Efficient computation of equifrequency surfaces and density of states in photonic crystals using Dirichlet-to-Neumann maps

Victor Liu* and Shanhui Fan

Department of Electrical Engineering, Stanford University, Stanford, California USA
**Corresponding author: vkl@stanford.edu*

Received February 15, 2011; revised May 14, 2011; accepted June 1, 2011;
posted June 9, 2011 (Doc. ID 142730); published July 6, 2011

We present an efficient method for computing the equifrequency surfaces (EFSs) and density of states of a photonic crystal. The method is based on repeatedly solving a small nonlinear eigenvalue problem formulated using the Dirichlet-to-Neumann map of the unit cell. A simple contouring algorithm is presented for sampling EFSs as well as computing group velocity vectors. We compare our method with several published results to demonstrate its efficiency and accuracy. © 2011 Optical Society of America

OCIS codes: 000.4430, 000.3860, 160.5298.

1. INTRODUCTION

The concepts of density of states (DOS) and the EFS have proven essential in the study of electromagnetic wave propagation through periodic structures. The EFSs of a photonic crystal are derived from its band structure, usually written as $\omega(\mathbf{k})$ describing the frequency ω of a mode that satisfies the wave equation in the crystal for a particular Bloch wave vector $\mathbf{k}$. At a particular frequency ω_0, the EFSs are the set of $\mathbf{k}$ such that $\omega_0 = \omega(\mathbf{k})$. The EFSs of the band structure are of great importance in understanding negative refraction behavior [1,2]; in designing superprisms, splitters, and lenses [3–5]; in modeling graded index structures; and recently, in designing transformation optics [6,7]. In all these applications, moreover, it is important to calculate the local group velocity vector normal to the EFS, as the group velocity vector governs light propagation in the periodic structure.

The computation of the EFS is also closely related to the computation of DOS. The DOS is usually defined as

$$N(\omega) = \sum_m \frac{1}{A_{\mathrm{BZ}}} \int_{\mathrm{BZ}} \delta(\omega - \omega_m(\mathbf{k}))\mathrm{d}^2\mathbf{k}, \tag{1}$$

where the integral is taken over the m-th band and A_{BZ} is the area of the Brillouin zone (BZ). Equation (1) can also be written as

$$N(\omega) = \sum_m \frac{1}{A_{\mathrm{BZ}}} \int_{\mathrm{EFS}_m} \| \frac{d\mathbf{k}}{d\omega} \| \mathrm{d}s, \tag{2}$$

where the integral is taken along the m-th EFS at frequency ω and $v_g^{-1} = d\mathbf{k}/d\omega$ is the inverse group velocity. The DOS was first used in understanding the modification of spontaneous emission in photonic crystals [8]. The design of photonic crystals and metamaterials for the modification of the DOS has been extensively studied [9,10]. More recently, it was shown that the DOS plays an important role in light trapping for solar cells [11] and in mode confinement in photonic crystal structures [12].

Existing methods for computing EFSs first compute the full band surface over the BZ, and then extract the EFSs as contours or surfaces from the bands [13–15]. These methods are slow as they require a fine sampling of the BZ in order to accurately resolve the EFSs. Moreover, if the EFSs at a single frequency or a narrow range of frequencies is desired, then computing the full band structure represents a significant waste of effort since only those bands near the frequency of interest are needed.

There are related problems in existing methods for computing the DOS of a photonic crystal. The definition in Eq. (1) suggests the typical method by which the DOS is computed: using the full band structure and binning by frequency to approximate the integral. The frequency binning method can be improved if the group velocities are also available. The DOS can also be obtained from the local density of states by an integral over the real space unit cell, or from the angular density of states by an integral over the BZ [16]. These integration methods are computationally intensive since they require repeated computation of Green's functions in the process of performing the integration. A survey of DOS integration techniques was presented in [17].

In this paper, based upon the recently proposed Dirichlet-to-Neumann (DtN) method for photonic crystal modeling, we present an efficient and direct method of computing EFSs, which simultaneously allows efficient semianalytic computation of the group velocity vector on an EFS. This method, in turn, enables the efficient computation of the DOS at a single frequency without having to compute anything at any other frequency. We will focus on photonic crystals with circular inclusions (cylindrical rods or holes) since such geometries are the most practically interesting. The DtN method itself can be used for geometries with arbitrarily shaped inclusions. Our methods are also applicable in those general cases since all we require as input is the DtN map. For circular inclusions, the DtN method is certainly one of most efficient algorithms,

with computational speed and accuracy likely comparable to the more widely used multiple multipole method [18] since both methods are based on cylindrical wave expansions. Our aim here is to extend the capability of DtN methods since it has already proven to be a highly efficient method for photonic crystal simulation.

The paper is organized as follows: We first provide an overview of the DtN method as it applies to computing band structures and EFSs in Section 2. In Section 3, we describe a basic unit of the algorithm we call the "oracle" that, given a line segment, computes all points of intersection between the segment and EFSs. We use the oracle as a building block in the sampling algorithm described in Section 4. In Section 5, we build on the capabilities of the preceding sections to develop integration techniques for computing the DOS. In Section 6, we present numerical examples and compare our methods against existing results. Throughout this paper for the purposes of illustration, we will work in 2D so that we work with only scalar fields (either the out-of-plane electric or magnetic field, depending on polarization). For the purposes of numerical simulation, we consider only photonic crystals with either a square lattice of circular dielectric cylinders in air, or a triangular lattice of circular air holes in a dielectric background. This method can also be generalized to photonic crystals with arbitrary unit cells and to 3D structures.

2. OVERVIEW OF DtN METHOD FOR EFS COMPUTATION

The DtN map has been used for computing band structures of 2D photonic crystals [19], analyzing waveguides [20], and analyzing photonic devices [21]. Such a method has also been shown to provide a basis for efficient optimization in systematic design of photonic structures [22,23]. We briefly summarize the essentials of the DtN method here. The DtN map Λ is a linear operator that maps the value of a function on the boundary of a region to its normal derivative over the boundary. We may write this as

$$\Lambda f|_\Omega = \left.\frac{\partial f}{\partial \hat{n}}\right|_\Omega, \tag{3}$$

where f is a function defined over a simply connected region of space Ω, and the partial derivative denotes the outward normal derivative of the function on the boundary. In practice, the function values are discretized into a finite number of points along the boundary of the region and Λ becomes a matrix operator.

In our case, when we model a photonic crystal (Fig. 1), we choose the region to be the unit cell containing a cylindrical rod or hole so the DtN matrix approximation may be computed analytically using a cylindrical wave decomposition of the fields within the unit cell. First, we compute the Dirichlet map Λ_D, mapping from coefficients of the cylindrical wave expansion to field values at the boundary of the cell. The (i,j)-th element of the matrix is given explicitly by

$$(\Lambda_D)_{i,j} = \phi_j(\mathbf{r}_i), \tag{4}$$

where $\mathbf{r}_i$ is the position of the i-th field discretization point on the boundary of the unit cell, and ϕ_j is the j-th cylindrical wave basis function. Similarly, we also compute the Neumann map Λ_N, mapping from cylindrical wave coefficients to the normal derivative of field values over the boundary of the cell. Clearly, in order for these maps to be one-to-one, we must use as many cylindrical waves as there are discretization points. The DtN map is then $\Lambda = \Lambda_N \Lambda_D^{-1}$. Details for computing the DtN map for unit cells containing cylindrical features are given in [19], and for arbitrary unit cells in [24].

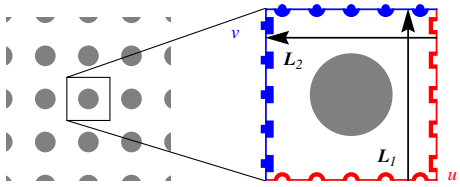

Fig. 1. (Color online) Illustration of a unit cell in a square lattice of dielectric rods (gray). The unit cell is discretized with five points per edge. The points on red edges correspond to field values in the u vector, while the blue points correspond to v. The shapes of the points indicate how they should be matched for the Bloch periodic boundary conditions. The lattice vectors corresponding to each pair of matching edges are indicated by the labeled arrows.

Once the DtN map at a given frequency ω is known, an eigenvalue equation can be formed for the Bloch wave vector $\mathbf{k}$ by applying Bloch periodic boundary conditions to the unit cell. Starting with Eq. (3) and assuming the boundary discretization points are chosen to match on opposite edges of the unit cell, we may partition Eq. (3) as

$$\begin{bmatrix} \Lambda_{uu} & \Lambda_{uv} \\ \Lambda_{vu} & \Lambda_{vv} \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \partial_n u \\ \partial_n v \end{bmatrix}, \tag{5}$$

where the vector v contains field values displaced from those in u by a lattice vector. We may think of each element in u being "matched" to its corresponding element in v after applying the Bloch periodic boundary conditions, as illustrated in Fig. 1.

For the system of Fig. 1, the Bloch periodic boundary conditions impose

$$v = \begin{bmatrix} e^{i\mathbf{k}\cdot\mathbf{L}_1} & & & & & \\ & \ddots & & & & \\ & & e^{i\mathbf{k}\cdot\mathbf{L}_1} & & & \\ & & & e^{i\mathbf{k}\cdot\mathbf{L}_2} & & \\ & & & & \ddots & \\ & & & & & e^{i\mathbf{k}\cdot\mathbf{L}_2} \end{bmatrix} u \tag{6}$$

$$= \mathrm{diag}(e^{i\mathbf{k}\cdot\mathbf{L}_1} I, e^{i\mathbf{k}\cdot\mathbf{L}_2} I) = Qu \tag{7}$$

$$\partial_n v = -Q\partial_n u, \tag{8}$$

where the operator Q is block diagonal, with each block being the identity operator of appropriate dimension multiplied by a Bloch phase factor using the corresponding lattice vector $\mathbf{L}_1$ or $\mathbf{L}_2$ (in the case of a hexagonal unit cell, Eqs. (5)–(8) would have three blocks instead of two). Combining Eqs. (5)–(8), we obtain

$$[\Lambda_{vu} + \Lambda_{vv}Q + Q\Lambda_{uu} + Q\Lambda_{uv}Q]u = 0, \tag{9}$$

which is a generally nonlinear eigenvalue equation in $\mathbf{k}$ and parameterized by frequency ω (Q is a function of $\mathbf{k}$ and Λ is a function of ω), which we can write as

$$A(\mathbf{k};\omega)u = 0. \tag{10}$$

We will suppress the explicit parametric dependence on ω in the remainder of the text, since all computations are performed at a fixed frequency. To determine the EFSs, we will in the next two sections discuss the procedure for determining all the $\mathbf{k}$ at a given frequency ω by repeatedly solving Eq. (10).

The eigensystem derived above is similar to that obtained in band structure calculations using scattering or transfer matrices [25,26]. However, the field representation within the unit cell is of much lower rank in the DtN method compared to these prior works, leading to much smaller eigensystems.

3. THE ORACLE

A. Motivation

In 2D, computing the EFSs corresponds to approximating the level set contours of Eq. (10) in the BZ. Any of the conventional methods for isosurface extraction can be used, such as marching cubes [27] or mesh refinement techniques [28]. All of these methods query the surface by way of an "oracle" that, given a line segment, determines all points on the segment that lie on the surface. Therefore, we present here an implementation of an oracle that can efficiently answer such queries for the purposes of computing EFSs and DOS.

We require the oracle to compute two things. First, on a specified simple 1D curve within the BZ (we focus here on line segments), the oracle must solve the eigensystem on the curve to compute all $\mathbf{k}$ points lying on an EFS. Second, at each point found on an EFS, the oracle must also compute the eigenderivatives to determine an in-plane tangent vector along the EFS as well as the (inverse) group velocity at that point. Below we describe an efficient and numerically stable implementation of these two steps. The algorithm presented here is limited to computing only real-valued solutions; evanescent solutions are beyond the scope of this work.

B. Eigensystem

We must solve the nonlinear eigensystem $A(\mathbf{k})u = 0$ where $A \in \mathbb{C}^{n\times n}$ where $\mathbf{k}$ is constrained to a line segment and n is half the total number of discretization points on the boundary of the unit cell. A line segment is parameterized by $\mathbf{k}(t) = \mathbf{k}_0 + t(\mathbf{k}_1 - \mathbf{k}_0) = \mathbf{k}_0 + t\Delta\mathbf{k}$ where $t \in [0,1]$ such that the eigensystem becomes $A(t)u = 0$. We are interested in finding those t for which $A(t)$ becomes singular, which can be done by finding the zeros of $\det A(t)$ or minimizing the smallest singular value $\sigma_{\min}(A(t))$. The advantage of working with the determinant is that its derivative is available; however, the smallest singular value is a more reliable indicator of singularity. To address these considerations we use a combination of a relaxed Newton's method and Müller's method, which solve for the roots of $\det A(t)$ by iteratively refining an initial guess [29]. We initialize the approximation to the eigenvalue t by using a fixed number of Newton iterations (for all examples presented here we use 5), then if convergence is not achieved, switch to Müller's method to minimize $\sigma_{\min}(A(t))$. The implementation of Müller's method with deflation closely follows a textbook description [29], so we will only describe the details of the Newton iteration since it contains nontrivial formulas concerning derivatives.

If the i-th estimate t_i is known, then a relaxed Newton's method gives the next estimate as

$$t_{i+1} = t_i - \alpha \frac{\det A(t_i)}{\frac{d}{dt}\det A(t_i)}, \tag{11}$$

where $0 < \alpha < 1$ is the relaxation factor. A value of $\alpha = 0.5$ was used; the exact value did not strongly affect convergence of the iteration unless α was very near 0 or 1. The derivative in Eq. (11) may be computed analytically:

$$\frac{d}{dt}\det A(t) = \mathrm{Tr}\left[\mathrm{adj}(A(t))\frac{d}{dt}A(t)\right] = \mathrm{Tr}[\mathrm{adj}(A)\Delta\mathbf{k}\cdot\nabla_{\mathbf{k}}A(\mathbf{k})], \tag{12}$$

where $\mathrm{adj}(A)$ is the adjugate of A. The adjugate of a matrix is the transposed matrix of its cofactors. For a nonsingular matrix $\mathrm{adj}(A) = \det(A)A^{-1}$ [30]; hence, with the use of Eq. (12), the Newton iteration in Eq. (11) simplifies to

$$t_{i+1} = t_i - \alpha[\mathrm{Tr}(A^{-1}\Delta\mathbf{k}\cdot\nabla_{\mathbf{k}}A(\mathbf{k}))]^{-1}. \tag{13}$$

In the course of the Newton iteration, we expect A to be nonsingular except at an eigenvalue. We therefore write Eq. (13) keeping in mind that A^{-1} must be computed with care later on.

The gradient with respect to $\mathbf{k}$ of A is

$$\begin{aligned}\nabla_{\mathbf{k}}A(\mathbf{k}) = A_{\mathbf{k}} &= \hat{k}_x\frac{\partial}{\partial k_x}A(\mathbf{k}) + \hat{k}_y\frac{\partial}{\partial k_y}A(\mathbf{k}) \\ &= \begin{bmatrix}\frac{\partial}{\partial k_x}A(\mathbf{k}) \\ \frac{\partial}{\partial k_y}A(\mathbf{k})\end{bmatrix} \equiv \begin{bmatrix}A_{k_x} \\ A_{k_y}\end{bmatrix}.\end{aligned} \tag{14}$$

Hence,

$$\Delta\mathbf{k}\cdot A_{\mathbf{k}} = \Delta k_x A_{k_x} + \Delta k_y A_{k_y}, \tag{15}$$

which is an $n\times n$ matrix. Substituting in the form of A from Eq. (9),

$$\Delta\mathbf{k}\cdot A_{\mathbf{k}} = \Lambda_{vv}RQ + QR\Lambda_{uu} + QR\Lambda_{uv}Q + Q\Lambda_{uv}RQ, \tag{16}$$

with $R = \mathrm{diag}(i\Delta\mathbf{k}\cdot\mathbf{L}_1 I, i\mathbf{k}\cdot\mathbf{L}_2 I)$.

In order to compute multiple roots using a Newton iteration, it is typically necessary to deflate the method. This may be accomplished easily; if r_m are previously found roots, then the deflated expression is simply

$$t_{i+1} = t_i - \alpha\left[\mathrm{Tr}(A^{-1}\Delta\mathbf{k}\cdot A_{\mathbf{k}}) - \sum_m \frac{1}{t_i - r_m}\right]^{-1}. \tag{17}$$

Note that often we are searching for roots in the neighborhood of a previously found root, and we may thus initialize the Newton iteration with a value of t_0 near the previous root, greatly accelerating convergence. Once all the roots in t and hence the corresponding $\mathbf{k}$ are found, the first requirement of the oracle is satisfied.

We return now to the difficulty of computing A^{-1}. One would naturally expect that the step size tends to zero near

an eigenvalue t, and thus should be well behaved. However, the naive application of Eq. (17) clearly results in an ill-conditioned A^{-1} as A becomes nearly singular, which arises often in practice during the Newton iteration. In order to compute the update expression in Eq. (17) robustly, we perform a singular value decomposition (SVD) of $A = U\Sigma V^H$. Then, $A^{-1} = V\Sigma^{-1}U^H$, and we may simplify the trace expression:

$$\begin{aligned}\mathrm{Tr}(V\Sigma^{-1}U^H\Delta\mathbf{k}\cdot A_{\mathbf{k}}) &= \mathrm{Tr}[\Sigma^{-1}U^H(\Delta\mathbf{k}\cdot A_{\mathbf{k}})V] \\ &= \mathrm{diag}(\Sigma^{-1})\cdot\mathrm{diag}(U^H(\Delta\mathbf{k}\cdot A_{\mathbf{k}})V), \quad (18)\end{aligned}$$

where the operator $\mathrm{diag}(\cdot)$ in this context refers to the vector formed from the diagonal entries of its argument. The last expression above is still ill-conditioned since Σ may have arbitrarily small entries. However, designate $\sigma_{\min}$ the smallest singular value, then we may compute the update equation robustly by

$$t_{i+1} = t_i - \alpha\frac{\sigma_{\min}}{\sum_j \frac{\sigma_{\min}}{\sigma_j}u_j^H(\Delta\mathbf{k}\cdot A_{\mathbf{k}})v_j - \sum_m (t_i - r_m)^{-1}}, \quad (19)$$

where the sum runs over all the σ_j (the j-th singular value), and u_j and v_j are the j-th left and right singular vectors, respectively. Equation (19) is correct and numerically stable even when A is singular. In the numerical implementation of Eq. (19), we compute the SVD of the matrix A using standard routines in LAPACK [31].

Finally, we note that the field pattern at any point on an EFS can be computed easily by computing the null space of A evaluated at the point. The null space will be rank one at all non-degenerate points on an EFS and gives the vector u in Eqs. (6)–(8), from which the field at all boundary points may be found. In fact, during the iteration, successive approximations to the null space are computed from the SVD in v_n. In our implementation, the DtN map is computed using the Dirichlet map derived from a cylindrical wave basis [19]. This map may be inverted to obtain the cylindrical wave coefficients and obtain the field at any point in the interior. In the general case, the field along the boundary of the cell uniquely determine the fields in the interior via knowledge of the Green's function of the cell.

C. Eigenderivatives

In order to compute the tangent vector to an EFS, we take the gradient of $\det A(\mathbf{k})$. The same derivative expression as in Eq. (12) is used to obtain

$$\mathbf{g} \equiv \nabla_{\mathbf{k}}\det A(\mathbf{k}) = \mathrm{Tr}[\mathrm{adj}(A)A_{\mathbf{k}}]. \quad (20)$$

Using Eq. (14),

$$\mathbf{g} = \begin{bmatrix}\mathrm{Tr}[\mathrm{adj}(A)A_{k_x}] \\ \mathrm{Tr}[\mathrm{adj}(A)A_{k_y}]\end{bmatrix}. \quad (21)$$

Since the level set $\det A(k) = 0$ corresponds to an EFS, the gradient vector g must be normal to the EFS. A tangent vector may be obtained by a 90° rotation. Since $\mathbf{g}$ only depends on $A_{\mathbf{k}}$, which was computed in solving the eigensystem, computing tangent vectors requires very little additional effort.

Note that in Eq. (21), $\mathrm{adj}(A)$ cannot be computed from A^{-1} since it is assumed that A is exactly singular; we must compute the adjugate by another method. Utilizing the SVD of A and the identity $\mathrm{adj}(A) = \det(UV^H)V\mathrm{adj}(\Sigma)U^H$, we notice that by the cofactor expansion of $\mathrm{adj}(\Sigma)$ at a singular A, $\mathrm{adj}(\Sigma)$ may only have a single nonzero element in the last entry of its diagonal, assuming A only has a single zero eigenvalue. This is true except at points where multiple EFS intersect, when $\mathrm{adj}(\Sigma) = 0$ identically due to multiple zero eigenvalues of A. In the case where A only has a single zero eigenvalue,

$$\mathrm{adj}(\Sigma) = \mathrm{diag}(0,\ldots,s) \quad \text{where } s = \prod_{j=1}^{n-1}\sigma_j, \quad (22)$$

assuming the singular values are ordered such that $\sigma_i \geq \sigma_j$ for $i < j$. The quantities s, u_n, and v_n can be obtained from the SVD of the matrix A, which was already computed in the previous section when solving the eigensystem.

The group velocity for a given $(\omega, \mathbf{k})$ point on an EFS may be obtained by setting to zero the total derivative of the eigenequation with respect to frequency:

$$\frac{d}{d\omega}\det A = \mathbf{g}\cdot\frac{d\mathbf{k}}{d\omega} + \mathrm{Tr}[\mathrm{adj}(A)A_\omega] = 0, \quad (23)$$

where A_ω is the partial derivative of $A(k;\omega)$ with respect to ω evaluated at $\mathbf{k}$ and ω, and can be obtained from Eq. (9) by replacing each Λ block by its frequency derivative. The DtN map for simple geometries can be computed from an analytic set of basis functions, so the frequency derivative of the DtN map may also be computed analytically. For arbitrary geometries, the lack of analytic basis functions does not pose a problem as it is always possible to compute derivatives using techniques such as automatic differentiation [32].

Since $\mathbf{g}$ is parallel to $d\mathbf{k}/d\omega$, from Eq. (23), we obtain

$$\frac{d\mathbf{k}}{d\omega} = -\mathrm{Tr}[\mathrm{adj}(A)A_\omega]\frac{\mathbf{g}}{\|\mathbf{g}\|^2}. \quad (24)$$

Using the technique of computing $\mathrm{adj}(A)$ using the SVD of A from above, the magnitude of the inverse group velocity is simplified to

$$\left\|\frac{d\mathbf{k}}{d\omega}\right\| = \frac{s}{\|\mathbf{g}\|}u_n^H A_\omega v_n. \quad (25)$$

4. SAMPLING EFSs

Using the oracle described, we may now construct the complete set of EFSs at a constant frequency. Instead of using one of the standard methods [27,28], we implement a simpler algorithm that takes advantage of the particular properties of EFSs to improve sampling efficiency. Our algorithm is quite robust in all the cases that we tested.

An EFS must be a closed loop when the first BZ is treated as a toric space. Therefore, determining a single point on an EFS allows tracing of the entire curve, which is the basic principle of our algorithm. In most photonic crystals, all EFSs intersect the boundaries of the irreducible BZ (IBZ); therefore, determining the set of EFS intersections with the IBZ boundary should produce at least two points per EFS. If there are EFSs that lie completely within the IBZ, then additional

intersections with segments interior to the IBZ are needed. We do not consider this case here.

Our algorithm works by first determining the set of all EFS intersections with the boundaries of the IBZ by invoking the oracle on the IBZ edges. At each intersection point, the tangent vector along the EFS is determined and oriented toward the interior of the IBZ. These points along with their tangent vectors serve as seed points for growing the curves incrementally. We will refer to each of these curves as "fragments", and an interior endpoint of a fragment as "active" if it is still possible to grow the fragment from it.

Once the initial active fragment endpoints are determined, we incrementally advance the endpoints by a predetermined sampling length δ, while merging fragments that have endpoints in close proximity. Each point-tangent pair $(\mathbf{k}, \mathbf{d})$ in the set of active endpoints is advanced by performing a line search along the line $\mathbf{k} + \delta\mathbf{d} + t\mathbf{d}^{\perp}$, where $\mathbf{d}^{\perp}$ is a 90° rotation of $\mathbf{d}$, and t is the eigenvalue to be found as described in the preceding sections. This is illustrated in Fig. 2 for a single fragment. After each set of updates to the active set, we check the pairwise distances between the fragment endpoints, and for endpoints within 2δ of each other, we merge the fragments and consider that particular EFS sample complete. We proceed in this way, alternating between growing fragments from their endpoints and merging fragments until the total number of fragments is half its initial value, indicating that each starting fragment has been paired with another. The steps of the algorithm are summarized in Fig. 3. The choice of δ depends on the maximum expected curvature of an EFS, and should be chosen so that the length of an EFS is approximately δ times the number of desired sample points per EFS contour. This method may occasionally fail at points where EFS have very small radii of curvature. In that case, δ must be reduced and an iteration limit can be imposed to guarantee termination. One may easily imagine more sophisticated methods that use an adaptive choice of δ. However, for the numerical examples shown in the next section, we have found the procedure as described above to be quite adequate.

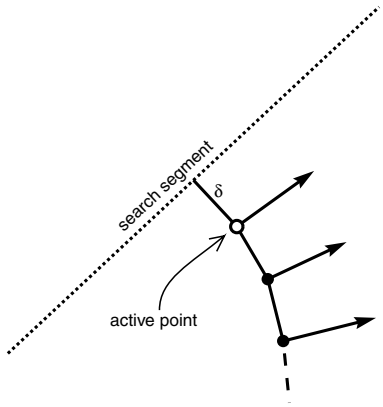

Fig. 2. Illustration of perpendicular search direction for extending fragments. The point of the fragment in the active set is shown in an open circle while the dashed line shows the line segment over which the next point is searched. The arrows indicate the directions of the group velocity direction (gradient vectors) of the EFS (the vector **g** in the text).

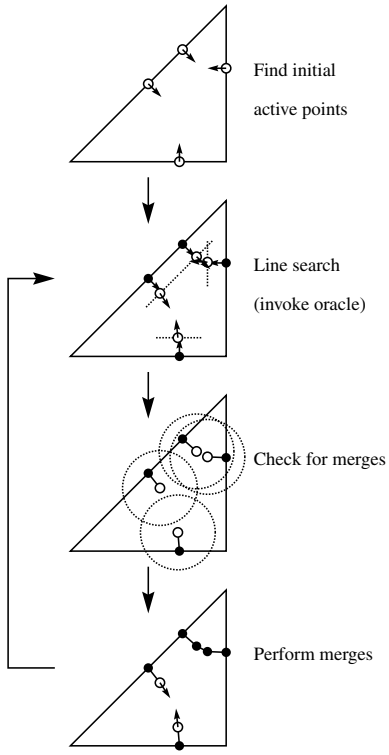

Fig. 3. Steps of the EFS sampling algorithm in the IBZ of a square lattice. Open circles indicate active endpoints of fragments.

5. COMPUTING DENSITY OF STATES

In order to compute the DOS, when sampling the EFS we record the value of the magnitude of the inverse group velocity at each sample point. The DOS can then be directly obtained using Eq. (2) by integrating along each EFS. Since the EFS are closed contours, and the DOS varies smoothly along the contours, we may apply the circular trapezoidal or Simpson's rules directly to the sampled data or to a polynomial spline interpolation. We fit the EFS contours within the IBZ to an interpolating polynomial spline and perform the integration using an arc length parameterization.

Calculation of the local density of states (LDOS) is also possible under this framework. Computing the LDOS requires the field pattern of the bands, which can be obtained using the method described in Section 3. The details will be omitted here since they are beyond the scope of this paper.

6. NUMERICAL EXAMPLES

To evaluate our method, we present some numerical examples in this section. In all these examples, we exploit symmetry to calculate EFS over the IBZ instead of the entire BZ. For all the examples presented, run times reported are for a 2.3 GHz AMD Phenom 9600 workstation.

First we consider the EFS of a square lattice of air holes in a dielectric medium for the H polarization (in-plane electric field). In Fig. 4, we reproduce figure 1 of [2] for a photonic crystal of air holes with radius $0.35a$, where a is the lattice constant in silicon ($n^2 = \epsilon = 12$). Here the DtN map was discretized using eight field points per edge of the unit cell and the EFS was discretized with $\delta = 0.005$. The curves shown correspond to EFSs of the lowest order band up to

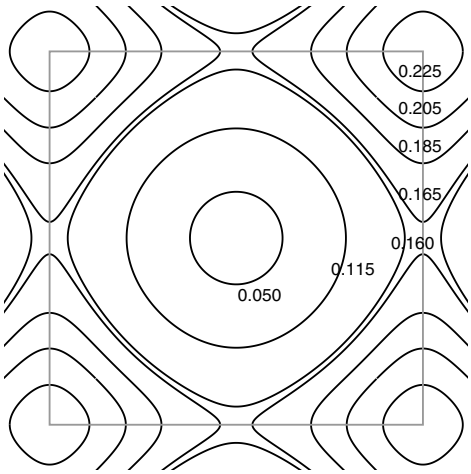

Fig. 4. Several EFS for a photonic crystal of air holes in silicon. Frequencies of EFS are given in units of $2\pi c/a$. The boundary of the first BZ is outlined in gray. The curves are essentially identical to those in originally computed in [2].

frequencies just below the bandgap starting at around $0.24 \times 2\pi c/a$. The curves when overlaid with those in [2] are indistinguishable. The entire figure took less than 1 s to generate. Using the MIT Photonic Bands package [33], **k**-points could be solved at a rate of ~450 per second. To produce a figure of comparable resolution (128^2 points) by binning would take over 30 min. In practice, a root-finding method similar to the one presented can be used, and to obtain fully resolved curves (200 points per curve, 10 root-finder iterations per point), the computation would take still take over 30 min.

Next we consider the DOS of a triangular lattice of GaAs rods for both polarizations. The DOS is shown in Fig. 5, which correspond to figures 3 and 4 of [34]. The locations of the van Hove singularities and relative magnitude of the DOS curves agree with those in [34] within 5% up to about $0.8 \times 2\pi c/a$. For the plots shown here, 11 discretization points were used per

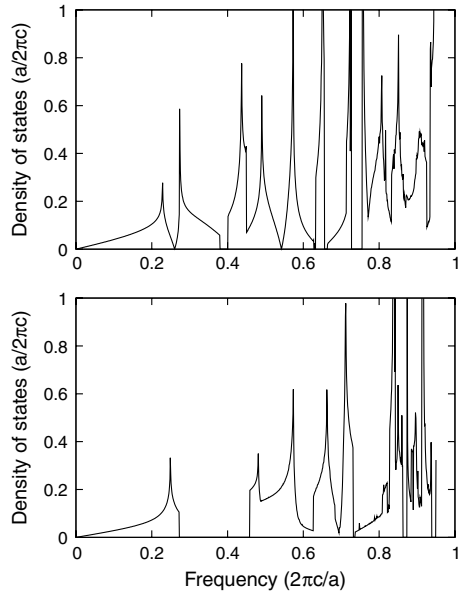

Fig. 5. Density of states for a triangular lattice of GaAs rods ($n = 3.6$, $r = 0.43a$) in air for E polarization (top) and H polarization (bottom).

edge of the hexagonal unit cell and $\delta = 0.001$ in units of the reciprocal lattice constant and each plot took approximately 10 min to generate.

Finally, we remark on the convergence of our method. In order to quantitatively analyze convergence, we computed the DOS at particular frequencies for increasing numbers of discretization points per edge of the unit cell (N) for the structure in the first example above. For the lower frequency bands below $\omega = 0.4 \times 2\pi c/a$, convergence to 0.5% relative error is achieved using $N = 11$ points per edge. Higher frequency bands are more difficult for this class of algorithm. The main cause for nonconvergence in our implementation is the failure of the oracle in finding seed points in the initial scan around the IBZ. By increasing the tolerance of the iterative root-finder or by manually specifying the seed points, the method becomes quite robust and achieves convergence with $N = 16$ for frequencies up to $\omega = 0.8 \times 2\pi c/a$.

7. CONCLUSION AND FINAL REMARKS

We have presented an efficient method to compute the EFSs of photonic crystals, as well as the DOS, using DtN maps. The method is particularly efficient at computing EFS and DOS at a single frequency or for a narrow range of frequencies. The speed of this method enables the design and optimization of large structures under the framework of Hamiltonian optics. Our method may generalize to the 3D case, with the contouring algorithm replaced by any surface meshing method, which typically successively refines a coarse mesh using an oracle exactly like the one described. Finally, the method is also applicable for computing EFS of phonon or electronic band structures as long as the DtN map of the unit cell, along with its derivatives, is available.

ACKNOWLEDGMENTS

This work is supported in part by the United States Air Force Office of Scientific Research (USAFOSR) grant FA9550-09-1-0704, and the National Science Foundation (NSF) grant DMS-0968809. Victor Liu is supported by a Stanford Graduate Fellowship.

REFERENCES

1. M. Notomi, "Theory of light propagation in strongly modulated photonic crystals: Refractionlike behavior in the vicinity of the photonic band gap," Phys. Rev. B **62**, 10696–10705 (2000).
2. C. Luo, S. G. Johnson, J. D. Joannopoulos, and J. B. Pendry, "All-angle negative refraction without negative effective index," Phys. Rev. B **65**, 201104 (2002).
3. X. Yu and S. Fan, "Bends and splitters for self-collimated beams in photonic crystals," Appl. Phys. Lett. **83**, 3251–3253 (2003).
4. D. Pustai, S. Shi, C. Chen, A. Sharkawy, and D. Prather, "Analysis of splitters for self-collimated beams in planar photonic crystals," Opt. Express **12**, 1823–1831 (2004).
5. L. Verslegers, P. B. Catrysse, Z. Yu, and S. Fan, "Deep-subwavelength focusing and steering of light in an aperiodic metallic waveguide array," Phys. Rev. Lett. **103**, 033902 (2009).
6. P. Russel and T. Birks, "Hamiltonian optics of nonuniform photonic crystals," J. Lightwave Technol. **17**, 1982–1988 (1999).
7. Y. A. Urzhumov and D. R. Smith, "Transformation optics with photonic band gap media," Phys. Rev. Lett. **105**, 163901 (2010).
8. E. Yablonovitch, "Inhibited spontaneous emission in solid-state physics and electronics," Phys. Rev. Lett. **58**, 2059–2062 (1987).
9. J. D. Joannopoulos, S. G. Johnson, J. N. Winn, and R. D. Meade, *Photonic Crystals* (Princeton Univ. Press, 2008).

10. Z. Jacob, I. Smolyaninov, and E. Narimanov, "Broadband Purcell effect: Radiative decay engineering with metamaterials," arXiv:0910.3981v2 (2009).
11. Z. Yu, A. Raman, and S. Fan, "Fundamental limit of nanophotonic light trapping in solar cells," Proc. Natl. Acad. Sci. USA **107**, 17491–17496 (2010).
12. M. Ibanescu, E. J. Reed, and J. D. Joannopoulos, "Enhanced photonic band-gap confinement via van Hove saddle point singularities," Phys. Rev. Lett. **96**, 033904 (2006).
13. G. Gilat and L. J. Raubenheimer, "Accurate numerical method for calculating frequency-distribution functions in solids," Phys. Rev. **144**, 390–395 (1966).
14. G. Lehmann and M. Taut, "On the numerical calculation of the density of states and related properties," Phys. Status Solidi B **54**, 469–477 (1972).
15. J. Hama, M. Watanabe, and T. Kato, "Correctly weighted tetrahedron method for k-space integration," J. Phys. Condens. Matter **2**, 7445 (1990).
16. R. C. McPhedran, L. C. Botten, J. McOrist, A. A. Asatryan, C. M. de Sterke, and N. A. Nicorovici, "Density of states functions for photonic crystals," Phys. Rev. E **69**, 016609 (2004).
17. C. J. Pickard and M. C. Payne, "Extrapolative approaches to brillouin-zone integration," Phys. Rev. B **59**, 4685–4693 (1999).
18. E. Moreno, D. Erni, and C. Hafner, "Band structure computations of metallic photonic crystals with the multiple multipole method," Phys. Rev. B **65**, 155120 (2002).
19. J. Yuan and Y. Y. Lu, "Photonic bandgap calculations with Dirichlet-to-Neumann maps," J. Opt. Soc. Am. A **23**, 3217–3222 (2006).
20. Y. Huang, Y. Y. Lu, and S. Li, "Analyzing photonic crystal waveguides by Dirichlet-to-Neumann maps," J. Opt. Soc. Am. B **24**, 2860–2867 (2007).
21. Z. Hu and Y. Y. Lu, "Efficient analysis of photonic crystal devices by Dirichlet-to-Neumann maps," Opt. Express **16**, 17383–17399 (2008).
22. V. Liu, Y. Jiao, D. A. B. Miller, and S. Fan, "Design methodology for compact photonic crystal based wavelength division multiplexers," Opt. Lett. **36**, 591–593 (2011).
23. Z. Hu and Y. Y. Lu, "Improved bends for two-dimensional photonic crystal waveguides," Opt. Commun. **284**, 2812–2816 (2011).
24. J. Yuan, Y. Y. Lu, and X. Antoine, "Modeling photonic crystals by boundary integral equations and Dirichlet-to-Neumann maps," J. Comp. Phys. **227**, 4617–4629 (2008).
25. L. C. Botten, N. A. Nicorovici, R. C. McPhedran, C. M. Sterke, and A. A. Asatryan, "Photonic band structure calculations using scattering matrices," Phys. Rev. E **64**, 046603 (2001).
26. Z.-Y. Li and L.-L. Lin, "Photonic band structures solved by a plane-wave-based transfer-matrix method," Phys. Rev. E **67**, 046607 (2003).
27. T. S. Newman and H. Yi, "A survey of the marching cubes algorithm," Comput. Graph. **30**, 854–879 (2006).
28. J.-D. Boissonnat and S. Oudot, "Provably good sampling and meshing of surfaces," Graph. Models **67**, 405–451 (2005).
29. A. Quarteroni, *Numerical Mathematics* (Springer, 2000).
30. G. Strang, *Linear Algebra and Its Applications* (Harcourt Brace Jovanovich, 1988).
31. E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen, *LAPACK Users' Guide*, 3rd ed. (Society for Industrial and Applied Mathematics, 1999).
32. A. Griewank and A. Walther, *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation, no. 105 in Other Titles in Applied Mathematics*, 2nd ed. (SIAM, 2008).
33. S. G. Johnson and J. D. Joannopoulos, "Block-iterative frequency-domain methods for Maxwell's equations in a plane wave basis," Opt. Express **8**, 173–190 (2001).
34. K. Busch and S. John, "Photonic band gap formation in certain self-organizing systems," Phys. Rev. E **58**, 3896–3908 (1998).