Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Computational Physics 204 (2005) 65–81
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

Photonic band structure calculations using nonlinear eigenvalue techniques

Alastair Spence [a], Chris Poulton [b,*]

[a] Department of Mathematical Sciences, University of Bath, Claverton Down, Bath BA2 7AY, UK
[b] High Frequency and Quantum Electronics Laboratory, University of Karlsruhe, Kaiserstrasse 12, Karlsruhe 76128, Germany

Received 5 July 2004; accepted 29 September 2004
Available online 5 November 2004

Abstract

This paper considers the numerical computation of the photonic band structure of periodic materials such as photonic crystals. This calculation involves the solution of a Hermitian nonlinear eigenvalue problem. Numerical methods for nonlinear eigenvalue problems are usually based on Newton's method or are extensions of techniques for the standard eigenvalue problem. We present a new variation on existing methods which has its derivation in methods for bifurcation problems, where bordered matrices are used to compute critical points in singular systems. This new approach has several advantages over the current methods. First, in our numerical calculations the new variation is more robust than existing techniques, having a larger domain of convergence. Second, the linear systems remain Hermitian and are nonsingular as the method converges. Third, the approach provides an elegant and efficient way of both thinking about the problem and organising the computer solution so that only one linear system needs to be factorised at each stage in the solution process. Finally, first- and higher-order derivatives are calculated as a natural extension of the basic method, and this has advantages in the electromagnetic problem discussed here, where the band structure is plotted as a set of paths in the (ω,k) plane.

Keywords: Photonic band-gap materials; Photonic crystals; Nonlinear eigenvalue problems; Newton's method

1. Introduction

We consider a class of problems which arises when a wave moves through a structure with periodic inhomogeneities. For any wave moving through a periodic material, there exists a relationship between the wave's frequency and its spatial periodicity known as a dispersion relation. The set of all dispersion curves

* Corresponding author.
E-mail address: c.poulton@ihq.uni-karlsruhe.de (C. Poulton).

0021-9991/$ - see front matter © 2004 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2004.09.016

is known as the band structure of the medium. The nature of the band structure is most important when calculating the properties of waveguides and also of photonic crystals, where waves possess many interesting characteristics, such as anomolous dispersion, negative refraction, and photonic band-gaps.

There are many different methods available for calculating dispersion relations in the literature, however, most share some common characteristics. Typically a propagating wave is expanded in terms of some specially chosen basis functions, with the expansion coefficients grouped into a vector of unknowns $\mathbf{x}$. The application of boundary conditions then reduces the problem to an eigenvalue problem

$$A(\omega, \mathbf{k}_{\text{bloch}})\mathbf{x} = 0, \tag{1}$$

where $A(\omega,\mathbf{k}_{\text{bloch}})$ is a Hermitian matrix whose coefficients depend not only on the frequency and the wavevector but also on the material properties of the structure under consideration. The eigenvectors $\mathbf{x}$ represent the modes (the eigenmodes) which are allowed to propagate in the structure, as represented in the appropriate basis.

These problems can then be classified as *linear* and *nonlinear* approaches, according to the form of the eigenvalue problem which results from the application of the method. The linear approaches include finite difference methods and finite element methods [1,2]. Among the nonlinear approaches are plane-wave expansions, transfer matrix methods and approaches based on integral equations [4–6]. The linear methods are characterized by very large matrices and hence put heavy demands on computing power, however, the matrices involved have the advantages of sparsity and of a known structure. The nonlinear methods by contrast typically involve the solution of relatively small, dense matrix systems. The disadvantage of these systems is that they depend on the parameters $(\omega,\mathbf{k}_{\text{bloch}})$ in a complicated manner.

In this paper, we concentrate on ways of facilitating the solution of these nonlinear matrix problems. Clearly the problem is equivalent to finding zeros of the determinant of $A(\omega,\mathbf{k}_{\text{bloch}})$, and if no zeros exist for a given value of ω then there exists a 'photonic band-gap', a region for which all propagation in the material is critically damped. A näive approach then would be to use a multitude of successive matrix evaluations and to look for parameters for which the determinant is vanishingly small. This approach is expensive in terms of computer time and also runs the risk of 'missing' a mode. More sophisticated search procedures can also be employed, notably numerical packages such as BRENT [7], and while procedures such as these fare better they are not optimized for matrix problems, or for those which depend on more than one parameter. A better approach is to use the techniques of matrix algebra, usually designed for linear problems, and adapt them to the new nonlinear situation. Such techniques were initiated by the early work in [15,14], where the technique of Rayleigh quotient iteration for the standard eigenvalue problem for a symmetric matrix was extended to the nonlinear case. These ideas were then extended in [13,12] using ideas from Newton's method for systems. More recently, in [16] the steps in the Newton's method were reorganised to provide the 'residual correction method' for computing various species of nonlinear waves. In [10], it was shown how high-order derivatives in nonlinear eigenvalue problems can be computed efficiently, and in [11] the elegant idea of deriving a scalar function that has the same zeros as $\det A(\omega,\mathbf{k}_{\text{bloch}})$ is discussed. In this paper, we present a straightforward self-contained treatment of the idea in [11], appropriate for the computation of band diagrams and physical properties of photonic crystals. In addition we introduce a variation of the numerical approach that retains the Hermitian character of the problem and is more robust in our computations. This variation is justified using a simple determinant argument.

To demonstrate the methods discussed here, we consider the classical problem of a scalar electromagnetic wave moving through a two-dimensional geometry of cylindrical inclusions. To solve this problem we use a multipole method, which has the advantage that the resulting fields can be represented semi-analytically, allowing the solution to be easily checked. The procedure, however, involves the solution of a dense, complex-valued matrix problem of the type stated previously. The problem is therefore one which is interesting physically and provides a challenging problem for nonlinear eigenvalue techniques.

It will be shown that the algorithms discussed here not only exhibit improvements in speed and robustness, but also offer additional extra information: specifically, the eigenmodes and group velocities of the wave emerge as a by-product of the solution process. In addition, a natural extension of the method means that additional derivatives of the dispersion curves can be calculated in a straightforward manner.

2. Physical background to the problem

We consider first the problem depicted in Fig. 1, which depicts a transverse electromagnetic wave, of either TE or TM polarization, propagating through a doubly periodic lattice of circular inclusions. This is the classical problem of wave propagation in a photonic crystal, which is formulated in [9] and elsewhere. The inclusions have a radius a, are separated by a distance d, and possess transport properties which are different from the matrix material which surrounds them. The geometry of the array is represented by the lattice vector $\boldsymbol{R}_{\mathrm{p}}$, where $p = (md, nd) \in \mathbb{R}^2$ is a multi-index which points to the centre of the pth cylinder.

Because the propagation is in-plane, we can represent the wave by a single scalar potential $u(r,\theta)$, which represents either the z-component of the magnetic field, if the wave happens to be TE polarized, or the z-component of the electric field, for TM polarization.

Considering the steady-state situation, then within the matrix material the field u satisfies the Helmholtz equation

$$(\Delta + \omega^2/c^2)u = 0. \tag{2}$$

Here, ω is the characteristic frequency of the wave, and c is the wave-speed within the material. Within the inclusions themselves we have

$$(\Delta + n_c^2\omega^2/c^2)u^{\mathrm{int}} = 0, \tag{3}$$

where n_c is the refractive index of the inclusions, relative to the surrounding material.

If we now assume that the material is strictly periodic then the scalar field u must obey the Bloch–Floquet condition [8]

$$u(\boldsymbol{r} + \boldsymbol{R}_p) = u(\boldsymbol{r})\, \mathrm{e}^{\mathrm{i}\mathbf{k}_{\mathrm{bloch}}\cdot\boldsymbol{R}_p}, \tag{4}$$

where $\mathbf{k}_{\mathrm{bloch}}$ is the Bloch wavevector.

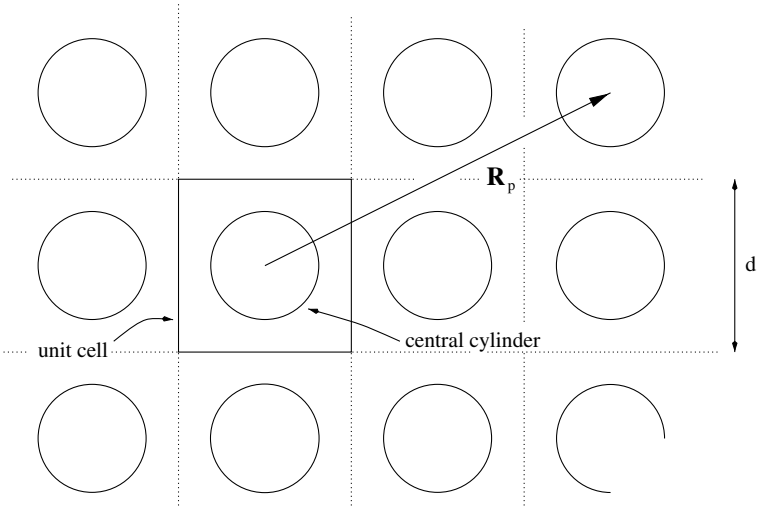

Fig. 1. The two-dimensional array of inclusions, through which the wave propagates.

On the surface of every cylinder we can specify the appropriate boundary conditions; in this paper, we consider an array of perfectly conducting cylinders surrounding by air, through which a TE-polarized mode is propagating, although we note that this method is applicable to any combination of non-dispersive materials and for each polarization. In our chosen situation we can ignore the field interior to the cylinders and specify a Neumann condition on the boundary of each cylinder:

$$\frac{\partial u}{\partial r} = 0 \tag{5}$$

for the exterior field.

The problem is now determined completely. For a given vector $\mathbf{k}_{\text{bloch}}$, the aim is to find values of ω for which u satisfies the relations given above. These solutions correspond to allowed propagating waves in the material, and the function $\omega(\mathbf{k}_{\text{bloch}})$ is known as the dispersion relation for the medium.

According to the Bloch–Floquet theorem [8], the Bloch vector can take any value within the first 'Brillouin zone', that is, within the region $[-\pi/d, \pi/d] \times [-\pi/d, \pi/d]$. All physical functions which depend on the Bloch vector (such as the dispersion relation) will merely repeat themselves periodically outside this zone. Furthermore, the symmetry of the lattice dictates further redundancy within the zone itself: for a square array, only one octant is needed to map the zone entirely. This first irreducible segment is usually written as being bounded by the three points Γ, M and X, as shown in Fig. 2.

Ideally one would like then to plot ω as a function of $\mathbf{k}_{\text{bloch}}$ for the entirety of the irreducible segment. Such plots are cumbersome to represent, however; for this reason the dispersion relation is usually given as a piecewise plot around the edge of the first irreducible segment of the Brillouin zone, as shown in Fig. 2. We note that in any such two-dimensional plot the path of $\mathbf{k}_{\text{bloch}}$ is always fixed, and so the two-variable problem which we introduced originally is in fact a problem for a single variable, which we can label k.

There are many methods available for calculating the dispersion relations. For the purpose of demonstrating our search procedure we employ a multipole method [6] often known as 'the Rayleigh method', which uses a decomposition of the field into regular and irregular cylindrical functions in order to obtain a matrix identity for the propagating modes. This procedure has the advantage that it is semi-analytical, in that it gives a solution in terms of analytical functions, and this makes analysis of the method relatively

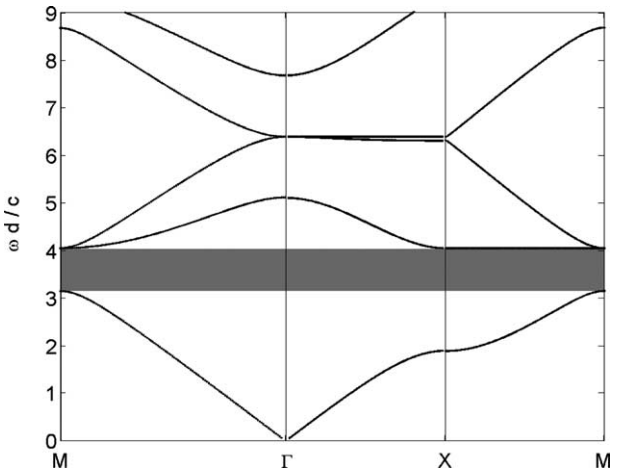

Fig. 2. Dispersion curves for a two-dimensional array of perfectly conducting circular cylinders, with radius $r = 0.40d$. The lowest photonic band-gap region is shaded in grey. The inset shows the first irreducible octant of the Brillouin zone. This curve was produced using Algorithm 3 with a very small step size.

transparent. In order to arrive at the solution, however, one must solve a particulary dense complex-valued matrix, of the form

$$A(\omega, \mathbf{k}_{\text{bloch}})\mathbf{x} = \mathbf{0}, \tag{6}$$

where $\mathbf{x}$ is a vector which contains all the multipole coefficients in the expansion of the propagating mode.

A dispersion diagram such as the one shown in Fig. 2 can have several interesting characteristics. First, the slope of any dispersion curve $\mathrm{d}\omega/\mathrm{d}k$ defines the group velocity in the material. The diagram may also exhibit the presence of *photonic band-gaps*, or regions of frequency where light cannot propagate. Some researchers have also proposed that light in such a periodic structure will behave in similar ways to an electron in a semiconductor, with the second derivative $\mathrm{d}^2\omega/\mathrm{d}k^2$ near the band edge playing the role of the photon's effective mass. These features are important for researchers attempting to create novel devices based on photonic crystals, and it is important to have algorithms which can compute these characteristics reliably and efficiently. The approaches presented in the next section enable the computation of dispersion curves, as well as their higher-order derivatives, in an elegant and stable manner.

3. Nonlinear inverse iteration and the implicit determinant method

In this section, we discuss some numerical algorithms for the solution of nonlinear eigenvalue problems. Nonlinear inverse iteration and techniques derived from Newton's method have been in use since the early 1960s (see, for example [15]). The method presented in [11], which we develop further here, is derived from [19], where the use of bordered matrices to compute bifurcation points in nonlinear problems is introduced. For completeness and to help understand fully the numerical approach, we provide a self-contained account here. In addition we introduce a variation on this method which in our experiments has a larger domain of convergence. We first consider the problem of solving the eigenvalue problem for a specific Bloch vector. Fixing $\mathbf{k}_{\text{bloch}}$, we write the system (6) in the form

$$A(\omega)\mathbf{x} = \mathbf{0}, \tag{7}$$

where $\mathbf{x}$ is normalised by

$$\mathbf{x}^{\mathrm{H}}\mathbf{x} = 1. \tag{8}$$

We denote solutions to (7) by $\mathbf{x}^*$ with eigenvalues ω^*. Throughout the paper, we assume $A(\omega)$ is a smooth function of ω and shall denote its derivatives by $A_\omega(\omega)$, $A_{\omega\omega}(\omega)$, etc.

It is clear that an equivalent formulation of (7) is

$$\det A(\omega) = 0, \tag{9}$$

and since A is Hermitian, (9) represents a real function of the real variable ω. Our problem is then one of finding zeros of a real valued function.

Assume that $\omega^{(i)}$ is an estimate of ω^*. It is natural to try Newton's method to improve $\omega^{(i)}$. The value of $\det A(\omega^{(i)})$ is found simply by an 'LU'-type factorisation for Hermitian matrices (e.g., using the Bunch–Kaufman algorithm [18]), however, it is not so easy to find an efficient formula to evaluate

$$\frac{\mathrm{d}}{\mathrm{d}\omega}(\det A(\omega^{(i)})).$$

For example, Liouville's formula

$$\frac{\mathrm{d}}{\mathrm{d}\omega}(\det A(\omega)) = \quad \operatorname{trace}(A^{-1}(\omega)A_\omega(\omega))\det A(\omega)$$

requires knowledge of the elements of $A^{-1}(\omega)$. This approach has been used in [3].

Most numerical techniques for the solution of $A(\omega)\mathbf{x} = \mathbf{0}$ are extensions of inverse iteration for the standard eigenvalue problem [14], or are variations of Newton's method (see, for example [13,12,23]). Indeed, since there is an intimate connection between inverse iteration and Newton's method [24] several methods can be derived using both approaches. One such algorithm, discussed in [23], is as follows:

Algorithm. (*Nonlinear inverse iteration*) Given $(\mathbf{x}^{(i)},\omega^{(i)})$ with ${\mathbf{x}^{(i)}}^{\mathrm{H}}\mathbf{x}^{(i)} = 1$:

(i) Solve: $A(\omega^{(i)})\mathbf{y}^{(i+1)} = -A_\omega(\omega^{(i)})\,\mathbf{x}^{(i)}$.
(ii) Rescale: $\mathbf{x}^{(i+1)} = \mathbf{y}^{(i+1)}/\|\mathbf{y}^{(i+1)}\|_2$.
(iii) Update ω:

$$\omega^{(i+1)} = \omega^{(i)} - \frac{{\mathbf{x}^{(i+1)}}^{\mathrm{H}} A(\omega^{(i)})\mathbf{x}^{(i+1)}}{{\mathbf{x}^{(i+1)}}^{\mathrm{H}} A_\omega(\omega^{(i)})\mathbf{x}^{(i+1)}}.$$

(iv) Repeat until convergence.

It is shown in [23] that this method has quadratic convergence provided the nondegeneracy condition

$$\mathbf{x}^{*\mathrm{H}} A_\omega(\omega^*)\mathbf{x}^* \neq 0 \tag{10}$$

holds. It is also shown in [23] that (10) implies

$$\frac{\mathrm{d}}{\mathrm{d}\omega}(\det A(\omega^*))|_{\omega=\omega^*} \neq 0, \tag{11}$$

that is, $\det A(\omega)$ crosses through zero with non-zero velocity at $\omega = \omega^*$. (Note, this is a different nondegeneracy condition from the requirement that $A(\omega)$ has a simple zero eigenvalue at ω^* as implied by (12).)

3.1. Derivation of the implicit determinant algorithm

In this section, we introduce an approach to the solution of (9), which we call the "implicit determinant method". This is motivated by the use of bordered systems from numerical bifurcation derived in [19], and has links with the idea of "test functions" to detect singularities in nonlinear problems (see, for example, [22]). The idea is to set up a scalar problem, $f(\omega) = 0$, that has the same roots as (9) but where $f_\omega(\omega)$ is easy to evaluate so that Newton's method is readily implemented. This idea was also explored in [11], where numerical aspects of various possible implementations are also discussed. Here, we present a self-contained account specialised to the Hermitian case, which allows some simplification and a more direct physical interpretation.

The main theoretical tool is the following lemma:

Lemma 1. *Let* $(\mathbf{x}^*,\omega^*)$ *solve* (7) *and* (8) *with* $A(\omega)$ *Hermitian. Assume that zero is a simple eigenvalue of* $A(\omega)$, *so that*,

$$(a) \quad \dim \mathrm{Null} A(\omega^*) = 1. \tag{12}$$

For some $\mathbf{b} \in \mathbb{C}^n$ *assume*

$$(b) \quad \mathbf{b}^{\mathrm{H}}\mathbf{x}^* \neq 0. \tag{13}$$

Then the $(n+1)\times(n+1)$ *matrix* $M(\omega)$ *defined by*

$$M(\omega) = \begin{pmatrix} A(\omega) & \mathbf{b} \\ \mathbf{b}^{\mathrm{H}} & 0 \end{pmatrix}$$

is nonsingular at $\omega = \omega^*$.

Proof. The result follows if we can show that

$$\begin{pmatrix} A(\omega^*) & \mathbf{b} \\ \mathbf{b}^{\mathrm{H}} & 0 \end{pmatrix} \begin{pmatrix} \mathbf{p} \\ q \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ 0 \end{pmatrix} \tag{14}$$

implies that $\mathbf{p} = \mathbf{0}$ and $q = 0$. The first row of (14) gives $A(\omega^*)\mathbf{p} + \mathbf{b}q = \mathbf{0}$, and left multiplication by $\mathbf{x}^{*\mathrm{H}}$ gives $\mathbf{x}^{*\mathrm{H}}A(\omega^*)\mathbf{p} + \mathbf{x}^{*\mathrm{H}}\mathbf{b}q = \mathbf{0}$. The first term is zero since $A(\omega^*)\mathbf{x}^* = \mathbf{0}$, so $q = 0$ using (13). Hence $A(\omega^*)\mathbf{p} = \mathbf{0}$ which implies, using (12), that $\mathbf{p} = \alpha\mathbf{x}^*$, for some scalar α. The second row of (14) gives $\alpha\mathbf{b}^{\mathrm{H}}\mathbf{x}^* = \mathbf{0}$, which implies $\alpha = 0$ using (13) and hence $\mathbf{p} = \mathbf{0}$. (This result is a special case of Lemma 2.8 of [21].) □

First, note that since $M(\omega^*)$ is nonsingular then $M(\omega)$ is nonsingular for ω near ω^* because $A(\omega)$ is a smooth function of ω. Throughout this paper we will assume that $M(\omega)$ is always nonsingular and that any $\mathbf{b}$ satisfies (13). We return to the question of the optimal choice of $\mathbf{b}$ later. It is common to refer to $M(\omega)$ as a *bordered* matrix.

Next, for ω near ω^* and $\mathbf{b}$ satisfying (13) we introduce the linear system

$$\begin{pmatrix} A(\omega) & \mathbf{b} \\ \mathbf{b}^{\mathrm{H}} & 0 \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ f \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ 1 \end{pmatrix}. \tag{15}$$

Since the matrix is nonsingular, $\mathbf{x}$ and f are smooth functions of ω and so we write $\mathbf{x}(\omega)$ and $f(\omega)$. Cramer's rule (see, for example [20, p. 414]) gives

$$f(\omega) = \frac{\det A(\omega)}{\det M(\omega)} \tag{16}$$

and note that since $A(\omega)$ and $M(\omega)$ are both Hermitian then $f(\omega)$ is real. The reason for introducing (15) is now clarified. Since $\det M(\omega)$ is nonzero then $f(\omega) = 0$ if and only if $\det A(\omega) = 0$, and the essence of our method is to seek zeros of $f(\omega)$ rather than $\det A(\omega)$. For completeness we state the following "equivalence" result.

Theorem 1. *Assume the conditions of Lemma* 1 *hold and that* $f(\omega)$ *and* $\mathbf{x}(\omega)$ *are given by* (15). *Then*

(*a*) $f(\omega) = 0$ *if and only if* $\det A(\omega) = 0$.
(*b*) *For* $\omega = \omega^*$,$\mathbf{x}(\omega^*)$ *obtained from* (15) *is a null vector of* $A(\omega^*)$.

Proof. The result of (a) was given above. Since $f(\omega^*) = 0$, the first row of (15) becomes $A(\omega^*)\ \mathbf{x}(\omega^*) = \mathbf{0}$. Hence $\mathbf{x}(\omega^*)$ is a scaled form of $\mathbf{x}^*$ (but will not be a unit vector since $\mathbf{b}^{\mathrm{H}}\ \mathbf{x}(\omega^*) = 1$ rather than $\mathbf{x}(\omega^*)^{\mathrm{H}}\ \mathbf{x}(\omega^*) = 1$). □

The "implicit determinant method" is precisely the application of Newton's method to $f(\omega)$ defined by (15). To do this we need to be able to evaluate $f_\omega(\omega)$ and this is accomplished readily as follows. Differentiating (15) with respect to ω gives

$$\begin{pmatrix} A(\omega) & \mathbf{b} \\ \mathbf{b}^{\mathrm{H}} & 0 \end{pmatrix} \begin{pmatrix} \mathbf{x}_\omega(\omega) \\ f_\omega(\omega) \end{pmatrix} = -\begin{pmatrix} A_\omega(\omega)\mathbf{x}(\omega) \\ 0 \end{pmatrix} \tag{17}$$

and so $f_\omega(\omega)$ is computed by solving a linear system with the *same matrix* as in (15) but with a different right-hand side. This leads to the following algorithm:

Algorithm 1. (*Implicit determinant*) Given $\omega^{(0)}$ and $\mathbf{b} \in \mathbb{C}^n$ such that $M(\omega^{(0)})$ is nonsingular:

(i) Solve (15) with $\omega = \omega^{(i)}$ to find $f(\omega^{(i)})$.
(ii) Solve (17) with $\omega = \omega^{(i)}$ (using $\mathbf{x}(\omega^{(i)})$ obtained from (i) on the right hand side) to find $f_\omega(\omega^{(i)})$.
(iii) Perform a Newton update:

$$\omega^{(i+1)} = \omega^{(i)} - f(\omega^{(i)})/f_\omega(\omega^{(i)}). \tag{18}$$

(iv) Repeat till convergence.

Assuming $A_\omega(\omega^{(i)})$ is easily evaluated, the computation of $f_\omega(\omega^{(i)})$ requires very little extra over the evaluation of $f(\omega^{(i)})$ and the main cost in one Newton step is the 'LU' factorisation of the matrix on the left-hand side of (15). This has the same complexity as the evaluation of $\det A(\omega^{(i)})$.

3.2. Convergence of the method

Standard Newton theory says that this algorithm will converge quadratically provided $f_\omega(\omega^*) \neq 0$. The following lemma gives a nondegeneracy condition that ensures this condition.

Lemma 2. *Assume the conditions of Lemma* 1. *Then*

(*a*) $f_\omega(\omega^*) \neq 0$ *if and only if* $\mathbf{x}(\omega^*)^H A_\omega(\omega^*)\mathbf{x}(\omega^*) \neq 0$.
(*b*) *If* $\mathbf{x}(\omega^*)^H A_\omega(\omega^*)\mathbf{x}(\omega^*) \neq 0$ *the implicit determinant method converges quadratically.*

Proof. The first equation of (17) shows that

$$\mathbf{b} f_\omega(\omega) = -A_\omega(\omega)\mathbf{x}(\omega) - A(\omega)\mathbf{x}_\omega(\omega). \tag{19}$$

Multiplication on the left by $\mathbf{x}(\omega)^{\mathrm{H}}$ gives

$$f_\omega(\omega) = -\mathbf{x}(\omega)^{\mathrm{H}} A_\omega(\omega)\mathbf{x}(\omega) + \mathbf{b}^{\mathrm{H}}\mathbf{x}_\omega(\omega) f(\omega) = -\mathbf{x}(\omega)^{\mathrm{H}} A_\omega(\omega)\mathbf{x}(\omega),$$

where we have used $\mathbf{x}(\omega)^{\mathrm{H}}\mathbf{b} = 1$ and $\mathbf{b}^{\mathrm{H}}\mathbf{x}_\omega(\omega) = 0$. Evaluation at $\omega = \omega^*$ gives the result stated in (a). Since $f(\omega^*) = 0$, $f_\omega(\omega^*) \neq 0$, ω^* is a simple root and hence Newton's method converges quadratically for a close enough starting guess. □

We note that manipulation of the first equation of (15) shows

$$f(\omega) = -\mathbf{x}(\omega)^{\mathrm{H}} A(\omega)\mathbf{x}(\omega)$$

and so the formula (18) can be written as

$$\omega^{(i+1)} = \omega^{(i)} - \frac{\mathbf{x}(\omega^{(i)})^{\mathrm{H}} A(\omega^{(i)})\mathbf{x}(\omega^{(i)})}{\mathbf{x}(\omega^{(i)})^{\mathrm{H}} A_\omega(\omega^{(i)})\mathbf{x}(\omega^{(i)})}, \tag{20}$$

which is very similar to the correction formula in the inverse iteration algorithms in [12,23], though the $\mathbf{x}(\omega^{(i)})$ vectors are calculated using a nonsingular system and depend on the choice of $\mathbf{b}$. We note that, not surprisingly, the condition for quadratric convergence in the implicit determinant method is precisely that given by equation (10), which is also needed for nonlinear inverse iteration to converge quadratically.

3.3. Numerical results

In Table 1, we illustrate the convergence of the implicit determinant algorithm (Algorithm 1), and compare it to the convergence of the inverse iteration method. In both cases the same initial estimate has been used: if we represent the solution eigenvector $\mathbf{x}^*$ as a set of complex numbers in decreasing order of magnitude, then for inverse iteration the initial estimate was $\mathbf{x}^{(0)} = (1,0,0,\ldots,0)$, and for Algorithm 1 the initial estimate is given by $\mathbf{b} = (1,0,0,\ldots,0)$. For the first three iterations the ratio $(\omega^{(i+1)} - \omega^*)/(\omega^{(i)} - \omega^*)^2$ remains roughly constant, showing that the convergence is quadratic for both inverse iteration and for Algorithm 1. For iterations $i = 3,4,5$ this quantity can no longer be reliably computed using 32 bit precision.

In the example given in Table 1, it should be noted that we have chosen a 'reasonable' starting value for $\mathbf{x}^{(0)}$, since it is known on physical grounds that the multipole coefficients decay with increasing order. We would expect additionally that the convergence of the implicit determinant algorithm depends on the choice of the vector $\mathbf{b}$. To see how this is so, we present additional calculations for the convergence when $\mathbf{b} = (0,1,0,0,\ldots)$ and when $\mathbf{b} = (0,0,1,0,\ldots)$. These are depicted in Table 2. We can see that Algorithm 1 maintains quadratic convergence for these different values of $\mathbf{b}$, however, the asymptotic coefficients change, and we see that the choice of $\mathbf{b}$ can affect the rate of convergence.

3.4. An improved algorithm

It is clear from the above discussion and numerical results that the implicit determinant algorithm depends on the choice of $\mathbf{b}$. We now discuss the optimal choice for $\mathbf{b}$ and hence derive an improved implicit determinant algorithm.

First, applying Cramer's rule to (17) we obtain

$$f_\omega(\omega) = \frac{\det D(\omega)}{\det M(\omega)}, \tag{21}$$

where we introduce a new bordered matrix $D(\omega)$ [1]

$$D(\omega) = \begin{pmatrix} A(\omega) & -A_\omega(\omega)\mathbf{x}(\omega) \\ \mathbf{b}^{\mathrm{H}} & 0 \end{pmatrix}. \tag{22}$$

Thus from (16) and (21) the Newton correction equation (18) can be written as

$$\omega^{(i+1)} = \omega^{(i)} - \frac{\det A(\omega^{(i)})}{\det D(\omega^{(i)})} \tag{23}$$

and it seems clear that it is a good idea to choose $\mathbf{b}$ in (15) and (17) to minimise the correction to $\omega^{(i)}$, that is, $\mathbf{b}$ should be chosen to maximise $|\det D(\omega^{(i)})|$. We show in Appendix Athat at the root ω^*, $|\det D(\omega^*)|$ is maximised if $\mathbf{b}$ is chosen in the direction $\mathbf{x}^*$. Since this is unknown in practice, this suggests that we choose $\mathbf{b}$ as our current best guess, namely, as (normalised) $\mathbf{x}(\omega^{(i)})$. This choice is reinforced by the fact, also proved in Appendix A, that this choice for $\mathbf{b}$ also maximises $|\det M(\omega^*)|$. Thus we obtain the following improved algorithm:

[1] We note that the matrix $D(\omega)$ is not used in practice, rather it is only used in the discussion of how best to choose $\mathbf{b}$. It is, however, used in the evaluation of derivatives in [10], but we do not utilize it here since it would destroy the Hermitian character of the algorithm.

Table 1
Comparison of the convergence of the inverse iteration method with Algorithm 1, starting at a point $w^{(0)} - w^* \approx 0.08$

i	Inverse iteration		Algorithm 1	
	$\omega^{(i)} - \omega^*$	$(\omega^{(i+1)} - \omega^*)/(\omega^{(i)} - \omega^*)^2$	$\omega^{(i)} - \omega^*$	$(\omega^{(i+1)} - \omega^*)/(\omega^{(i)} - \omega^*)^2$
0	0.07765013345804	2.03893280938146	0.07765013345804	−2.97412410440143
1	0.01229383350918	2.05758075807715	−0.01793260984713	−3.09013884544107
2	0.00031097934503	2.12218663769713	−0.00099372220213	−3.09342861476209
3	0.00000020523275	–	−0.00000305471069	–
4	0.00000000000430	–	−0.00000000009147	–
5	−0.00000000000001	–	0.00000000000000	–

Table 2
Illustration of the convergence of Algorithm 1, given for $\mathbf{b} = (0,1,0,0,\ldots)$ and $\mathbf{b} = (0,0,1,0,\ldots)$

i	$\mathbf{b} = (0,1,0,0,\ldots)$		$\mathbf{b} = (0,0,1,0,\ldots)$	
	$\omega^{(i)} - \omega^*$	$(\omega^{(i+1)} - \omega^*)/(\omega^{(i)} - \omega^*)^2$	$\omega^{(i)} - \omega^*$	$(\omega^{(i+1)} - \omega^*)/(\omega^{(i)} - \omega^*)^2$
0	0.07765013345804	2.29796743715589	0.07765013345804	8.96340344231721
1	0.01385569399439	2.27184085966440	0.05404522850799	8.96865470107751
2	0.00043614858998	2.31215061099929	0.02619642445283	8.97879754809233
3	0.00000043983022	–	0.00616172364812	8.99092203676799
4	0.00000000000948	–	0.00034135688328	9.05101178868019
5	0.00000000000000	–	0.00000105466482	–
6	0.00000000000000	–	0.00000000003169	–
7	0.00000000000000	–	0.00000000000000	–

Algorithm 2. (*Improved implicit determinant*) Given $\mathbf{x}^{(0)} \in \mathbb{C}^n$ and $\omega^{(0)}$ such that, with $\mathbf{b} = \mathbf{x}^{(0)}$, $M(\omega^{(0)})$ is nonsingular. For $i = 0,1,2,\ldots$

(i) Solve (15) with $\mathbf{b} = \mathbf{x}^{(i)}$ and $\omega = \omega^{(i)}$ to find $f(\omega^{(i)})$.
(ii) Solve (17) with $\mathbf{b} = \mathbf{x}^{(i)}$ and $\omega = \omega^{(i)}$ to find $f_\omega(\omega^{(i)})$.
(iii) Newton update:

$$\omega^{(i+1)} = \omega^{(i)} - f(\omega^{(i)})/f_\omega(\omega^{(i)})$$

and

$$\mathbf{x}^{(i+1)} = \mathbf{x}(\omega^{(i)}).$$

(iv) Repeat till convergence.

The work done in this algorithm per step is the same as for Algorithm 1.

One can think of Algorithm 2 as a ''hybrid'' Newton method to find a zero of $\det A(\omega)$. At each step of the iteration a scalar function f is derived that depends on $\mathbf{x}^{(i-1)}$ and has a zero at ω^*. Then *one step* of Newton's method with this particular f is carried out. That gives one step of a quadratically convergent method to find ω, i.e. $\omega^* - \omega^{(i+1)} = O((\omega^* - \omega^{(i)})^2)$. At the next step a new $\mathbf{b}$, namely $\mathbf{b} = \mathbf{x}^{(i)}$, determines a new f and one step of Newton's method is again performed. Overall we clearly maintain a quadratically convergent algorithm as is seen in Table 3. We also see that Algorithm 2 gives significantly better results than Algorithm 1, even though both exhibit quadratic convergence.

Table 3
Comparison of the convergence Algorithm 1 and Algorithm 2, starting at a point 0.3 from w^*

i	Algorithm 1		Algorithm 2	
	$\omega^{(i)} - \omega^*$	$(\omega^{(i+1)} - \omega^*)/(\omega^{(i)} - \omega^*)^2$	$\omega^{(i)} - \omega^*$	$(\omega^{(i+1)} - \omega^*)/(\omega^{(i)} - \omega^*)^2$
0	0.30765013345804	−2.44856034347227	0.30765013345804	−2.44856034347227
1	−0.23175281982956	−3.18330823140392	−0.23175281982956	−0.45975723097739
2	−0.17097347802953	−3.17293879345678	−0.02469327099838	0.37700508905504
3	−0.09275112530593	−3.14288138516662	0.00022988173058	0.14735698976490
4	−0.02703748960838	−3.09782306170899	0.00000000778717	–
5	−0.00226458871925	−3.08309382065328	0.00000000000016	–
6	−0.00001581122140	−4.37217188979223	−0.00000000000001	–
7	−0.00000000109302	–	0.00000000000000	–
8	−0.00000000000003	–	0.00000000000000	–
9	0.00000000000000	–	0.00000000000000	–

The convergence of each algorithm still depends on the initial choice for the vector **b**. A ''sensible choice'' would be one in which $\mathbf{b}^H$ is not orthogonal to the nullvector **x**. This is equivalent to making an ''educated guess'' on physical grounds as to what the solution vector should roughly look like. A poor choice would result in an ill-conditioned bordered matrix M. In the context of computing dispersion curves, where the Bloch parameter k is continuously varied, an appropriate **b** would be the **x** calculated from the previous parameter value.

The domains of convergence for a typical value of $\mathbf{k}_{\text{bloch}}$ (in this case, $\mathbf{k}_{\text{bloch}} = (\pi/2, \pi/2)$) are shown in Fig. 3. In this example, we have increased the distance between the original guess $\omega^{(0)}$ and the final ω^*, and counted the number of iterations each algorithm takes to converge to the correct solution; if after 50 iterations the algorithm had not converged to ω^* the procedure was terminated. One can see that nonlinear inverse iteration and Algorithm 1 possess a similar domain of convergence and that both possess

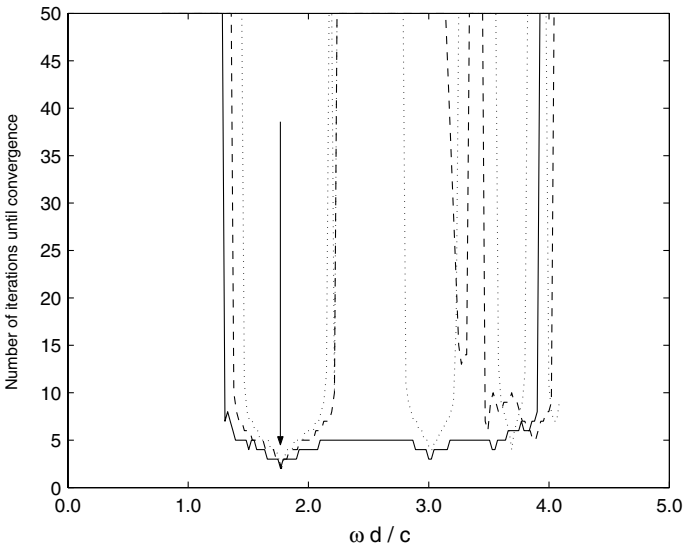

Fig. 3. The domain of convergence of nonlinear inverse iteration (dashed line), Algorithm 1 (dotted line), and Algorithm 2 (solid line), for a Bloch vector half-way along the ΓM curve shown in Fig. 2. The arrow indicates the final solution $\omega^* d/c$.

isolated basins far removed from the final solution. This is in contrast to Algorithm 2, which exhibits a very wide domain of convergence around the solution. We conclude that Algorithm 2, the Improved implicit determinant method, will behave more robustly in situations where the root is difficult to find. This may have implications on the decision of which algorithm to employ when computing dispersion curves, as is done in the following section.

4. The two parameter problem – computation of the band structure

We now consider two parameter problems of the form

$$A(\omega,k)\mathbf{x}=\mathbf{0}, \tag{24}$$

where k is a second real parameter. An interesting and important application of the preceding work lies in the ability to compute paths in the (ω,k) plane quickly and efficiently. Specifically, we wish to compute a path in the (ω,k) plane such that

$$\det A(\omega,k)=0. \tag{25}$$

Here, we have chosen k to be the Bloch vector as outlined in Section 2, however, it should be emphasised that k can in fact represent any other real parameter, such as inclusion radius, refractive index, or length of the unit cell.

The path-following problem can be stated as follows: Assume there is a solution at (ω_1,k_1). We then seek a solution at (ω_2,k_2) where $k_2 = k_1 + \Delta k$, where Δk is small (see Fig. 4). We will see that the methodology of Algorithms 1 and 2 provides a convenient framework for the calculation of the next step in the path, and in fact that the terms in the Taylor series of the curve can be quickly and easily calculated without having repeatedly to solve the matrix system.

For ω near ω^* and $\mathbf{b}$ satisfying (13) consider the linear system of Section 3:

$$\begin{pmatrix} A(\omega,k) & \mathbf{b} \\ \mathbf{b}^{\mathrm{H}} & 0 \end{pmatrix}\begin{pmatrix} \mathbf{x}(\omega,k) \\ f(\omega,k) \end{pmatrix}=\begin{pmatrix} \mathbf{0} \\ 1 \end{pmatrix}. \tag{26}$$

It is been noted in Section 3 that there is an equivalence between the zeros of the determinant of $A(\omega,k)$ and the zeros of the function $f(\omega,k)$. The problem then is one of computing the points on the path for which

$$f(\omega,k)=0. \tag{27}$$

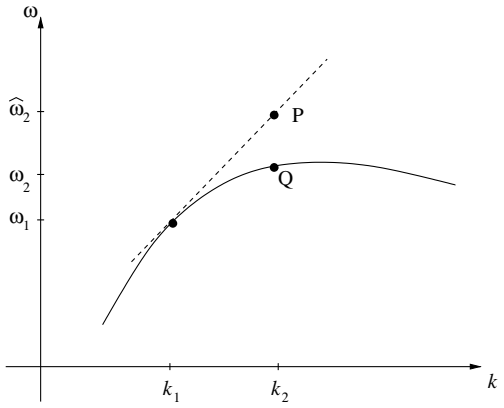

Fig. 4. Schematic to illustrate the approximation of (ω_2,k_2) by $(\hat{\omega},k_2)$.

Differentiating this equation with respect to k, we find that

$$f_\omega \frac{\mathrm{d}\omega}{\mathrm{d}k} + f_k = 0 \tag{28}$$

and so

$$\frac{\mathrm{d}\omega}{\mathrm{d}k} = -\frac{f_k}{f_\omega}. \tag{29}$$

The group velocity at the particular point on the dispersion curve can then be calculated, provided that the derivatives f_ω and f_k can be evaluated. If either of Algorithms 1 or 2 has been used to arrive at the previous point, then f_ω is already given from the solution to Eq. (17). To find f_k we use

$$\begin{pmatrix} A(\omega,k) & \mathbf{b} \\ \mathbf{b}^{\mathrm{H}} & 0 \end{pmatrix} \begin{pmatrix} \mathbf{x}_k(\omega,k) \\ f_k(\omega,k) \end{pmatrix} = -\begin{pmatrix} A_k(\omega,k)\mathbf{x}(\omega) \\ 0 \end{pmatrix}. \tag{30}$$

The matrix on the left-hand side has already been factorized in the previous step, and so only back-substitutions are necessary to solve the system (30). The matrix $A_k(\omega,k)$ must still be evaluated, however; depending on the formulation of the physical problem this may have a simple relationship to the original matrix $A(\omega,k)$. In most cases, including the one examined here, the relationship is not straightforward and so A_k is perhaps best computed using a finite difference step.

This suggests the following "path following algorithm":

Algorithm 3. (*Path following*) Given (ω_1,k_1) satisfying $f(\omega_1,k_1) = 0$ and a given step-size Δk:

(i) Solve (17) and (30) to compute f_ω and f_k.
(ii) Set $k_2 = k_1 + \Delta k$.
(iii) Set $\hat{\omega}_2 = \omega_1 - \Delta k \frac{f_k}{f_\omega}$.
(iv) Newton update: Apply Algorithm 2 to $f(\omega,k_2) = 0$ defined in (26) using $\hat{\omega}_2$ as a starting value for ω_2.

This algorithm has been tested and shows considerable time improvement for our current problem over more direct methods for computing the path, such as successive solves for the determinant using BRENT. There is also an improvement over the nonlinear inverse iteration algorithm in terms of the time taken to complete a given path in the (ω,k) plane, however, because both methods exhibit quadratic convergence (and because the dispersion curves do not bend rapidly) the time improvement is small. The clear advantage of Algorithm 3 is that it does not rely on the ability to solve a near-singular system, as does nonlinear inverse iteration. In addition it provides an elegant framework for the organisation of the computer code.

4.1. Higher-order derivatives and Taylor series

One can easily extend the implicit determinant algorithm to compute higher-order derivatives of ω with respect to k on the path; this has applications for determining the shape of the dispersion surface, or for more efficient computation of dispersion curves. We see that once again the derivatives depend only on the derivatives of the matrix $A(\omega,k)$ with respect to k and ω. Moreover, the computation of these derivatives involves only back-substitutions into an already factorized system.

Differentiating (28) with respect to k we see that

$$f_{\omega\omega}\left(\frac{\mathrm{d}\omega}{\mathrm{d}k}\right)^2 + 2f_{\omega k}\frac{\mathrm{d}\omega}{\mathrm{d}k} + f_\omega \frac{\mathrm{d}^2\omega}{\mathrm{d}k^2} + f_{kk} = 0. \tag{31}$$

Re-arranging, we obtain

$$\frac{d^2\omega}{dk^2} = -\frac{1}{f_\omega}\left[f_{\omega\omega}\left(\frac{d\omega}{dk}\right)^2 + 2f_{\omega k}\frac{d\omega}{dk} + f_{kk}\right]. \tag{32}$$

To compute the second and higher derivatives of f one merely differentiates (26) the appropriate number of times. For example, to compute f_{kk} one solves the following equation:

$$\begin{pmatrix} A(\omega,k) & \mathbf{b} \\ \mathbf{b}^{\mathrm{H}} & 0 \end{pmatrix}\begin{pmatrix} \mathbf{x}_{kk}(\omega,k) \\ f_{kk}(\omega,k) \end{pmatrix} = -\begin{pmatrix} 2A_k(\omega,k)\mathbf{x}_k + A_{kk}(\omega,k)\mathbf{x} \\ 0 \end{pmatrix}. \tag{33}$$

Once again the work has been reduced to finding the higher-order derivatives of the matrix $A(\omega,k)$, and performing back-substitutions into an already factorized matrix. The calculation of mixed derivatives is also straightforward; for example, to compute $f_{\omega k}$ one simply differentiates (17) with respect to ω in order to find the appropriate matrix equation.

The higher-order derivatives often possess important physical meanings – for example, the first derivative $d\omega/dk$ corresponds to the group velocity of a pulse moving through the medium. They can also be used to formulate a higher-order difference scheme for even faster computation of the curve by computing a second-order correction to $\hat{\omega}_2$ at step 3 of Algorithm 3. In addition, the higher-order derivative $f_{\omega\omega}$ could be used in the application of Leguerre's method, which has cubic convergence, for an even faster computation of ω^*.

Alternatively, the Taylor series of the dispersion curves can be computed directly from derivatives of the matrix, rather than from repeated solves to find the first few points on the curve. This gives enormous advantages in several situations: if the matrix is difficult to solve or expensive to evaluate, if only a rough characterisation of the dispersion curves is necessary, or if it is only necessary to calculate the curve in the vicinity of certain (typically symmetry) points.

We present an example of this in Fig. 5. For the lower-order dispersion curve at $\mathbf{k}_{\mathrm{bloch}} = (\pi/2,0)$, we have computed that $\omega = 1.215059$, $d\omega/dk = 0.700711$ and $d^2\omega/dk^2 = -0.25142$. The dispersion curve can then be approximated by the equation

$$\omega(k) = (1.215059) + (0.700711)(k - \pi/2) + \frac{1}{2}(-0.251420)(k - \pi/2)^2. \tag{34}$$

This curve is given by the solid line in Fig. 5. At the point $\mathbf{k}_{\mathrm{bloch}} = (3.0916,0)$, we compute $\omega = 1.89345$, $d\omega/dk = -3.58381\mathrm{e} - 2$ and $d^2\omega/dk^2 = -0.84536$, leading to the equation

$$\omega(k) = (1.89345) + (-3.58381e - 2)(k - 3.0916) + \frac{1}{2}(-0.84536)(k - 3.0916)^2. \tag{35}$$

These approximations have been directly compared with a step-by-step solution using Algorithm 3, solving for 20 points in succession (crosses in Fig. 5). Although the approximations involve only one matrix solve and five back-substitutions one can see in Fig. 5 that they fit quite well to the more laborious path-following computation over a reasonable range of frequencies.

In the vicinity of the Γ point, the slope of the dispersion curve is related to the homogenisation properties of the material, i.e. the effective properties of the material at long wavelengths. Specifically, one defines the effective refractive index of the periodic material as

$$\lim_{\omega\to 0} N_{\mathrm{eff}} = \frac{dk}{d\omega}. \tag{36}$$

For dilute composites this value will be very close to the approximation of Maxwell-Garnett [17], however, when the filling fraction of the inclusions becomes large the calculation of this quantity is not so straightforward; one can then directly calculate the effective refractive index of a given periodic material

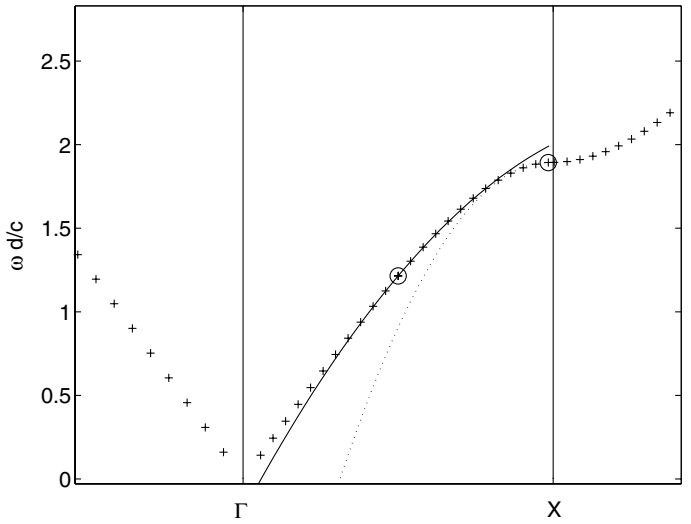

Fig. 5. The dispersion curve computed using Algorithm 3 along the ΓX segment of the Brillouin zone is represented by the sequence of crosses. The solid and fine dotted lines show the Taylor series expansions given by Eqs. (34) and (35). The circles represent the points about which the expansions are constructed.

using a numerical method together with the procedure outlined above. In addition, the second derivative $d^2k/d\omega^2$ gives an *error estimate* for the effective refractive index as a function of the frequency. This is very useful for researchers who would like to know the range of validity of their effective medium model.

It is worth noting that Algorithm 3 can also be applied to any real parameter of the problem. For example, if one wants to compute the dispersion surface, one must compute ω for each value of the Bloch vector $\mathbf{k}_{\text{bloch}} = (k_x, k_y)$. The procedure outlined above gives a simple way of computing the derivatives $\partial\omega/\partial k_x$, $\partial\omega/\partial k_y$, $\partial^2\omega/(\partial k_x \partial k_y), \ldots$ The dependance of ω at critical points (say, at the edge of the band-gap) or on other variables such as the refractive index and the radius can also be easily computed. In each case the calculation reduces to a set of back-solves into systems such as (30) which have already been factorized by the root solving procedure.

5. Conclusion

We have described an approach to the solution of nonlinear eigenvalue problems which arise in the computation of photonic band structures in two-dimensional periodic materials such as photonic crystals. In doing so, we have examined various methods for the solution of nonlinear eigenvalue problems and applied them to this physical situation. One of these methods, the so-named implicit determinant algorithm, we have re-formulated for Hermitian systems and developed a variation which shows improvements not only in speed but also in domain of convergence. The reasons for this have been analysed. We have also shown the advantage of these methods in calculating high-order derivatives which can be used to efficiently compute dispersion curves.

Acknowledgement

C. Poulton acknowledges support from the Deutsche Forschungsgemeinshft (DFG), through the Center for Functional Nanostructures within Project A3.1.

Appendix A

In this appendix, we prove the two results on $|\det M(\omega^*)|$ and $|\det D(\omega^*)|$ that are used in the discussion on the optimal choice of $\mathbf{b}$ in the paper.

Denote the eigenvalues and orthonormalised eigenvectors of $A(\omega^*)$ by $(\mu_j,\mathbf{v}_j)$, $j = 1,\ldots,n$, so that $\mu_1 = 0$ and $\mu_j \neq 0$, for $j = 2,\ldots,n$, since zero is a simple eigenvalue of $A(\omega^*)$. Also, $\mathbf{v}_i^{\mathrm{H}}\mathbf{v}_j = \delta_{i,j}$, and the solution $\mathbf{x}^*$ obtained from solving (15) at $\omega = \omega^*$ satisfies $\mathbf{x}^* = \alpha\,\mathbf{v}_1$ with $\alpha = (\mathbf{b}^{\mathrm{H}}\mathbf{v}_1)^{-1}$. We have the following lemma:

Lemma A.1. *Given that the conditions of Lemma* 1 *hold with* $\mathbf{b}^{\mathrm{H}}\mathbf{b} = 1$ *and* $\mathbf{b}^{\mathrm{H}}\mathbf{x}^* = 1$. *Then*

(*a*) $$|\det M(\omega^*)| = |\mathbf{v}_1^{\mathrm{H}}\mathbf{b}|^2|\Pi_{j=2}^n\mu_j|. \tag{A.1}$$

(*b*) *The choice* $\mathbf{b} = \mathbf{v}_1$ *maximises* $|\det M(\omega^*)|$ *over all* $\mathbf{b}$ *satisfying* $\mathbf{b}^{\mathrm{H}}\mathbf{b} = 1$.

Proof. First expand $\mathbf{b}$ in terms of the eigenvectors of $A(\omega^*)$ to give $\mathbf{b} = \sum_{j=1}^n \beta_j\mathbf{v}_j$, with $\beta_j = \mathbf{v}_j^{\mathrm{H}}\mathbf{b}$. In particular $\beta_1 = \mathbf{v}_1^{\mathrm{H}}\mathbf{b}$. Note that $|\beta_j| \leqslant 1$ since $\sum_{j=1}^n|\beta_j|^2 = 1$. Now $A(\omega^*) = V\Lambda V^{\mathrm{H}}$, where $\Lambda = \mathrm{Diag}\{0,\mu_2,\ldots,\mu_n\}$, a real diagonal matrix, and V is the matrix of eigenvectors of $A(\omega^*)$. Thus

$$\begin{pmatrix} A(\omega^*) & \mathbf{b} \\ \mathbf{b}^{\mathrm{H}} & 0 \end{pmatrix} = \begin{pmatrix} V & \mathbf{0} \\ \mathbf{0}^{\mathrm{H}} & 1 \end{pmatrix}\begin{pmatrix} \Lambda & \boldsymbol{\beta} \\ \boldsymbol{\beta}^{\mathrm{H}} & 0 \end{pmatrix}\begin{pmatrix} V^{\mathrm{H}} & \mathbf{0} \\ \mathbf{0}^{\mathrm{H}} & 1 \end{pmatrix},$$

where $(\beta)_i = \beta_i$ and $\mathbf{b} = V\beta$. Since V is an orthogonal matrix (and so $\det V = 1$) it follows that

$$\det M(\omega^*) = \det\begin{pmatrix} \Lambda & \boldsymbol{\beta} \\ \boldsymbol{\beta}^{\mathrm{H}} & 0 \end{pmatrix} = (-1)^{n+1}|\beta_1|^2|\Pi_{j=2}^n\mu_j|$$

from which the result (A.1) follows. To prove (b) note that the determinant is maximised for $\beta_1 = 1,\beta_j = 0,j = 2,\ldots,n$ which arises for the choice $\mathbf{b} = \mathbf{v}_1$. □

An extension of this lemma is the following result.

Corollary A.1. *Let*

$$D(\omega) = \begin{pmatrix} A(\omega) & -A_\omega(\omega)\mathbf{x}(\omega) \\ \mathbf{b}^{\mathrm{H}} & 0 \end{pmatrix}.$$

Then

$$|\det D(\omega^*)| = |\mathbf{b}^{\mathrm{H}}\mathbf{v}_1|^2|\mathbf{x}^{*\mathrm{H}}A_\omega(\omega^*)\mathbf{x}(\omega^*)||\Pi_{j=2}^n\mu_j|, \tag{A.2}$$

and the choice $\mathbf{b} = \mathbf{v}_1$ *maximises* $|\det D(\omega^*)|$.

Proof. First write $-A_\omega(\omega)\mathbf{x}(\omega) = \sum_{j=1}^n\delta_j\mathbf{v}_j = V\boldsymbol{\delta}$, with $\delta_j = -\mathbf{v}_j^{\mathrm{H}}A_\omega(\omega)\mathbf{x}(\omega)$. Now, as in the proof of the above lemma, we write

$$\begin{pmatrix} D(\omega^*) & -A_\omega(\omega)\mathbf{x}(\omega) \\ \mathbf{b}^{\mathrm{H}} & 0 \end{pmatrix} = \begin{pmatrix} V & \mathbf{0} \\ \mathbf{0}^{\mathrm{H}} & 1 \end{pmatrix}\begin{pmatrix} \Lambda & \boldsymbol{\delta} \\ \boldsymbol{\beta}^{\mathrm{H}} & 0 \end{pmatrix}\begin{pmatrix} V^{\mathrm{H}} & \mathbf{0} \\ \mathbf{0}^{\mathrm{H}} & 1 \end{pmatrix},$$

and so

$$\det D(\omega^*) = (-1)^{n+2}(\mathbf{b}^{\mathrm{H}}\mathbf{v}_1)(\mathbf{v}_1^{\mathrm{H}}A_\omega(\omega^*)\mathbf{x}(\omega^*))|\Pi_{j=2}^n\mu_j|. \tag{A.3}$$

Then, with $\mathbf{v}_1 = \alpha^{-1}\mathbf{x}(\omega^*)$, we have

$$|\det D(\omega^*)| = |\alpha|^{-2}|\mathbf{x}^{*\mathrm{H}}\mathbf{b}||\mathbf{x}^{*\mathrm{H}}A'(\omega^*)\mathbf{x}^*||\Pi_{j=2}^{n}\mu_j|, \tag{A.4}$$

and since $\mathbf{b}^{\mathrm{H}}\mathbf{x}^* = 1$ and $\alpha = (\mathbf{b}^{\mathrm{H}}\mathbf{v}_1)^{-1}$ we have the result (A.2). Note that all the terms on the right-hand side of (A.2) are real, so $\det D(\omega^*)$ is real as expected. □

References

[1] A. Taflove (Ed.), Finite Difference Time Domain Methods for Electrodynamic Analyses, Artech House, New York, 1995.
[2] B.B. Hiet, J.M. Geronowicz, S.J. Cox, M. Molinari, D. Beckett, G.J. Parker, K.S. Thomas, Finite element modelling of photonic crystals, Proc. PREP 2001 (2001) 87–88.
[3] V.B. Khazanov, V.N. Kublanovskaya, Spectral problems for matrix pencils: methods and algorithms II, Sov. J. Numer. Anal. Math. Modelling 3 (1988) 467–485.
[4] K.M. Leung, Y.F. Liu, Full vector wave calculation of photonic band structures in face-centred cubic dielectric media, Phys. Rev. Lett. 65 (1990) 2646–2649.
[5] J.B. Pendry, P.M. Bell, Transfer matrix techniques for electromagnetic wavesNATO ASI Series E: Applied Sciences, vol. 315, Kluwer, Dordrecht, 1996, p. 203.
[6] R.C. McPhedran, N.A. Nicorovici, L.C. Botten, Ke-Da Bao, Green's function, lattice sum and Rayleigh's identity for a dynamic scattering problemIMA Volumes in Mathematics and its Applications, vol. 96, Springer, New York, 1997, pp. 155–186.
[7] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, Numerical Recipes in Fortran 77, section 9.3, CUP, Cambridge, MA, 1992.
[8] C. Kittel, Introduction to Solid State Physics, third ed., Wiley, New York, 1966, p. 244.
[9] K. Sakoda, Optical Properties of Photonic Crystals, Springer, Berlin, 2001.
[10] A.L. Andrew, K.-W.E. Chu, P. Lancaster, Derivatives of eigenvalues and eigenvectors of matrix functions, SIAM J. Matrix Anal. Appl. 14 (1993) 903–906.
[11] A.L. Andrew, K.-W.E. Chu, P. Lancaster, On the numerical solution of nonlinear eigenvalue problems, Computing 55 (1995) 91–111.
[12] A. Ruhe, Algorithms for the nonlienear eigenvalue problem, SIAM J. Numer. Anal. 10 (1973) 674–689.
[13] P.M. Anselone, L.B. Rall, The solution of characteristic value-vector problems by newton's method, Numer. Math. 11 (1968) 38–45.
[14] P. Lancaster, Lambda-Matrices and Vibrating Systems, Pergamon, Oxford, 1966.
[15] P. Lancaster, A generalised Rayleigh quotient iteration for lambda-matrices, Arch. Rat. Mech. Anal. 8 (1961) 309–322.
[16] A. Neumaier, Residual inverse iteration for the nonlinear eigenvalue problem, SIAM J. Numer. Anal. 20 (1983) 914–923.
[17] J.C. Maxwell-Garnett, Colours in metal glasses and in metallic films, Philos. Trans. R. Soc. Lond. 203 (1904) 385–420.
[18] J.R. Bunch, L. Kaufman, Some stable methods for calculating inertia and solving symmetric linear systems, Math. Comput. 31 (1977) 163–179.
[19] A. Griewank, G.W. Reddien, Characterisation and computation of generalised turning points, SIAM J. Numer. Anal. 21 (1984) 176–185.
[20] A. Jeffrey, Mathematics for Engineers and Scientists, Nelson, 1969.
[21] H.B. Keller, Numerical solution of bifurcation and nonlinear eigenvalue problems, in: P. Rabinowitz (Ed.), Applications of Bifurcation Theory, Academic Press, New York, 1977, pp. 359–384.
[22] R. Seydel, Practical Bifurcation and Stability Analysis: From equilibrium to Chaos, Springer, New York, 1994.
[23] A. Spence, C. Poulton, Inverse iteration for nonlinear eigenvaluen problems, in: A.B. Movchan Kluwer (Ed.), Electromagnetic Scattering in IUTAM Symposium on Asymptotics, Singularities and Homogenisation in Problems of Mechanics, 2003, pp. 585–594.
[24] G. Symm, J.H. Wilkinson, Realistic error bounds for a simple eigenvalue and its associated eigenvector, Numer. Math. 35 (1980) 113–126.