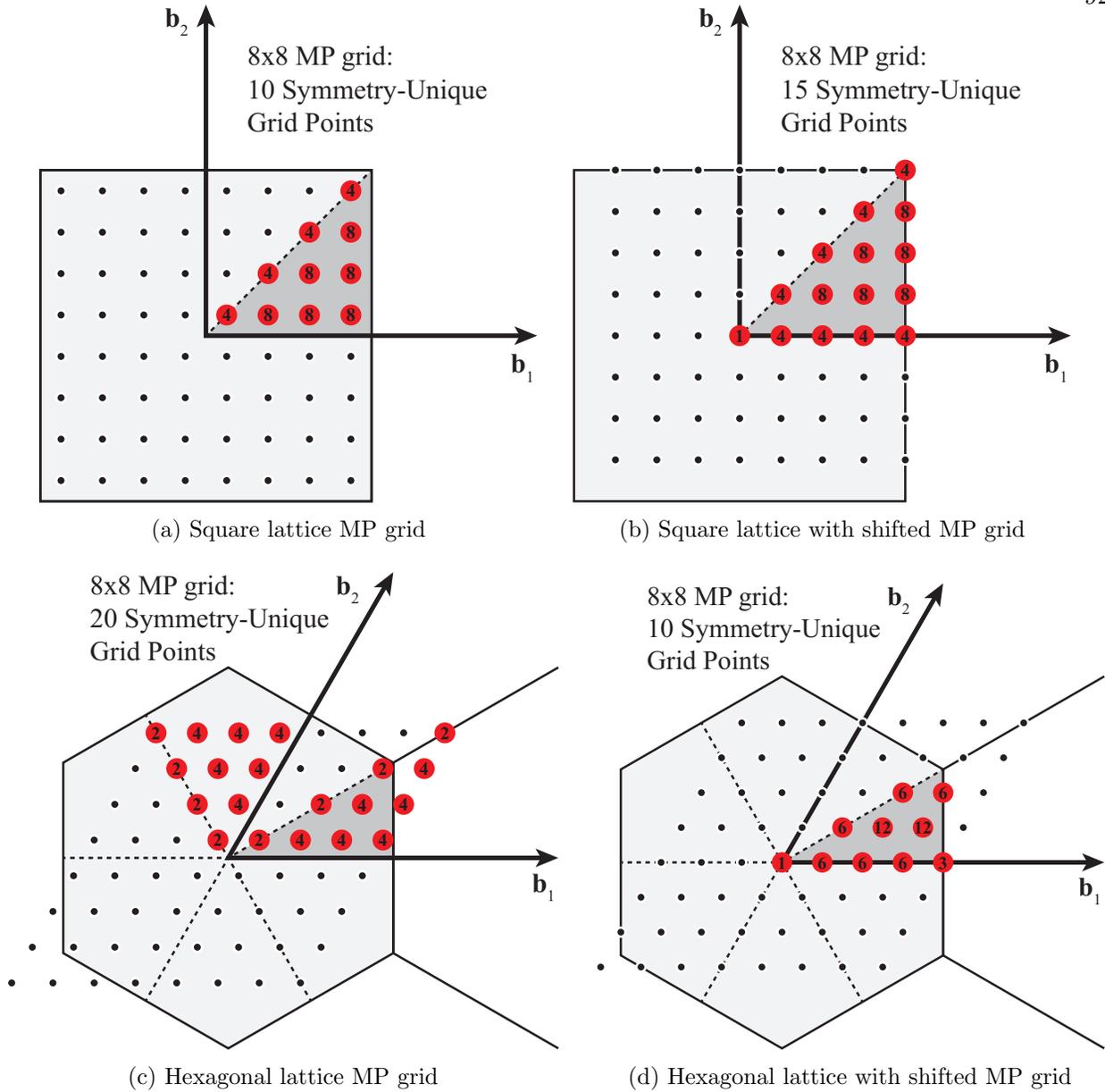


Figure 5.2: MP grids for a square lattice (top), and for a hexagonal lattice (bottom). Grids are shown with no shift (left) and with a half-grid point shift (right). For each diagram, the IBZ is highlighted, and the symmetry reduced set of grid points is shown in red with the corresponding weight multiplier.

Using MP integration, it is possible to approximate the average band energy in the BZ as follows,

$$E_b = \frac{1}{\Omega_{\text{BZ}}} \int_{\text{BZ}} E_b(\mathbf{k}) d\mathbf{k} \approx \frac{1}{N_{\text{MP}}} \sum_{\mathbf{k} \in \mathbf{k}^{\text{MP}}} E_b(\mathbf{k}) = \frac{1}{N_{\text{MP}}} \sum_{\mathbf{k} \in \mathbf{k}^{\text{MPR}}} E_b(\mathbf{k}) w(\mathbf{k}). \quad (5.10)$$

Returning to the charge density, one cannot simply sum the weighted charge densities from

the symmetry-reduced \mathbf{k} points. To illustrate the problem, consider two \mathbf{k} points, \mathbf{k}_1 and \mathbf{k}_2 that are related by a reflection about the y -axis. The charge density computed for \mathbf{k}_1 will also be reflected about the y -axis. This situation is illustrated in Fig. 5.3. This slightly complicates the symmetry-reduced integration because $\rho(\mathbf{x}, \mathbf{k}_1) + \rho(\mathbf{x}, \mathbf{k}_2)$ is not equivalent to $2\rho(\mathbf{x}, \mathbf{k}_1)$. To correctly obtain the integrated electron density, one must first “symmetrize” the charge density obtained at each symmetry-reduced MP point.

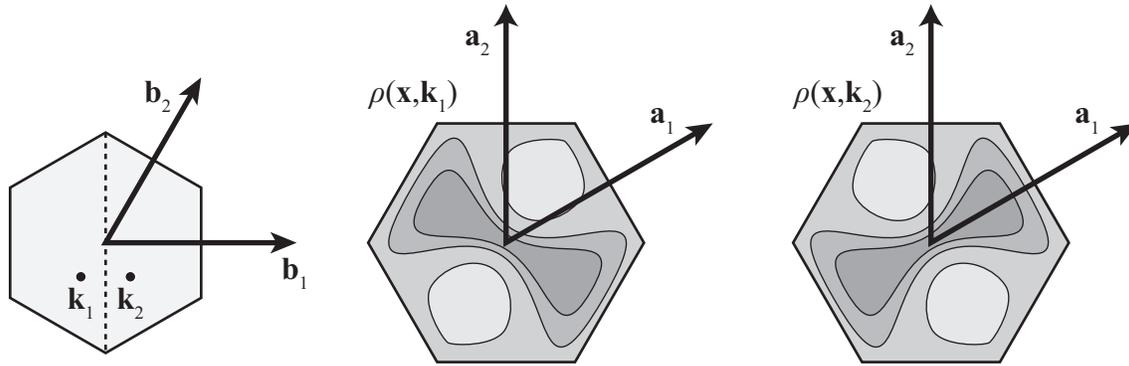


Figure 5.3: Two \mathbf{k} points related by reflection about the y -axis (left) and examples of corresponding electron densities (center and right) also exhibiting a reflection about the y -axis.

To explain the symmetrization, a quick detour into crystallography is necessary. More thorough descriptions can be found in [65, 66]. A crystal’s space group is the set of point group operations (rotations, inversions, reflections and their combinations) paired with translations under which the crystal is invariant. The space group can be represented by a collection of all such point-group operations and corresponding translations $(R_1, \mathbf{t}_1), \dots, (R_{N_g}, \mathbf{t}_{N_g})$. One can then show that the electron density can be transformed as follows,

$$R_j \rho(\mathbf{x}, \mathbf{k}) = \rho(R_j \mathbf{x} + \mathbf{t}_j, \mathbf{k}). \quad (5.11)$$

One can also show that

$$\rho(\mathbf{x}, R_j \mathbf{k}) = R_j \rho(\mathbf{x}, \mathbf{k}). \quad (5.12)$$

Combining these expressions gives

$$\rho(\mathbf{x}, R_j \mathbf{k}) = \rho(R_j \mathbf{x} + \mathbf{t}_j, \mathbf{k}). \quad (5.13)$$

This relationship can be used to obtain a symmetrized electron density, $\bar{\rho}(\mathbf{x}, \mathbf{k})$ that is the average over all symmetry-related \mathbf{k} points,

$$\bar{\rho}(\mathbf{x}, \mathbf{k}) = \frac{1}{N_g} \sum_{j=1}^{N_g} \rho(R_j \mathbf{x} + \mathbf{t}_j, \mathbf{k}). \quad (5.14)$$

With this description it is possible to perform the electron density integration using the reduced MP grid,

$$\rho(\mathbf{x}) = \frac{1}{\Omega_{\text{BZ}}} \int_{\text{BZ}} \rho(\mathbf{x}, \mathbf{k}) d\mathbf{k} \approx \frac{1}{N_{\text{MP}}} \sum_{\mathbf{k} \in \mathbf{k}^{\text{MP}}} E_b(\mathbf{k}) = \frac{1}{N_{\text{MP}}} \sum_{\mathbf{k} \in \mathbf{k}^{\text{MPR}}} \bar{\rho}(\mathbf{x}, \mathbf{k}) w(\mathbf{k}). \quad (5.15)$$

5.2.4 Finite Element Discretization

The solution of Eq. (5.4) can be done for practical multi-atom systems by performing a spatial discretization. The discretization proceeds by expanding the Kohn-Sham orbitals and the Hamiltonian in terms of a set of basis functions. The choice of basis functions can have a significant impact on performance. As mentioned earlier, a common choice for this basis is a truncated set of PWs, but certain limitations have led to the exploration of alternative bases including the FE method. This section gives a very brief derivation of the FE matrices that represent the discretized Kohn-Sham equations. Detailed derivations can be found in [67, 68].

Equation (5.4) is first expressed in weak form by multiplying by an arbitrary test function v^* , and integrating over the unit-cell domain,

$$\int_{\Omega} v^* \left[-\frac{\hbar^2}{2m_e} \nabla^2 \Phi_i(\mathbf{x}) + V_{\text{eff}}(\mathbf{x}) \Phi_i(\mathbf{x}) - E_i \Phi_i(\mathbf{x}) \right] d\mathbf{x} = 0. \quad (5.16)$$

Similar to the expansion of the displacement field in terms of shape functions for the elastic FE method, the expansion of the wave function in terms of the FE shape functions is as follows,

$$\Phi_i(\mathbf{x}) = \mathbf{N} \mathbf{q}_i, \quad (5.17)$$

where

$$\mathbf{N} = [N_1, \dots, N_n], \quad \text{and} \quad \mathbf{q}_i = [q_1, \dots, q_n]^T. \quad (5.18)$$

Integrating the ∇^2 term by parts, and expanding the wave function in terms of the shape functions gives the FE discretized Kohn-Sham equations:

$$(\mathbf{H} - E_i \mathbf{S}) \mathbf{q}_i = 0, \quad (5.19)$$

where the Hamiltonian matrix is given by

$$\mathbf{H} = \int_{\Omega} \left[-\frac{\hbar^2}{2m_e} (\nabla \mathbf{N})^T \nabla \mathbf{N} + V_{eff} \mathbf{N}^T \mathbf{N} \right] d\mathbf{x}, \quad (5.20)$$

and the structure matrix is given by

$$\mathbf{S} = \int_{\Omega} \mathbf{N}^T \mathbf{N} d\mathbf{x}. \quad (5.21)$$

Remember from the elastic FE discussion that the shape functions associated with each element are completely local (i.e., they are nonzero outside of that element). Thus, integration is performed over individual element domains using Gaussian quadrature, and then the element matrices are assembled into the global FE matrices by summing over all elements. Those who are familiar with the PW Kohn-Sham formulation may be surprised to see that the FE Kohn-Sham equations of Eq. (5.19) take the form of a generalized eigenvalue problem. This occurs because unlike the PW basis functions, the FE shape functions are in general not orthogonal.

At this point, Bloch's solution has not yet been considered. To obtain a Bloch-periodic solution, periodic BCs can be applied by forming the Bloch-periodicity transformation, $\mathbf{P}(\mathbf{k})$, of Eq. (2.20). The Bloch-periodic Kohn-Sham equations are then

$$\left(\hat{\mathbf{H}}(\mathbf{k}) - E_i \hat{\mathbf{S}}(\mathbf{k}) \right) \hat{\mathbf{q}}_i = 0. \quad (5.22)$$

5.2.5 Bloch Mode Synthesis

The FE discretized Kohn-Sham equations shown in Eq. (5.19) closely resemble the FE equations describing elasticity,

$$(\mathbf{K} - \omega_i^2 \mathbf{M}) \mathbf{q}_i, \quad (5.23)$$

The Hamiltonian matrix is analogous to the stiffness matrix, the structure matrix is analogous to the mass matrix, and the energy eigenvalues are analogous to the squared natural frequencies.

Thus, it is no great leap to expect BMS to apply in the same way as it does to the equations of elasticity. There are some fundamental differences however between the matrices. The stiffness matrix is symmetric positive semidefinite, whereas the Hamiltonian matrix is symmetric indefinite. The structure matrix, like the mass matrix, is symmetric positive definite. As a result, the energy eigenvalues of the discrete Kohn-Sham equations can be both positive and negative, in contrast with the elasticity eigenvalues which cannot be negative. We will see that these differences do not impact the ability of the BMS basis to faithfully reproduce the full-model solutions. Some of the linear algebra algorithms used to perform the BMS reduction may need to change however. For example, the constraint mode calculation performed with the Hamiltonian matrix,

$$\Psi = -\mathbf{H}_{\text{II}}^{-1}\mathbf{H}_{\text{IA}},^{\ddagger} \quad (5.24)$$

requires a different linear solver algorithm than the elastic constraint mode calculation because \mathbf{H}_{II} is indefinite whereas \mathbf{K}_{II} is positive definite. Rather than the very efficient Cholesky solver used for positive definite system solutions, a more general (and less efficient) LDL solver must be used. Despite this and a few other small differences between the elastic and electronic problems, the performance of BMS is qualitatively quite similar between the two.

5.3 Numerical Examples

In order to evaluate the performance of BMS, electronic-structure calculations are performed for crystalline silicon. Rather than performing the electronic-structure calculations with the commonly used 8-atom conventional unit cell shown in Fig. 5.4a, the calculations are performed with the primitive cell shown in Fig. 5.4b. This takes advantage of the flexibility of the FE mesh to model more difficult geometries. The effective potential is approximated using an empirical pseudopotential [69]. A set of isosurface contours is plotted for this pseudopotential in Fig. 5.5.

[‡]There is an unfortunate similarity in notation between the constraint modes, Ψ , and the wavefunction, Ψ , as well as normal modes, Φ , and Kohn-Sham orbitals, Φ . This notation is used to be consistent with notation in literature with the hope that the context is sufficient to prevent confusion.

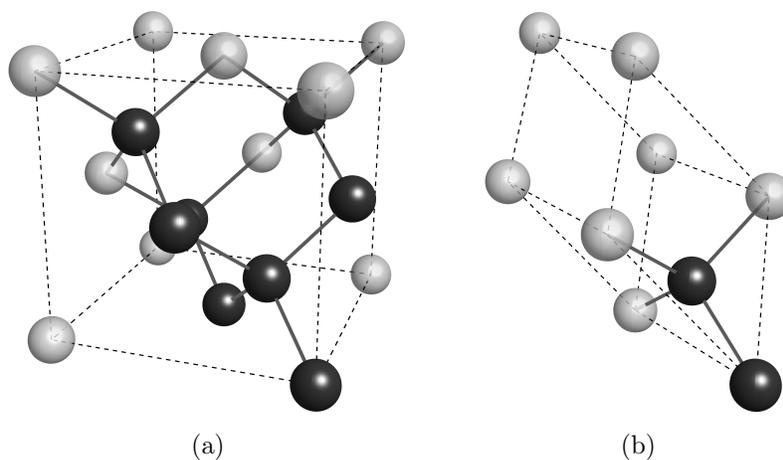


Figure 5.4: Silicon crystalline unit cells. The subfigure on the left (a) shows the 8-atom conventional cell and the subfigure on the right (b) shows the 2-atom primitive cell

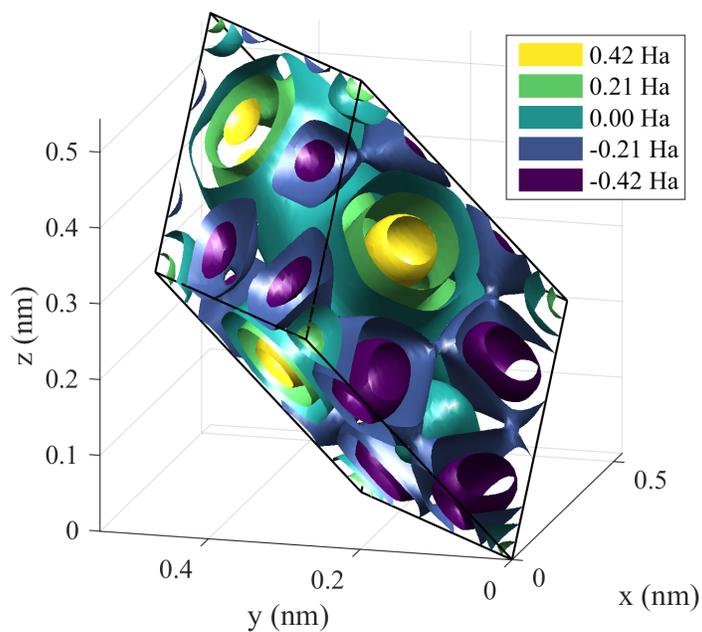


Figure 5.5: Empirical pseudopotential isocontours for silicon primitive cell

In previous chapters, the error in each BMS model is evaluated with respect to the base FE model from which it was created. In this chapter a slightly different approach is taken. Rather than evaluating error with respect to the base FE model, error is evaluated with respect to a highly converged reference model. The reference model consists of a $5 \times 5 \times 5$ regular mesh of hexic (6th

order) elements containing 27,000 DOF. A $20 \times 20 \times 20$ MP rule (with 256 symmetry-reduced \mathbf{k} points) is used for BZ integration of the reference model.

Assuming that numerical error in the reference model is small enough to disregard, the only error in the highly converged reference model is due to the pseudopotential. Since this pseudopotential is obtained from empirical results, the reference model is numerically exact with respect to these measurements. An error tolerance of 1×10^{-4} Hartrees per atom (Ha/atom) is used throughout this chapter as the maximum tolerable error in the average (integrated) band energy.

Before analyzing the performance of BMS, a base case must be selected as a foundation from which to form the BMS models. Each case is defined by the FE mesh and the MP integration rule. Since the pseudopotential is relatively smooth, regular (uniform) meshes are used in all cases. The elements use Lagrange polynomial shape functions making it easy to switch the element order. Since the mesh is completely uniform, it is completely defined by the element type and the model size (i.e. number of DOFs). The strategy for selecting the element type, model size, and MP rule is to vary each and evaluate the error with respect to the reference mesh as well as the computation time necessary for each base calculation. Then, the cases that achieve the error tolerance in the shortest amount of time are selected. To first narrow down the space of possible combinations, a preliminary test is performed by computing the band energy with 13824 DOF models comprised of linear, quadratic, cubic, and quartic elements, and using MP rules ranging from $1 \times 1 \times 1$ to $15 \times 15 \times 15$. The band energy error is plotted with respect to the MP rule in Fig. 5.6. Based on the error values in this preliminary study, only MP rules from $7 \times 7 \times 7$ to $10 \times 10 \times 10$ are considered. Furthermore, only cubic, quartic and quintic elements are considered.

Figure 5.7 shows the band energy error versus computation time for various base cases. The MP rule is incremented from $7 \times 7 \times 7$ to $10 \times 10 \times 10$, element order is allowed to be cubic, quartic, and quintic, and model sizes from 100 to 13824 DOFs. Based on these results, four base cases are selected from which to evaluate BMS performance. These are highlighted in Fig. 5.7 and also detailed in Table 5.1. Base case “A” provides the fastest possible FE calculation at approximately 88 seconds, but it is already so close to the error tolerance that the BMS error must be extremely

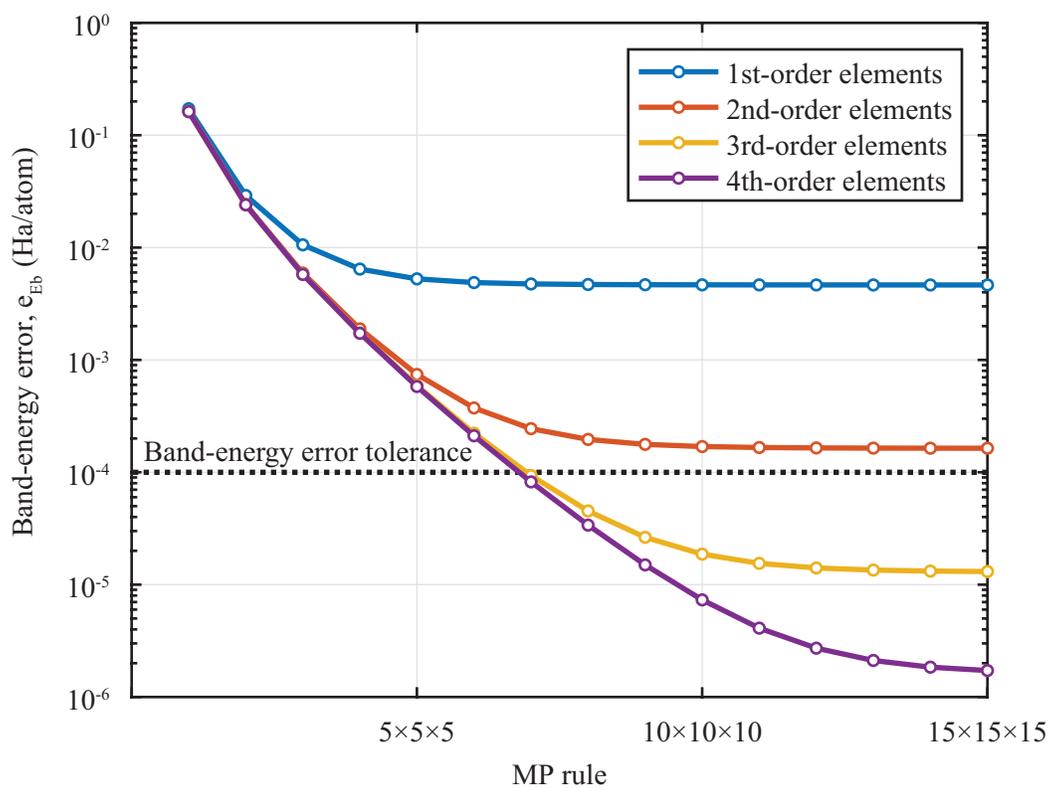


Figure 5.6: Error in band energy with respect to MP rule for 13,824 DOF meshes consisting of linear, quadratic, and quartic elements.

small. This makes it an impractical base for BMS reduction, but it is useful for comparing the reduced and full-model performance.

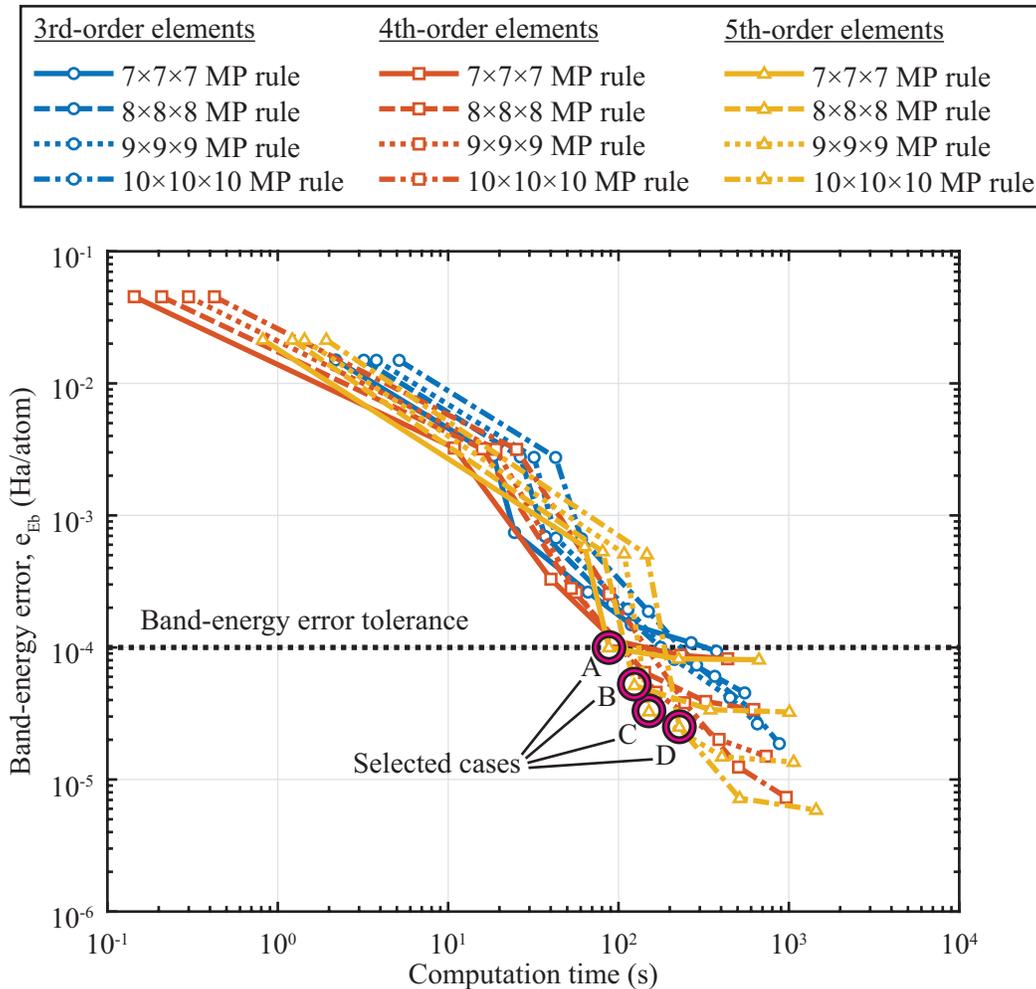


Figure 5.7: Error in band energy with respect to computation time for various calculation cases

A series of BMS model reductions with varying number of FI modes are created for each base case and used to solve for the band energy of silicon. The resulting band energy error and computation time are plotted in Fig. 5.8. As mentioned earlier, the fastest full FE calculation to satisfy the error tolerance takes about 88 seconds. In comparison, the fastest BMS calculation to satisfy the error tolerance takes 18 seconds. This represents a speedup by a factor of about 4.9 times. Note that although the BMS models always have higher error than their respective base models, the best (fastest without exceeding error tolerance) BMS model actually has lower error

Table 5.1: Reference and base FE model parameters

Case	Element order	Mesh Size <# DOFs>	MP rule <# unique \mathbf{k} points>	Error (Ha/atom)	Computation time (s)
Reference	6	$5 \times 5 \times 5$ <27,000>	$20 \times 20 \times 20$ <256>	-	-
A	5	$3 \times 3 \times 3$ <3375>	$7 \times 7 \times 7$ <20>	9.98×10^{-5}	87.9
B	5	$3 \times 3 \times 3$ <3375>	$8 \times 8 \times 8$ <29>	5.16×10^{-5}	123.9
C	5	$3 \times 3 \times 3$ <3375>	$9 \times 9 \times 9$ <35>	3.27×10^{-5}	150.9
D	5	$3 \times 3 \times 3$ <3375>	$10 \times 10 \times 10$ <47>	2.50×10^{-5}	227.6

than the best base FE model.

Since self-consistent iteration represents the main computational burden of the electronic-structure calculation, it makes sense to use MP integration which forms the core of each step in the iteration to study performance. After the iterative process has converged however, the band-structure diagram is obtained by tracing a path through the BZ as was done with elastic band-structure calculations. The band-structure diagram for silicon is computed using the 3375 DOF quintic-element model used for base cases A-D, and also with a BMS reduction of this model containing 10 FI modes and 283 interface modes. The band-structure is obtained by discretizing the L- Γ -X-K- Γ path through the BZ into 65 \mathbf{k} points and solving for the first eight energy levels at each point. The resulting energy bands are plotted in Fig. 5.9. The BMS reduction provides a speedup by over a factor of 10 and the energy bands show no noticeable deviation from the full model.

For points A, B, and C, highlighted in Fig. 5.9, the local wave function is used to compute a local charge density (by squaring the complex magnitude at every point in space). The resulting charge densities are once again shown in Fig. 5.10. As with the band-structure plot, it is readily apparent that the BMS model faithfully reproduces the full-model charge densities.

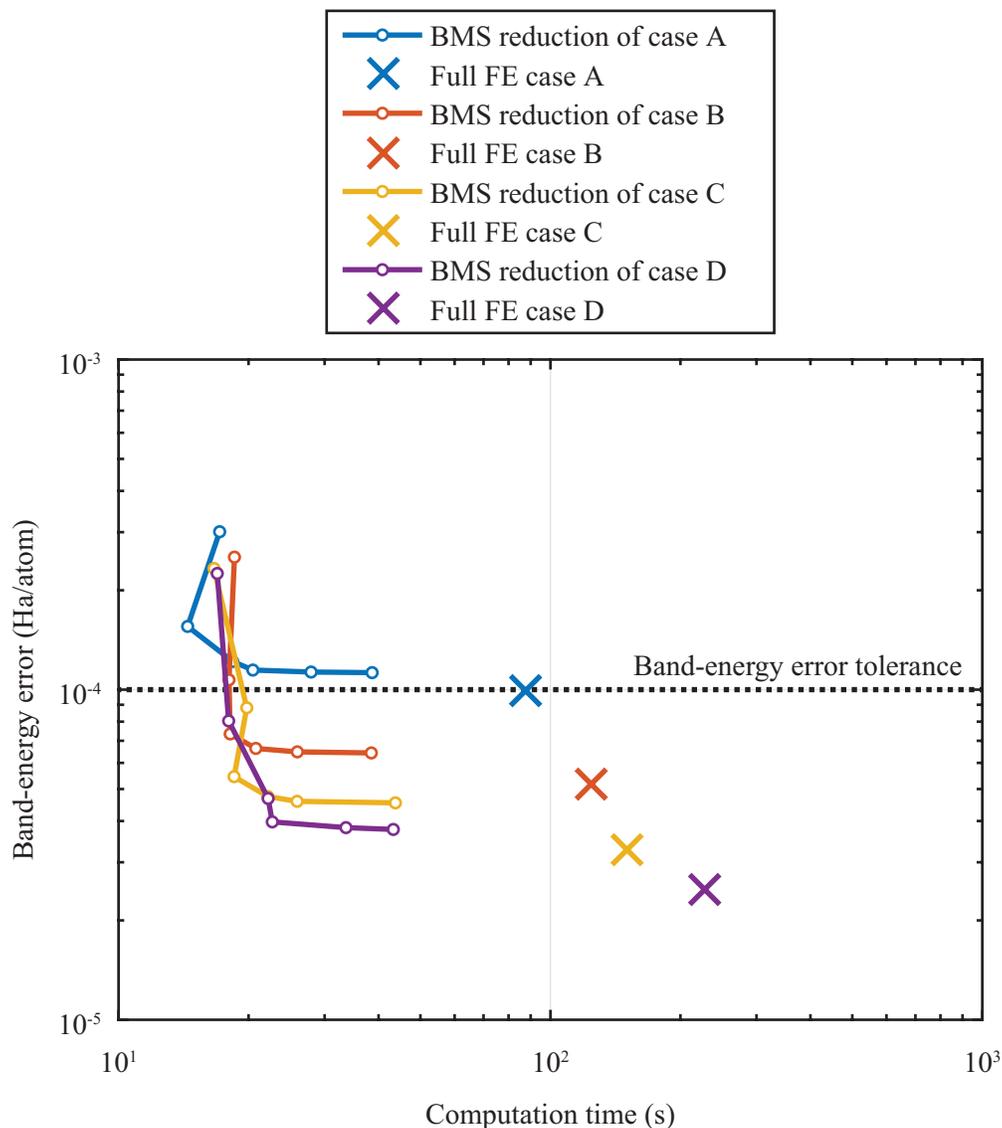


Figure 5.8: Error in band energy with respect to computation time for various BMS model reductions of the base cases in Table 5.1.

5.4 Summary

This chapter introduced the fundamentals of electronic-structure calculation, including an overview of Schrödinger's equation and its simplification via DFT to the Kohn Sham equations. Then, the application of FE discretization to the Kohn Sham equations was described. Finally, the extension of BMS to electronic-structure calculations was demonstrated using a simple silicon unit cell (modeled using an empirical pseudopotential). The results showed speedups in computation

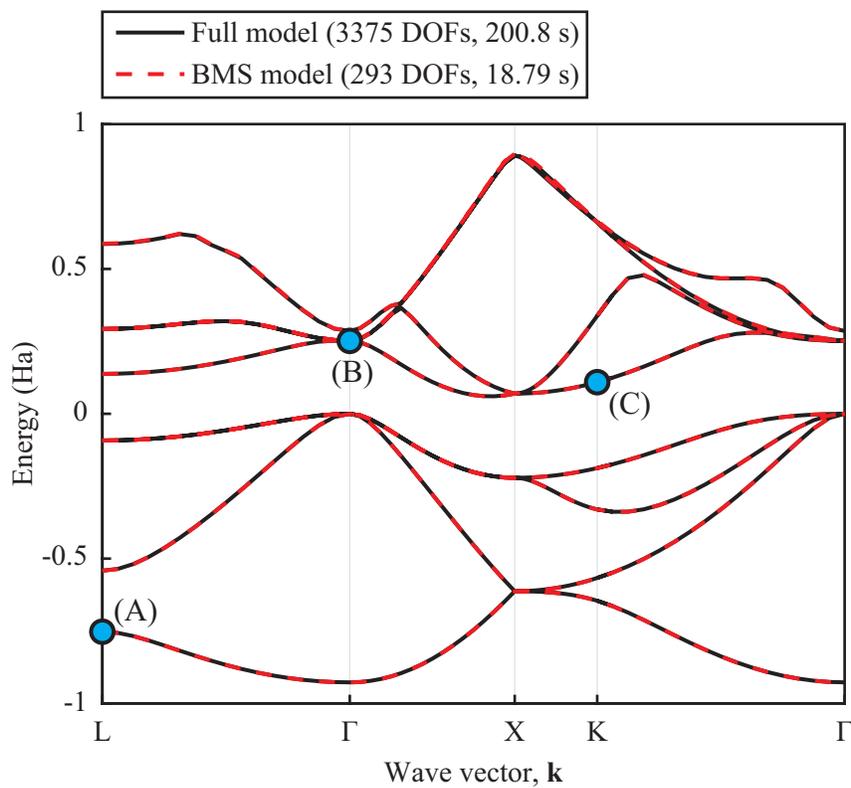


Figure 5.9: Band-energy diagram obtained with full FE model and with BMS reduced-order model. The band energy is obtained with 65 \mathbf{k} points

time of over an order of magnitude compared to the full FE model, while retaining highly accurate solutions with less than 1×10^{-4} Ha/atom error in integrated band energy.

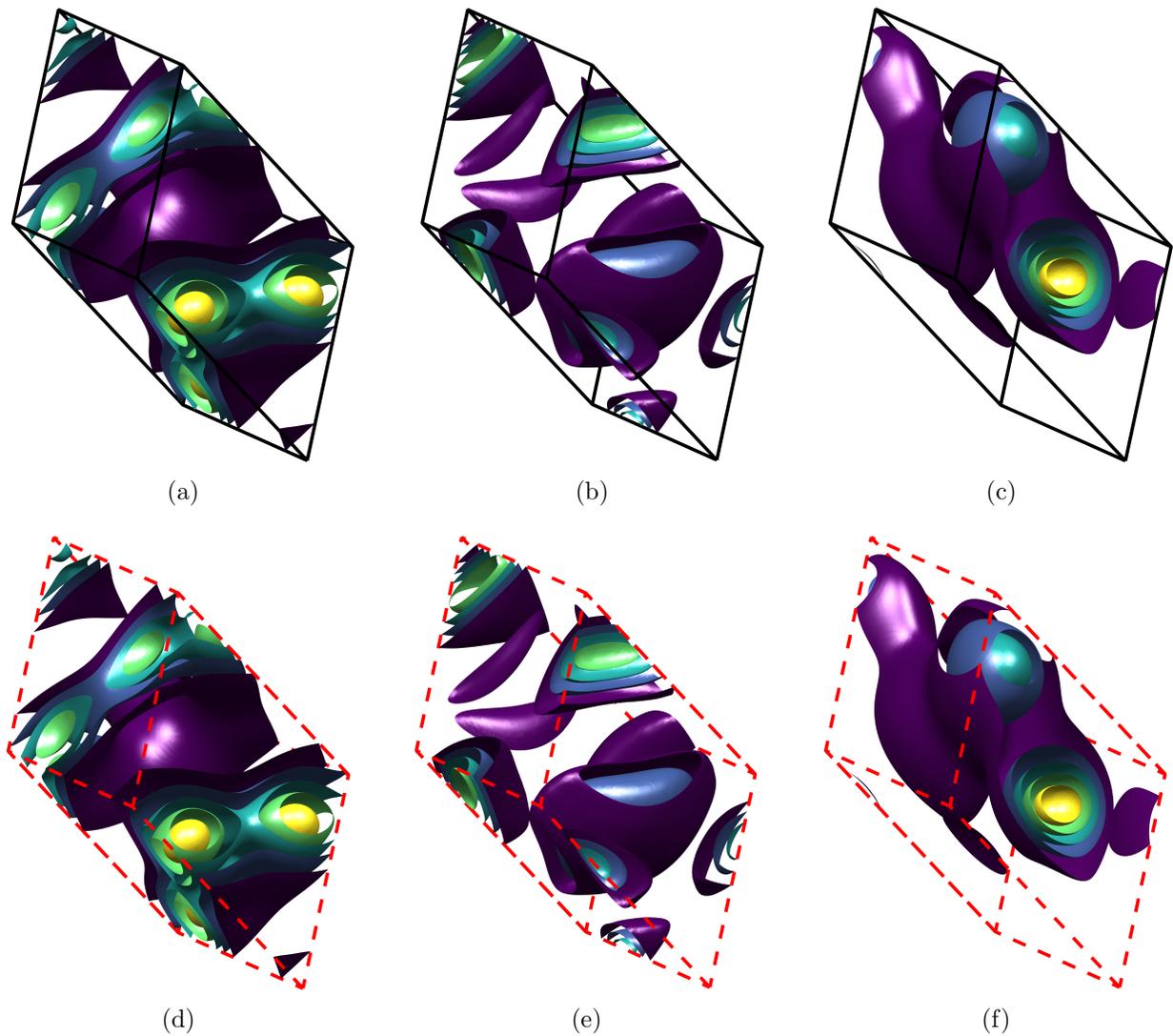


Figure 5.10: Comparison of charge densities between full model (top row) and BMS model (bottom row). Each charge density is computed for a single band-structure solution. The charge densities correspond to points A (left column), B (middle column), and C (right column) in Fig. 5.9.

Chapter 6

Extension of Bloch Mode Synthesis to Full-Spectrum Phonon Calculations

6.1 Introduction

The term “phonon” first emerged in the description of vibrations of atomic scale lattices (i.e., conventional crystals). These quantized vibration packets, alongside electrons, are the main mechanism for heat transfer in materials. A material’s thermal properties are determined by the constituent atoms, and also the arrangement of these atoms. For example, amorphous silicon which does not have a periodic arrangement of atoms has much lower thermal conductivity than crystalline silicon. The ability to tune the crystalline structure can thus be used to engineer thermal properties. An extreme example of this is nanostructuring which has emerged as a promising way to design materials with desirable thermal properties [60, 70].

A common framework for analyzing thermal transport in crystals is lattice dynamics [71]. The main approximation made in lattice dynamics calculations is that the atomic interactions are harmonic. This gives a linear system of equations that can be solved for the band structure. The Boltzmann transport equation can then be used to find the thermal conductivity. This includes an integral over the BZ, and also over all energy carrying bands.

Continuum models that have been discretized (e.g., by the FE method) should only be trusted to accurately obtain low-frequency solutions. The highest branches are not physically meaningful because they are strongly affected by the discretization. Models of naturally discrete systems like spring-mass assemblies or atomic crystals do not suffer from this pathology. Every band computed in a lattice dynamics calculation is physically meaningful and in principle capable

of carrying energy. Thus, a complete calculation of the thermal properties requires a full-spectrum band-structure calculation with all branches computed.

Since the band-structure calculation discussed in previous chapters sought only to obtain the lowest frequency/energy branches, iterative methods were used. Iterative eigenvalue solution algorithms are very fast when the number of eigenpairs, k^\ddagger , is relatively small ($k < \approx 200$). As k is increased however, iterative methods slow down considerably. A faster alternative is to use a direct method, but as seen in Fig. 2.14, the quadratic algorithm scales approximately cubically with model size, so unit cells with many atoms quickly become intractable to analyze.

This chapter establishes several techniques for speeding up the full-spectrum band-structure problem. First, the model creation, which can itself be quite costly already, is targeted. Two new contributions are introduced for speeding up atomic-scale model creation. Then, the focus is turned to the eigenvalue solver. A windowing approach is introduced in order to reduce the computational complexity of obtaining full-spectrum eigenvalue solutions. Finally, the BMS formulation is modified so it can also be applied to a windowed calculation.

6.2 Fast System Assembly for Empirical Interatomic Potentials

The forces holding atoms in crystals together are primarily due to the complex electrostatic interactions between the atomic nuclei and their electrons. Describing these interactions can be done using DFT as described in chapter 5, but in the current chapter, an empirical interatomic potential developed by Tersoff, [72] is used to model the forces involved in atomic interactions. This empirical potential approximates bonds as 3-body interactions. Each bond can be likened to a complex spring whose stiffness depends on the locations of nearby atoms. A cutoff radius is also specified such that first- and second-nearest neighbor interactions are included. The benefit of this truncation is that the system of equations is sparse. The stiffness matrix is formed by differentiating the potential energy with respect to the atomic positions. This can be done analytically, but in

[‡] k was used earlier to describe the magnitude of the wave vector, but in this chapter it denotes the number of computed eigenpairs. \mathbf{k} still denotes the wave vector.

practice is done numerically by applying small perturbations to each atom and computing the change in energy using the empirical potential. For each atom in the system, the interaction between every possible neighbor must be considered to compute the stiffness matrix. For periodic materials, a new stiffness matrix is formed for every wave vector. This process is computationally burdensome and limits the size of unit cells that can be considered.

6.2.1 Quasi-Free Unit Cell

Rather than computing a new model for every wave vector, it is possible to create a quasi-free description of the unit cell. Then, a Bloch BC transformation can be applied for any wave vector in order to quickly obtain the Bloch-periodic model. The first step in creating the quasi-free description is to pad the unit cell with a set of phantom atom locations. This is illustrated for a silicon thin film model in Fig. 6.1. The stiffness matrix assembly then proceeds as before by looping through all non-phantom atoms and computing the stiffness of bonds with neighbor atoms. Whenever this results in an atom on an edge of the unit cell being bonded to a neighbor atom that is an adjacent unit cell, the corresponding stiffness terms are added into stiffness matrix locations that have been allocated for the phantom atoms. In the traditional assembly, every bond's stiffness is multiplied by a PW term corresponding to that bond's direction and length. In the present approach, no PW terms are included at this time.

Note that the three-body bond stiffnesses are computed under the assumption that neighboring unit-cell atoms *are* present. This subtle detail is necessary if the modified approach is to produce the identical stiffness matrix to the traditional approach once Bloch BCs are applied. The stiffness terms computed with three-body interactions are not symmetric for a single bond (this is a departure from the way that conventional Hookean springs work). The symmetry of the bonds in space allows the final assembled stiffness to become symmetric. In the present approach, the spatial symmetry of the atomic locations is temporarily broken (until Bloch BCs are applied). This results in a non-symmetric stiffness matrix which is not equivalent to a similarly sized nanodot. If the modes of vibration are computed for this non-symmetric model, no rigid body modes (i.e., free

translation or rotation modes) will be found. Thus, it has different characteristics than a nanodot model with an identical configuration of atoms. This is why this unit cell model is referred to as quasi-free.

With this model it is possible to apply Bloch BCs for any wave vector without recomputing any stiffnesses from the empirical interatomic potential, thus reducing the computational time for producing the model by a factor of the number of \mathbf{k} points.

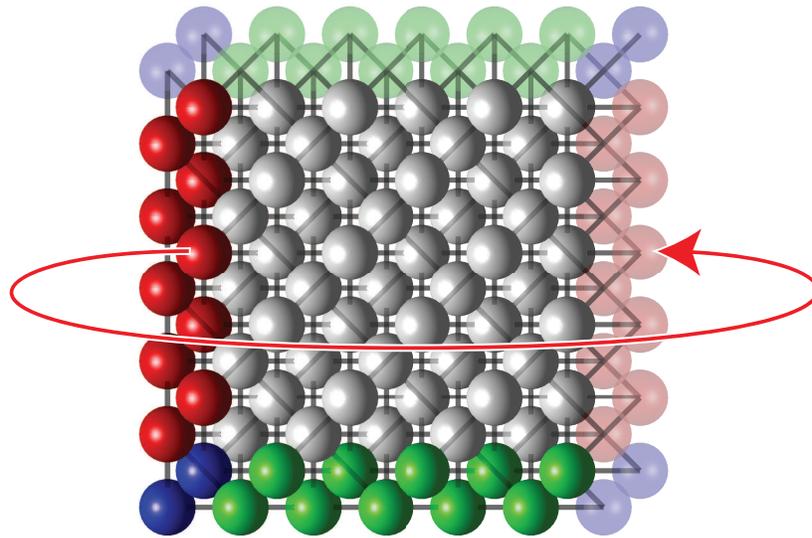


Figure 6.1: Silicon thin film unit cell viewed orthogonal to the thin film plane with a layer of phantom atoms added. Phantom atoms are shown in transparent colors corresponding to their periodic counterparts within the unit cell.

6.2.2 Building Block Assembly Approach

A nanostructured silicon unit cell can be built up by using the silicon conventional cell as a building block. The conventional cell (shown in Fig. 5.4a) can be patterned to create large features relative to the individual atom spacing. For a unit cell that has been built in this way it makes sense to try to build the stiffness matrix using the same idea. Due to the three-body interaction between atoms, the stiffness matrix for any conventional cell depends on the presence of all neighboring atoms. This means that for each unique combination of neighboring cells, a slightly different stiffness matrix will be obtained. Picture a $3 \times 3 \times 3$ cube of conventional cells representing

a single conventional cell with its 26 neighbors. Since each neighboring conventional cell can either be occupied or vacant, there are 2^{26} unique neighbor-cell configurations. Typically however, a small subset of these configurations need to be considered. The unit cell for a nanopillared thin film shown in Fig. 6.2 for example only contains 59 unique configurations of neighboring conventional cells.

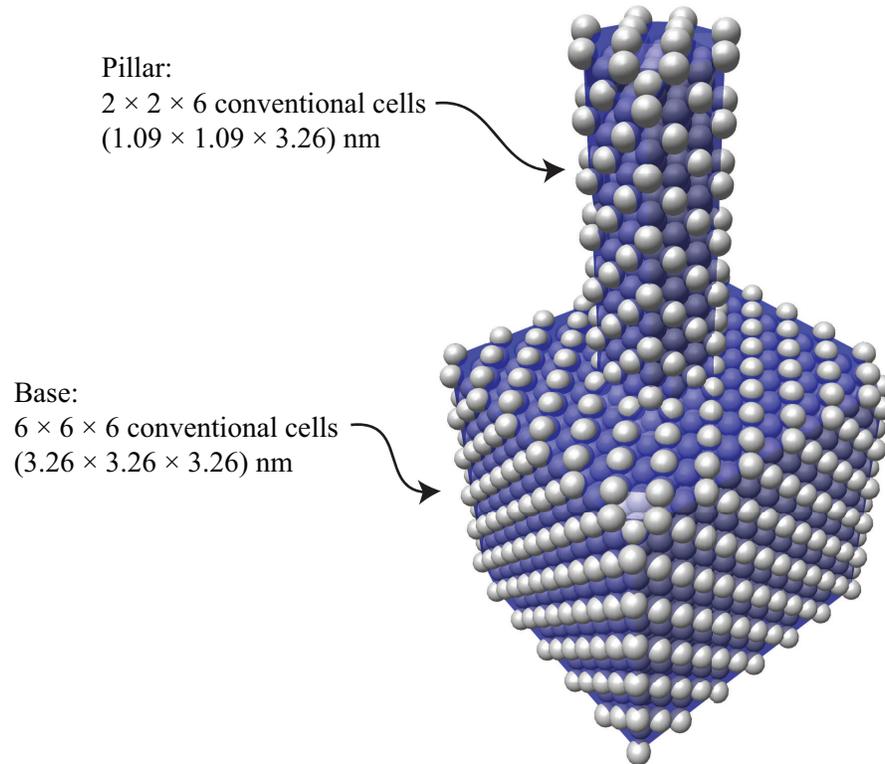


Figure 6.2: Silicon nano-pillared thin film unit cell.

Each unique configuration of neighboring cells constitutes a building block that can be used to assemble the full model stiffness matrix. For each of the eight atoms within a building block, and for every bond that that atom makes with any other atoms (inside or outside the building block), a 3×3 stiffness matrix is stored. During the stiffness matrix assembly, anytime a new building block is encountered (this is checked by testing the presence of neighboring cells), the empirical interatomic potential is used to obtain the building block's bond stiffness matrices. This building block is then stored for future use. Anytime a building block has already been encountered, the bond stiffness matrices are simply added into the global stiffness matrix via the direct stiffness

method used commonly in FE methods.

This building strategy avoids many redundant computations and offers much faster calculation of the stiffness matrix. The method is slightly less general than the traditional approach because it does not allow for the atomic coordinates to be changed even slightly (i.e., every instance of a building block must be identical). This precludes any relaxation wherein the atoms are allowed to settle to new coordinates in order to minimize the total interatomic potential energy. It may be possible however to obtain a good approximation to the relaxed system if the building block approach is only applied to internal building blocks and any blocks with free surfaces are allowed to relax as normal. This is left for further investigation.

6.3 Full-Spectrum Eigenvalue Solution by Spectrum Slicing

The full-spectrum band-structure calculation is performed by computing all of the eigenvalue solutions at every \mathbf{k} point along a path following the boundary of the IBZ. Although iterative eigenvalue solvers can obtain multiple eigenpairs at a time, they do not perform well when the number of eigenpairs is very large. Thus, the traditional approach is to use direct solvers which are designed to compute the full eigenvalue spectrum. The direct solvers that are commonly used for hermitian problems are based on the QR decomposition and thus scale cubically with matrix size. This severely limits the size of systems that can feasibly be analyzed and often does not take advantage of sparsity.

An alternative method for obtaining the full spectrum of sparse hermitian matrices is to use spectrum slicing [73, 74]. Since all eigenvalues lie on the real axis, the main idea is to split the real axis into a number of windows (or spectral slices) and use an iterative solver to obtain solutions in that window. For a matrix of size $(n \times n)$ the shift-and-invert Lanczos method (implemented in Matlab's `eigs`) can obtain a fixed number of eigenpairs in $\mathcal{O}(n)$ computation time. The total number of windows needed should be proportional to n as well, so the full-spectrum calculation requires $\mathcal{O}(n^2)$ computation time. For large matrices, the savings in computation time over direct methods can be quite significant. An additional benefit of spectrum slicing calculations is that they

are simple to parallelize [75].

To achieve good performance with a spectrum slicing algorithm the windows must be carefully defined. The algorithm presented here is simple and undoubtedly can be improved, but it already shows promising performance. A window can be defined as a frequency range, or it can be defined by a central frequency with a specified number of nearest eigenvalues. The latter is chosen here so that the number of eigenvalues computed for each window is constant. The spectrum slicing algorithm is then implemented as follows:

- (1) Compute the k eigenpairs with largest magnitude eigenvalue and split them into a negative and a positive set: $\lambda_1^1 \leq \dots \leq \lambda_j^2 < 0$ and $0 \leq \lambda_1^2 \leq \dots \leq \lambda_{k-j}^2$.
- (2) If either set is empty, compute the k eigenpairs with smallest magnitude eigenvalue and place them in that set. The two sets represent the endpoints of the eigenvalue spectrum.
- (3) Find the gaps in the eigenvalue spectrum by ordering the sets and comparing the maximum of each set with the minimum of the next set above it.
- (4) If the maximum gap size is positive, use the center of the maximum gap as the shift about which to compute the next set of k eigenpairs.
- (5) Repeat steps three and four until the maximum gap size is below a negative tolerance. Requiring that there is a minimum amount of overlap between all of the sets ensures that no degenerate eigenvalues are lost between sets.
- (6) Step through all computed sets and keep only a single copy of any eigenpairs that overlap with neighboring sets*.

The spectrum slicing algorithm is illustrated by calculating the eigenvalues of a 2D unit cell made up of springs and masses arranged in a “plus” shaped configuration shown in Fig. 6.3.

*For an eigenvalue of degeneracy m that has some overlap between two sets, the overlapping eigenvectors should be grouped together and used to produce m orthogonal vectors

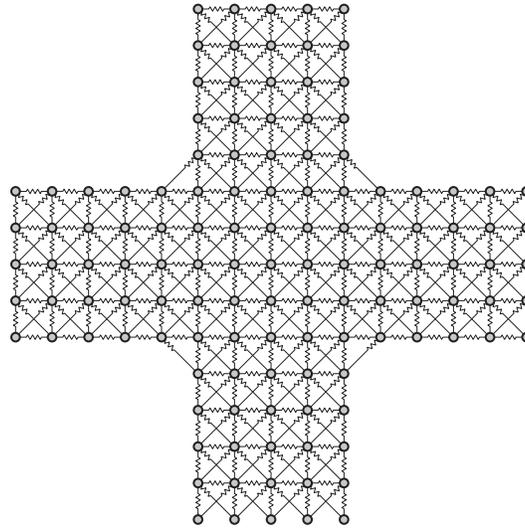


Figure 6.3: 2D “plus” shaped spring-mass unit cell used to investigate spectrum slicing algorithm

Figure 6.4 shows the iterative solution of the full eigenvalue spectrum by consecutively identifying the maximum gap and centering the next eigenvalue window there until overlap is achieved between all sets.

To test the scaling of the spectrum slicing algorithm, the number of masses in the plus-shaped spring-mass unit cell is incremented from 115 to 10035 and the full eigenvalue spectrum is calculated for a single \mathbf{k} point (the M point). In all cases, k , the number of eigenvalues computed per window, is 30. The results of the scaling analysis are summarized in table 6.1. The average number of eigenvalues kept per window can be used to gauge the amount of overlap between neighboring spectrum slices. This stays remarkably constant as the model size increases. The timing results for the direct algorithm (Matlab’s `eig`) are also included for comparison. In all cases, the error between the two is extremely small.

The timing results plotted in Fig. 6.5 versus model size. Best fit lines are added in order to elucidate the computational order. The direct algorithm is approximately cubic and the spectrum slicing algorithm is approximately quadratic. The two algorithms seem to break even around a model size of $n = 20,000$, but this result is not general. The break-even point will be model dependent because the number of non-zero elements per row of the stiffness matrix will affect each

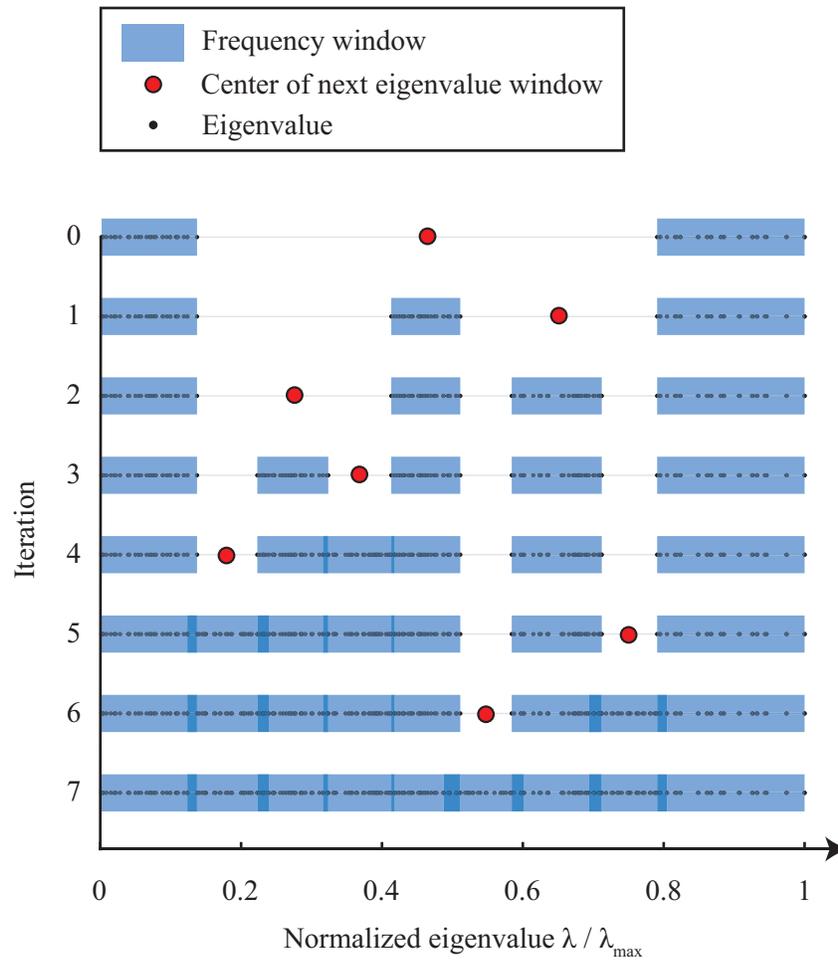


Figure 6.4: Illustration of spectrum slicing algorithm showing consecutive windows being computed

Table 6.1: Scaling analysis results for full-spectrum eigenvalue calculation using spectrum slicing algorithm and direct algorithm.

Number of masses	Matrix size, n	Number of windows	Avg. number of eigenvalues kept per window	Spectrum slicing algorithm computation time (s)	Direct algorithm computation time (s)	Maximum relative eigenvalue error (%)
115	230	9	25.6	2.16	0.06	1.57e-12
480	960	49	19.6	16.04	0.98	3.35e-12
1095	2190	115	19.0	62.00	7.54	9.03e-12
1960	3920	202	19.4	176.33	50.30	1.76e-11
3075	6150	310	19.8	444.14	177.84	1.48e-10
4440	8880	450	19.7	862.68	513.77	1.26e-10
6055	12110	619	19.6	1698.63	1107.85	6.94e-11
7920	15840	824	19.2	2974.36	2318.21	1.94e-10
10035	20070	1017	19.7	5002.60	4102.90	3.59e-10

algorithm differently.

6.4 Windowed Bloch Mode Synthesis for Reduced-Order Spectrum Slicing

The ideas presented in the previous section are not specific to band-structure calculations. Spectral slicing can be applied to any hermitian eigenvalue problem. Ideally, for full-spectrum band-structure calculations one would take advantage of spectral slicing and also use model-order reduction to speed up the traversal of the BZ. In order for BMS to be used in conjunction with spectral slicing, it must first be modified to allow for arbitrary frequency windows to be considered.

As discussed earlier, the accuracy of mode-synthesis methods deteriorates as one moves to higher frequencies. Thus, one cannot simply set a centering frequency outside the accurate range of the reduce-order model and hope to obtain the correct results. For the HCB model, one could simply increase the number of FI modes used in the interior to extend the accurate range, but this defeats the purpose of the reduced-order model. Ideally, one would only need to keep the fixed-interface modes that are near the centering frequency, but this doesn't work either. To understand why, it is useful to think of the HCB representation as a static (Guyan) condensation whose accurate frequency range can be extended further and further by adding more fixed-interface modes.

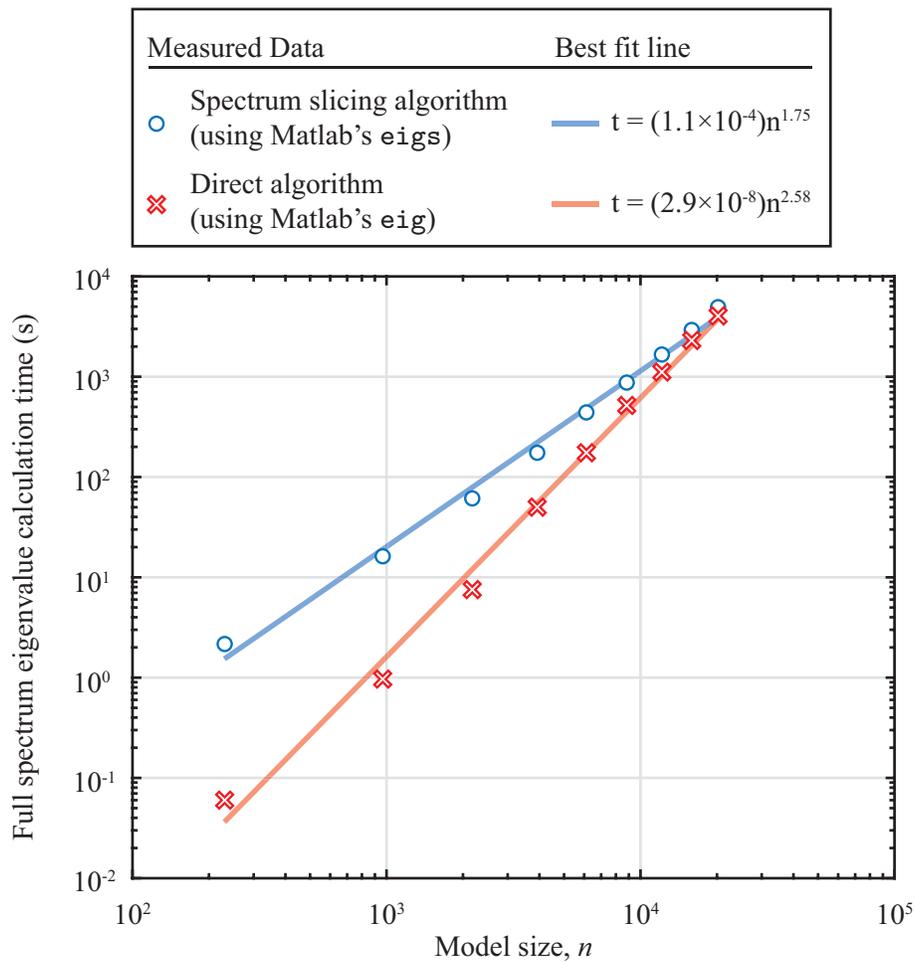


Figure 6.5: Timing results for full-spectrum eigenvalue calculations on increasing model sizes. The computational complexity is observed to be approximately $\mathcal{O}(n^3)$ for direct eigenvalue solution, and $\mathcal{O}(n^2)$ for the spectrum slicing algorithm.

This gives a clue as to how the HCB representation must be modified in order to obtain accurate solutions in an arbitrary frequency range while keeping only FI modes near the central frequency. Considering the partitioned free equations of motion,

$$\left(\begin{bmatrix} \mathbf{K}_{II} & \mathbf{K}_{IA} \\ \mathbf{K}_{AI} & \mathbf{K}_{AA} \end{bmatrix} - \omega^2 \begin{bmatrix} \mathbf{M}_{II} & \mathbf{M}_{IA} \\ \mathbf{M}_{AI} & \mathbf{M}_{AA} \end{bmatrix} \right) \begin{Bmatrix} \mathbf{q}_I \\ \mathbf{q}_A \end{Bmatrix} = 0, \quad (6.1)$$

the constraint modes are typically obtained by solving for the static response (by setting $\omega = 0$), of the interior due to consecutive unit displacements of the interface:

$$\Psi = -\mathbf{K}_{II}^{-1} \mathbf{K}_{IA}. \quad (6.2)$$

Alternatively, a dynamic version of the constraint modes can be created that by setting the frequency equal to a centering frequency $\omega = \omega_c$, and then solving for the interior displacements due to unit amplitudes at the interface:

$$\Psi_{QS} = -(\mathbf{K}_{II} - \omega_c^2 \mathbf{M}_{II})^{-1} (\mathbf{K}_{IA} - \omega_c^2 \mathbf{M}_{IA}). \quad (6.3)$$

These modified constraint modes, first introduced in reference[76], are termed “quasi-static” because they arise from an extension of static mode compensation techniques. Using only the quasi-static constraint modes as a basis (without adding any FI modes), is equivalent to performing a dynamic condensation about the centering frequency. Recall that dynamic condensation produces a model that is exact at the centering frequency and that deteriorates in accuracy away from the centering frequency. Just like the traditional HCB representation where a static condensation is made more accurate by including FI modes near zero frequency, the quasi-static HCB representation allows the frequency range of a dynamic condensation to be expanded by adding FI modes that are near the centering frequency. A practical point to consider is that the matrix inversion in Eq. (6.3) will be poorly conditioned if ω_c is too close to any natural frequencies of the FI eigenvalue problem. This is easily avoided however because the FI eigenvalue solutions in the vicinity of the centering frequency must be obtained anyway.

Compared to the original HCB approach, the quasi-static HCB method typically requires many more FI modes to accurately span a similar sized frequency range. It is possible however

to improve the accuracy in a frequency range by including multiple quasi-static mode sets [77]. Three quasi-static mode sets equally spaced in eigenvalue space show give very good accuracy. For a frequency range spanning from ω_a to ω_b , a good selection for the centering frequencies is:

$$\omega_{c1} = \omega_A, \quad \omega_{c2} = \sqrt{\frac{1}{2}(\omega_A^2 + \omega_B^2)}, \quad \omega_{c3} = \omega_B. \quad (6.4)$$

The transformation matrix to obtain the quasi-static HCB model is then formed simply by collecting all of the constraint mode sets together with the FI modes that fall in the frequency window of interest,

$$\underbrace{\begin{Bmatrix} \mathbf{q}_I \\ \mathbf{q}_A \end{Bmatrix}}_{\mathbf{q}} = \underbrace{\begin{bmatrix} \Phi_I & \Psi_{QS1} & \Psi_{QS2} & \Psi_{QS3} \\ \mathbf{0} & \mathbf{I} & \mathbf{I} & \mathbf{I} \end{bmatrix}}_{\mathbf{B}_{QS}} \underbrace{\begin{Bmatrix} \eta_I \\ \mathbf{q}_{A1} \\ \mathbf{q}_{A2} \\ \mathbf{q}_{A3} \end{Bmatrix}}_{\mathbf{g}}. \quad (6.5)$$

There is no guarantee that the three sets of constraint modes are linearly independent, so it may be beneficial to discard any modes that are too similar to other modes, or to perform an orthogonalization. In the present work however, the mode sets are simply left as is without noticeable consequence. The transformation is applied to the full free mass and stiffness matrices to obtain the free quasi-static HCB matrices. Bloch BCs can be applied in order to obtain the band-structure solution inside the frequency window. Since there are now three separate sets of interface DOFs, Bloch BCs must be applied to each interface set.

The steps used to compute a quasi-static BMS model and obtain a segment of the band-structure centered about frequency ω_S are then as follows:

- (1) Compute the k eigenpairs of the FI eigenvalue problem with eigenvalue nearest to ω_S^2 , ordered as $\lambda_1^{\text{FI}} \leq \dots \leq \lambda_k^{\text{FI}}$.
- (2) Define the quasi static centering frequencies for the window based on the FI eigenvalues:
$$\left\{ \omega_{c1}^2 \quad \omega_{c2}^2 \quad \omega_{c3}^2 \right\} = \lambda_1^{\text{FI}} + \left\{ 0.05 \quad 0.5 \quad 0.95 \right\} (\lambda_k^{\text{FI}} - \lambda_1^{\text{FI}}).$$

- (3) Combine the FI eigenvalues and quasi-static constraint mode sets into a reduced-order basis and obtain the quasi-static BMS mass and stiffness matrices.
- (4) At every \mathbf{k} point, apply Bloch BCs to the reduced order model and compute k eigenpairs.

With this quasi-static BMS framework for computing a reduced-order band structure in an arbitrary frequency window, the spectral slicing algorithm presented earlier can be applied with two small changes to obtain the full-spectrum band-structure window by window. First, the lowest and highest spectral slices are centered at the maximum and minimum frequencies obtained from an iterative solution of the full model at the Γ point. Second, for the overlap between band-structure windows, the lowest maximum frequency (across all \mathbf{k} points) for a window is compared to the highest minimum of the next window.

6.4.1 Numerical Example: Silicon Nanopillared Membrane Full-Spectrum Band Structure

To demonstrate the spectrum-sliced BMS algorithm, it is used to find the full-spectrum band structure of the silicon nanopillared membrane shown in Fig. 6.2. This model contains 1920 silicon atoms and 5760 DOFs (after enforcing periodic BCs). The band-structure calculation is performed just over the $\Gamma - X$ portion of the BZ which is discretized into 17 \mathbf{k} points. In order to provide a reference, the full-spectrum band structure is first computed using the full model with a direct eigenvalue solver. Then, the windowed BMS approach is used to perform the same calculation. The resulting band-structure diagram is shown in Fig. 6.6. The high density of branches makes it difficult to resolve much detail in the band-structure diagram, so insets are included to show certain frequency ranges in more detail. From the insets, it appears that the BMS solution reproduces the full solution very well, however the BMS calculation does produce some spurious solutions (see circled points in top inset). The number of spurious solutions is quite small however compared to the number of true solutions. If the number of spurious solutions was very large, then the density of states curves shown on the right side of Fig. 6.6 would not match so closely. The density of

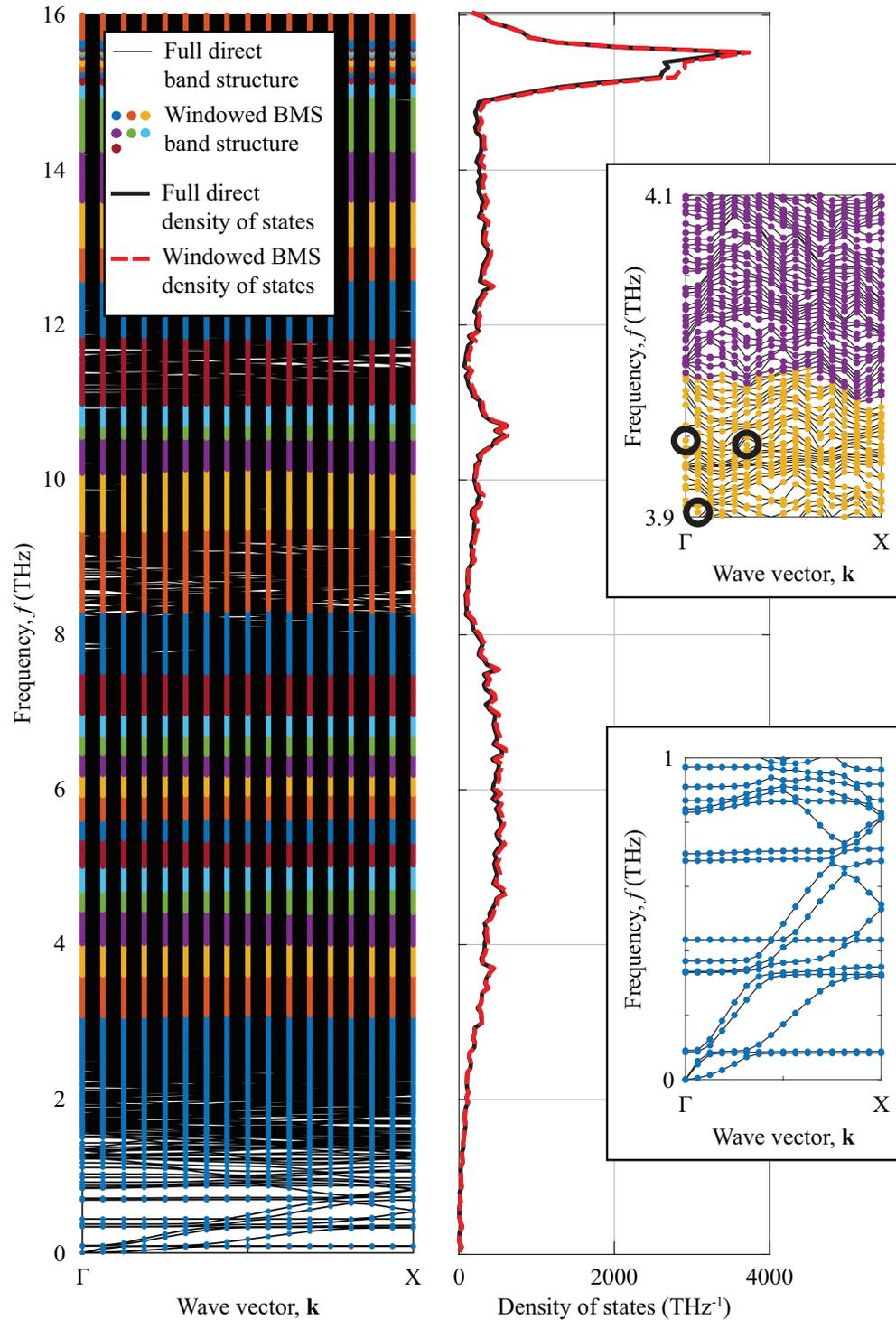


Figure 6.6: Full-spectrum band-structure diagram for silicon nanopillared membrane (left) and corresponding density of states, $DOS = n_{bin}/(n_k \times \Delta\omega_{bin})$. Results for both full-direct calculation and spectrum-sliced BMS calculation are shown. The BMS band structure solutions are shown as colored dots with different colors illustrating different frequency windows. The insets show more detail for select frequency ranges. Circled points indicate spurious BMS solutions

states is computed by splitting the frequency range into 250 bins, counting the total number of band-structure solutions in every bin, and dividing by the bin width multiplied by the number of \mathbf{k} points.

The computation time required for the full direct solution is 2,585 seconds. The windowed BMS calculation on the other hand takes 62,828 seconds. Thus no reduction in computation time has been achieved. The reason for this is that the number of boundary DOFs in the BMS model is very large because three sets of constraint modes are required. The windowed BMS calculation would clearly benefit from an interface reduction, but all attempts thus far have failed to obtain accurate results when local interface reduction is applied to the boundary. It is possible that interface reduction fails because the quasi-free representation used to model the silicon nanopillar does not yield true boundary modes. Thus, one possibility is to use a system-level interface reduction rather than local interface reduction. Although the interface reduction would have to be computed at every \mathbf{k} point, it may be possible to do so independently for each of the three quasi-static constraint mode sets. This would divide the single large eigenvalue problem into three smaller eigenvalue problems that can be solved much more quickly.

Another possibility for speeding up the windowed BMS calculation is to extend residual-mode enhancement to the quasi-static BMS model. If this is possible, it would allow for larger (and hence fewer) windows to be considered in the spectrum-sliced BMS calculation.

6.5 Summary

This chapter presented a framework for improving the computational efficiency of full-spectrum phononic band-structure calculations for atomic crystals. First, the modeling approach itself was improved by utilizing a building-block approach to perform the system assembly. Also, a quasi-free unit-cell representation was developed to allow for just a single model to be formed (rather than one for every \mathbf{k} point). Next, the computational cost of the full-spectrum eigenvalue solution was addressed. The approach proposed in this chapter was to employ spectrum slicing so that the $\mathcal{O}(n^3)$ direct eigenvalue solution can be replaced with a series of $\mathcal{O}(n)$ iterative eigenvalue solutions.

The number of spectrum slices is proportional to n as well, so the overall computational order of the spectrum slicing algorithm is $\mathcal{O}(n^2)$. This approach has a much higher constant multiplier so “break-even” does not occur until model sizes are quite large (about 20,000 DOFs for the 2D example considered, and higher when the number of non-zero terms in the system matrices increases). Next, an extension of BMS using quasi-static constraint mode sets was presented. This extension allows for reduced-order calculation of band structures in an arbitrary frequency range (i.e., not just the lowest frequency range). With the quasi-static representation it is possible to implement a reduced-order spectral slicing algorithm. Although this approach was shown to be relatively accurate in obtaining the band structure, the computational cost is considerable even with respect to the full-model direct solution. Thus, further work is necessary to improve the reduced-order spectrum slicing method. The extension of both interface reduction and residual mode enhancement to the quasi-static BMS representation would make the method much more effective.

Chapter 7

Conclusion

7.1 Summary

This thesis describes BMS, a novel unit-cell reduction technique inspired by CMS that dramatically reduces band-structure calculation times compared to full FE calculations. The examples shown in this thesis achieve speedups of one to two orders of magnitude for the $\omega(\mathbf{k})$ problem, but even better performance can be expected when the number of \mathbf{k} points is very large. For the $\mathbf{k}(\omega)$ problem, speedups of three orders of magnitude are demonstrated.

The BMS model is created by starting with a free, real-space model of the unit cell without any BCs applied. This free model allows for application of Bloch-periodic BCs, which incorporate phase terms corresponding to a PW with a given wave vector. The full-model band-structure calculation proceeds by stepping through a set of \mathbf{k} points in the BZ and at each \mathbf{k} point: (1) applying the corresponding Bloch BC transformation, and (2) solving the resulting eigenvalue problem for the band-structure frequencies (or energies). Rather than computing this full-model solution, a costly process due to the repeated eigenvalue solutions, one can first perform a BMS model-order reduction. This reduction begins with a real-space partitioning of the model into interior and boundary DOFs. The dynamics of the interior can be approximated very accurately using a small set of FI normal modes. The boundary partition is represented with a set of static constraint modes which preserve the boundary in its entirety so that the real-space application of Bloch BCs follows the exact same approach as with the full model. Thus steps (1) and (2) are exactly the same for the full and reduced-order models. In order for the BMS to obtain a very

accurate solution with the smallest possible model size, a residual-mode enhancement is applied in order to approximate the contribution of the interior modes that were truncated. Rather than computing the interior reduction all in one step, it can be computed using AMLS, a multi-level approach wherein the unit cell is iteratively split into smaller and smaller substructures until the substructure size is small enough to analyze very quickly. Each substructure is then reduced with its own set of FI and constraint modes and then the full model is assembled by recombining the substructures in each level until the reduced-representation of the full unit cell is recovered.

The reduction of the unit-cell interior is achieved via a secondary reduction of the unit cell (the primary reduction being the FE discretization). This provides significant benefit because the interior typically contains the majority of the model's DOFs. After reduction of the interior, the boundary terms typically represent the majority of BMS model's DOFs. Thus a tertiary reduction to reduce the size of the boundary size is applied. This boundary reduction is complicated by the fact that it must retain compatibility across interface sets that are to be coupled by Bloch BCs. A local interface reduction is used which first computes a global set of normal modes for the boundary, slices them into interface sets, and achieves compatibility by hybridizing the interface mode sets. The hybridization process simply concatenates all of the sliced modes that are to be coupled with their counterparts from opposite unit-cell faces and performs an orthogonalization to avoid ill conditioning. This allows for a reduction in the number of boundary DOFs that is somewhat less dramatic than the interior reduction, but that still provides significant improvement in computation time.

The extension of BMS to electronic-structure calculations is relatively straightforward. Once the Kohn-Sham equations have been discretized over the unit-cell domain with a real-space technique, the application of BMS is virtually identical to the elastic band-structure problem. The main differences compared to phononic band-structure calculations are that the BMS band-structure calculation is performed as part of a BZ integration that makes up one step in a self-consistent iteration, and that the error tolerances are quite a bit lower than those seen in elastic wave propagation. The current workhorse techniques rely on PW expansion which has many limitations. BMS brings FE

models one step closer to surpassing the performance of PW techniques.

This chapter introduced the fundamentals of electronic-structure calculation, including an overview of Schrödinger's equation and its simplification via DFT to the Kohn Sham equations. Then, the application of FE discretization to the Kohn Sham equations was described. Finally, the extension of BMS to electronic-structure calculations was demonstrated using a simple silicon unit cell (modeled using an empirical pseudopotential). The results showed speedups in computation time of over an order of magnitude compared to the full FE model, while retaining highly accurate solutions with less than 1×10^{-4} Ha/atom error in integrated band energy.

The BMS representation is very useful for computing a small number of the lowest frequency branches very quickly, but the accuracy of the solutions deteriorates as the frequency of the branches approaches the frequency of the highest interior/boundary modes used in the basis. In order to compute band-structure frequencies that are arbitrarily high, BMS must first be extended. This is possible by replacing the single set of static constraint modes with three sets of quasi-static constraint modes, each of which is computed about a centering frequency. These centering frequencies are chosen to give coverage of the frequency range of interest, and only those FI modes falling into this frequency range are kept in the reduced order basis. This modified BMS representation is referred to as windowed BMS because it allows for consideration of frequency windows. Combined with a spectrum slicing algorithm, it can be used to compute the full-spectrum band structure, which is particularly useful for atomic-scale phonon calculations. The performance of this technique is currently limited because it expands the number of interface DOFs by a factor of three, and does not subsequently admit a local interface reduction. This is a challenge that must be overcome before windowed BMS can make an impact for full-spectrum band structure calculations.

7.2 Future Work

There are several ways in which the applicability and performance of BMS can be expanded. First, for macroscale phononic crystals, the presence of damping can have a marked effect on the band structure. Damped band-structure calculation is more difficult because it requires a

state-space solution approach. Furthermore, the difficulty of damped band-structure calculation is compounded by the fact that the eigenvalues are complex and it may not always be easy for iterative solvers to obtain the desired branches first. In some cases this requires a direct eigenvalue solution which significantly limits the size of models that can be considered. The extension of CMS techniques to include damping is relatively straightforward, so the extension of BMS techniques to include damping should be as well.

Typically, CMS representations are not used to speed up the calculation of characteristic solutions (i.e., eigensolutions), because the up-front cost is usually too large. Thus, one of the main applications of CMS is to provide reduced-order models for time integration. The reduced-order models have a twofold benefit for these problems. First, the smaller model size reduces the computational cost at every time step. Second, truncating the high frequency modes provides a stabilizing effect that allows explicit methods to take larger time steps. Similar benefits are expected if BMS is applied to reduced-order direct numerical simulation of the unit-cell [78]. This is an application that has not been tested, but that should require no modification to the BMS model-order reduction itself.

The residual-mode enhancement is capable of markedly lowering the error introduced by the BMS model. The end effect is that much fewer interior modes need to be kept to achieve good accuracy. A similar correction applied to the boundary modes could allow the size of the reduced-order model to be decreased as well, leading to improved performance. Residual-mode enhancement is quite simple to extend to system-level boundary modes (after interfaces are coupled), but does not work for the local interface description. Thus, further investigation into the extension of residual-enhancement techniques for application to local interface reduction is necessary.

As mentioned in the previous section, the windowed BMS reduction does not provide much, if any, computational benefit for full-spectrum problems because the number of interface DOFs is very large and in the current description does not allow for local interface reduction or residual enhancement. Extending both of these techniques to the quasi-static BMS model could allow it to provide significant computational benefit. An alternate avenue for obtaining a reduced-order

model for spectrum slicing is to extend RBME to higher frequency windows.

Finally, the FE models used as a basis for BMS model reduction are very simple based on Lagrangian shape functions. A large benefit of the BMS representation is that it can be applied to any real-space model that allows a decomposition into interior and boundary DOFs. Thus, commercial FE packages can be employed to produce the base model and then reduced with BMS. In order to compare with techniques based on PW expansion, FE-based BMS should be tested with the highest performing base FE model possible.

Bibliography

- [1] Felix Bloch, "Über die Quantenmechanik der Elektronen in Kristallgittern," Z. Physik, 1928.
- [2] M. I. Hussein, M. J. Leamy, and M. Ruzzene, "Dynamics of Phononic Materials and Structures: Historical Origins, Recent Progress, and Future Outlook," Applied Mechanics Reviews, vol. 66, pp. 040802–040802, May 2014.
- [3] C. Kittel, Introduction to solid state physics. Hoboken, NJ: Wiley, 8th ed ed., 2005.
- [4] W. H. Bragg and W. L. Bragg, "The Reflection of X-rays by Crystals," Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences, vol. 88, pp. 428–438, July 1913.
- [5] Z. Liu, "Locally Resonant Sonic Materials," Science, vol. 289, pp. 1734–1736, Sept. 2000.
- [6] C. Felippa, "Stress Strain Material Laws." 2015.
- [7] J. Tersoff, "New empirical approach for the structure and energy of covalent systems," Physical Review B, vol. 37, pp. 6991–7000, Apr. 1988.
- [8] F. H. Stillinger and T. A. Weber, "Computer simulation of local order in condensed phases of silicon," Physical Review B, vol. 31, pp. 5262–5271, Apr. 1985.
- [9] D. S. Sholl and J. A. Steckel, Density functional theory: a practical introduction. Hoboken, N.J: Wiley, 2009.
- [10] W. Axemann and P. Kuchment, "An Efficient Finite Element Method for Computing Spectra of Photonic and Acoustic Band-Gap Materials: I. Scalar Case," Journal of Computational Physics, vol. 150, pp. 468–481, 1999.
- [11] R. C. Raffenetti, "A simultaneous coordinate relaxation algorithm for large, sparse matrix eigenvalue problems," Journal of Computational Physics, vol. 32, no. 3, pp. 403–419, 1979.
- [12] Y. Saad, "Iterative Methods for Sparse Linear Systems Second Edition," Society for Industrial and Applied Mathematics, 2003.
- [13] D. C. Dobson, J. Gopalakrishnan, and J. E. Pasciak, "An Efficient Method for Band Structure Calculations in 3d Photonic Crystals," Journal of Computational Physics, vol. 161, pp. 668–679, July 2000.

- [14] R. L. Chern, C. C. Chang, C. C. Chang, and R. R. Hwang, "Large full band gaps for photonic crystals in two dimensions computed by an inverse method with multigrid acceleration," Physical Review E, vol. 68, Aug. 2003.
- [15] R. J. Guyan, "Reduction of Stiffness and Mass Matrices," AIAA Journal, vol. 3, no. 2, p. 380, 1965.
- [16] R. Anderson, B. Irons, and O. Zienkiewicz, "Vibration and Stability of Plates Using Finite Elements," International Journal of Solids and Structures, vol. 4, pp. 1031–1055, 1968.
- [17] E. L. Shirley, "Optimal basis sets for detailed Brillouin-zone integrations," Physical Review B, vol. 54, no. 23, p. 16464, 1996.
- [18] M. I. Hussein, "Reduced Bloch mode expansion for periodic media band structure calculations," Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences, vol. 465, pp. 2825–2848, Sept. 2009.
- [19] C. Scheiber, A. Schultschik, O. Biro, and R. Dyczij-Edlinger, "A Model Order Reduction Method for Efficient Band Structure Calculations of Photonic Crystals," IEEE Transactions on Magnetics, vol. 47, pp. 1534–1537, May 2011.
- [20] D. Krattiger and M. I. Hussein, "Bloch mode synthesis: Ultrafast methodology for elastic band-structure calculations," Physical Review E, vol. 90, Dec. 2014.
- [21] C. W. Zhou, J. P. Lainé, M. N. Ichchou, and A. M. Zine, "Multi-scale modelling for two-dimensional periodic structures using a combined mode/wave based approach," Computers & Structures, vol. 154, pp. 145–162, July 2015.
- [22] C. W. Zhou, J. P. Lainé, M. N. Ichchou, and A. M. Zine, "Wave Finite Element Method Based on Reduced Model for One-Dimensional Periodic Structures," International Journal of Applied Mechanics, vol. 07, p. 1550018, Apr. 2015.
- [23] O. R. Bilal and M. I. Hussein, "Ultrawide phononic band gap for combined in-plane and out-of-plane waves," Physical Review E, vol. 84, Dec. 2011.
- [24] O. Sigmund and J. S. Jensen, "Systematic design of phononic band-gap materials and structures by topology optimization," Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences, vol. 361, pp. 1001–1019, May 2003.
- [25] L. Brillouin, Wave Propagation in Periodic Structures: Electric Filters and Crystal Lattices. Dover Publications, 1953.
- [26] Robert D. Cook, David S. Malkus, Michael E. Plesha, and Robert J. Witt, Concepts and Applications of Finite Element Analysis. Wiley, 2001.
- [27] T. J. R. Hughes, The Finite Element Method: Linear Static and Dynamic Finite Element Analysis. Dover Publications, Aug. 2000.
- [28] E. Manconi, Modeling Wave Propagation in Two-Dimensional Structures Using a Wave/Finite Element Technique. PhD thesis, University of Parma, 2008.
- [29] B. R. Mace and E. Manconi, "Modelling wave propagation in two-dimensional structures using finite element analysis," Journal of Sound and Vibration, vol. 318, pp. 884–902, Dec. 2008.

- [30] F. Tisseur, “Backward error and condition of polynomial eigenvalue problems,” Linear Algebra and its Applications, vol. 309, pp. 339–361, Apr. 2000.
- [31] F. Tisseur and K. Meerbergen, “The quadratic eigenvalue problem,” SIAM review, vol. 43, no. 2, pp. 235–286, 2001.
- [32] M. Collet, M. Ouisse, M. Ruzzene, and M. N. Ichchou, “Floquet-Bloch decomposition for the computation of dispersion of two-dimensional periodic, damped mechanical systems,” International Journal of Solids and Structures, vol. 48, pp. 2837–2848, Oct. 2011.
- [33] D. Krattiger, R. Khajehtourian, C. Bacquet, and M. Hussein, “Anisotropic dissipation in lattice metamaterials,” AIP Advances, vol. 6, p. 121802, Dec. 2016.
- [34] M. J. Frazier and M. I. Hussein, “Generalized Bloch’s theorem for viscous metamaterials: Dispersion and effective properties based on frequencies and wavenumbers that are simultaneously complex,” Comptes Rendus Physique, vol. 17, pp. 565–577, May 2016. arXiv: 1601.00683.
- [35] R. R. Craig and Z. Ni, “Component mode synthesis for model order reduction of nonclassical-lydamped systems,” Journal of Guidance, Control, and Dynamics, vol. 12, no. 4, pp. 577–584, 1989.
- [36] B. de Kraker and V. F. Werktuigkunde, “Generalization of the craig-bampton cms procedure for general damping,” Eindhoven, Februari, 1993.
- [37] Y. Chen, T. A. Davis, W. W. Hager, and S. Rajamanickam, “Algorithm 887: CHOLMOD, Supernodal Sparse Cholesky Factorization and Update/Downdate,” ACM Trans. Math. Softw., vol. 35, pp. 22:1–22:14, Oct. 2008.
- [38] W. C. Hurty, “Dynamic analysis of structural systems using component modes,” AIAA Journal, vol. 3, pp. 678–685, Apr. 1965.
- [39] M. C. C. Bampton and R. R. Craig, “Coupling of substructures for dynamic analyses,” AIAA Journal, vol. 6, no. 7, pp. 1313–1319, 1968.
- [40] R. R. Craig and C. J. Chang, “On the use of attachment modes in substructure coupling for dynamic analysis,” (San Diego, CA), American Institute of Aeronautics and Astronautics, Mar. 1977.
- [41] R. M. Hintz, “Analytical Methods in Component Modal Synthesis,” AIAA Journal, vol. 13, no. 8, pp. 1007–1016, 1975.
- [42] S. Rubin, “Improved Component-Mode Representation for Structural Dynamic Analysis,” AIAA Journal, vol. 13, no. 8, pp. 995–1006, 1975.
- [43] R. Bladh, M. P. Castanier, and C. Pierre, “Component-Mode-Based Reduced Order Modeling Techniques for Mistuned Bladed Disks—Part I: Theoretical Models,” Journal of Engineering for Gas Turbines and Power, vol. 123, pp. 89–99, Apr. 2000.
- [44] A. Kropp and D. Heiserer, “Efficient Broadband Vibro-Acoustic Analysis of Passenger Car Bodies Using an FE-Based Component Mode Synthesis Approach,” Journal of Computational Acoustics, vol. 11, pp. 139–157, June 2003.

- [45] L. Suarez and M. Singh, “Improved Fixed Interface Method for Modal Synthesis,” AIAA Journal, vol. 30, no. 12, pp. 2952–2958, 1992.
- [46] J.-G. Kim and P.-S. Lee, “An enhanced Craig-Bampton method,” International Journal for Numerical Methods in Engineering, vol. 103, pp. 79–93, July 2015.
- [47] S.-H. Boo, J.-G. Kim, and P.-S. Lee, “A simplified error estimator for the CB method and its application to error control,” Computers & Structures, vol. 164, pp. 53–62, Feb. 2016.
- [48] J.-G. Kim, S.-H. Boo, and P.-S. Lee, “An enhanced AMLS method and its performance,” Computer Methods in Applied Mechanics and Engineering, vol. 287, pp. 90–111, Apr. 2015.
- [49] M. F. Kaplan, Implementation of Automated Multilevel Substructuring for Frequency Response Analysis of Structures. PhD thesis, The University of Texas at Austin, Dec. 2001.
- [50] J. K. Bennighof and R. B. Lehoucq, “An Automated Multilevel Substructuring Method for Eigenspace Computation in Linear Elastodynamics,” SIAM Journal on Scientific Computing, vol. 25, pp. 2084–2106, Jan. 2004.
- [51] A. Pothen, H. D. Simon, and K.-P. Liou, “Partitioning sparse matrices with eigenvectors of graphs,” SIAM journal on matrix analysis and applications, vol. 11, no. 3, pp. 430–452, 1990.
- [52] M. P. Castanier, Y.-C. Tan, and C. Pierre, “Characteristic Constraint Modes for Component Mode Synthesis,” AIAA Journal, vol. 39, pp. 1182–1187, June 2001.
- [53] D. Krattiger, L. Wu, M. Zacharczuk, M. Buck, R. J. Kuether, M. S. Allen, P. Tiso, and M. R. Brake, “Interface Reduction for Hurty/Craig-Bampton Substructured Models: Review and Improvements,” Mechanical Systems and Signal Processing (Under Review).
- [54] S.-K. Hong, B. I. Epureanu, and M. P. Castanier, “Next-generation parametric reduced-order models,” Mechanical Systems and Signal Processing, vol. 37, pp. 403–421, May 2013.
- [55] E. Schrödinger, “An Undulatory Theory of the Mechanics of Atoms and Molecules,” Physical Review, vol. 28, pp. 1049–1070, Dec. 1926.
- [56] P. a. M. Dirac, “Quantum Mechanics of Many-Electron Systems,” Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences, vol. 123, pp. 714–733, Apr. 1929.
- [57] S. Sanvito, C. Oses, J. Xue, A. Tiwari, M. Zic, T. Archer, P. Tozman, M. Venkatesan, M. Coey, and S. Curtarolo, “Accelerated discovery of new magnets in the Heusler alloy family,” Science Advances, vol. 3, p. e1602241, Apr. 2017.
- [58] P. Hohenberg and W. Kohn, “Inhomogeneous Electron Gas,” Physical Review, vol. 136, pp. B864–B871, Nov. 1964.
- [59] W. Kohn and L. J. Sham, “Self-Consistent Equations Including Exchange and Correlation Effects,” Physical Review, vol. 140, pp. A1133–A1138, Nov. 1965.
- [60] B. L. Davis and M. I. Hussein, “Nanophononic Metamaterial: Thermal Conductivity Reduction by Local Resonance,” Physical Review Letters, vol. 112, p. 055505, Feb. 2014.

- [61] J. Lowney, A. Kahn, J. Blue, and C. Wilson, “Disappearance of impurity levels in silicon and germanium due to screening,” Journal of Applied Physics, vol. 52, pp. 4075–4080, June 1981.
- [62] S. R. White, J. W. Wilkins, and M. P. Teter, “Finite-element method for electronic structure,” Physical Review B, vol. 39, pp. 5819–5833, Mar. 1989.
- [63] R. Stowasser and R. Hoffmann, “What Do the Kohn-Sham Orbitals and Eigenvalues Mean?,” Journal of the American Chemical Society, vol. 121, pp. 3414–3420, Apr. 1999.
- [64] H. J. Monkhorst and J. D. Pack, “Special points for Brillouin-zone integrations,” Physical Review B, vol. 13, pp. 5188–5192, June 1976.
- [65] R. M. Martin, Electronic Structure: Basic Theory and Practical Methods. Cambridge University Press, Apr. 2004. Google-Books-ID: dmRTFLpSGNsC.
- [66] N. W. Ashcroft and N. D. Mermin, Solid State Physics. Holt, Rinehart and Winston, 1976. Google-Books-ID: 1C9HAQAIAAJ.
- [67] J. E. Pask, “Finite-Element Methods in Electronic-Structure Theory,” Computer Physics Communications, vol. 135, no. 1-34, 2001.
- [68] J. E. Pask and P. A. Sterne, “Finite element methods in ab initio electronic structure calculations,” Modelling and Simulation in Materials Science and Engineering, vol. 13, no. 3, p. R71, 2005.
- [69] M. L. Cohen and T. K. Bergstresser, “Band Structures and Pseudopotential Form Factors for Fourteen Semiconductors of the Diamond and Zinc-blende Structures,” Physical Review, vol. 141, pp. 789–796, Jan. 1966.
- [70] H. Honarvar and M. I. Hussein, “Spectral energy analysis of locally resonant nanophononic metamaterials by molecular simulations,” Physical Review B, vol. 93, p. 081412, Feb. 2016.
- [71] M. Dove, “Introduction to the theory of lattice dynamics,” École thématique de la Société Française de la Neutronique, vol. 12, pp. 123–159, 2011.
- [72] J. Tersoff, “Empirical interatomic potential for silicon with improved elastic properties,” Physical Review B, vol. 38, pp. 9902–9905, Nov. 1988.
- [73] G. Schofield, J. R. Chelikowsky, and Y. Saad, “A spectrum slicing method for the Kohn–Sham problem,” Computer Physics Communications, vol. 183, pp. 497–505, Mar. 2012.
- [74] C. Campos and J. E. Roman, “Strategies for spectrum slicing based on restarted Lanczos methods,” Numerical Algorithms, vol. 60, pp. 279–295, June 2012.
- [75] H. M. Aktulga, L. Lin, C. Haine, E. G. Ng, and C. Yang, “Parallel eigenvalue calculation based on multiple shift–invert Lanczos and contour integral based spectral projection method,” Parallel Computing, vol. 40, pp. 195–212, July 2014.
- [76] W.-H. Shyu, Z.-D. Ma, and G. M. Hulbert, “A New Component Mode Synthesis Method: Quasi-Static Mode Compensation,” Finite Elements in Analysis and Design, vol. 24, pp. 271–281, 1997.

- [77] W.-H. Shyu, J. Gu, G. M. Hulbert, and Z.-D. Ma, “On the use of multiple quasi-static mode compensation sets for component mode synthesis of complex structures,” Finite Elements in Analysis and Design, vol. 35, pp. 119–140, 2000.
- [78] A. Cebrecos, D. Krattiger, K. Maute, V. Sánchez-Morcillo, K. Park, I. Oh, and M. Hussein, “Fluidic metamaterial: An elastic medium with a time-changing band structure,” in Proceedings of Phononics 2015, (Paris, France), June 2015.

Appendix A. Acronyms

AMLS automated multi-level substructuring. 53, 54, 61, 63, 124

AMLS+ residual-enhanced AMLS. 53, 61, 63

BC boundary condition. 7, 8, 17–19, 21, 22, 25, 26, 28, 30, 32, 34, 35, 41, 42, 45, 46, 48, 49, 54, 65, 66, 69–71, 79, 83, 108, 109, 118, 119, 123, 124

BMS Bloch mode synthesis. 7–9, 18, 34, 37, 39, 40, 42, 46, 49, 56, 57, 61, 65, 66, 69, 73, 74, 76, 79, 81, 83, 87, 89, 90, 97–103, 105, 107, 116, 118, 119, 121–126

BZ Brillouin zone. 7, 12, 15, 18, 25, 41, 45, 66, 70, 81, 91–93, 99, 102, 103, 106, 116, 119, 123

CC characteristic constraint. 7, 65

CMS component mode synthesis. 7, 8, 37, 46, 49, 63, 65, 123, 125

DFT density functional theory. 85, 87, 88, 90, 103, 107

DOF degree of freedom. 5–7, 11, 17, 18, 21, 22, 24, 32, 40, 41, 46, 48, 49, 53, 54, 57, 58, 61, 63, 65, 70, 71, 73, 76, 77, 79–81, 83, 99, 100, 102, 118, 119, 121, 123–126

EC exact-compatibility. 66, 75, 83

FD finite difference. 4, 9, 39

FE finite element. x, 4, 9, 18–20, 23, 26, 30, 34, 39, 54, 56, 63, 64, 79, 86, 87, 95–101, 105, 106, 110, 123, 124, 126

FI fixed interface. 46, 48, 49, 61, 63, 79, 83, 101, 102, 117, 118

FRF frequency response function. 44, 45

H-CC hybrid-level CC. 66

HCB Hurty/Craig-Bampton. 46, 48–51, 53, 59, 61, 63–65, 70, 116–118

HCB+ residual-enhanced HCB. 53, 63

IBZ irreducible Brillouin zone. 15, 61, 92, 111

L-CC local-level CC. 66, 70, 75, 83, 84

MP Monkhorst-Pack. 91–95, 99, 100, 102

ODE ordinary differential equation. 4, 45

PW plane-wave. 4, 9, 86, 87, 95, 96

RBME reduced Bloch mode expansion. 7, 45, 126

S-CC system-level CC. 65, 66, 69, 84

SICOR simultaneous coordinate over-relaxation. 5, 6

SVD singular value decomposition. 71

UWC uncoupled weak-compatibility. 66

VN virtual node. 66

WC weak-compatibility. 66