Journal of Computational Electronics 1: 411–414, 2002

Low-Rank Update Eigensolver for Supercell Band Structure Calculations

MING GU AND BERESFORD PARLETT
Department of Mathematics, University of California, Berkeley, CA 94720, USA

DAVID Z.-Y. TING
Jet Propulsion Laboratory, California Institute of Technology, CA 91109, USA

JIANLIN XIA
Department of Mathematics, University of California, Berkeley, CA 94720, USA

Abstract. In photonic or electronic and sonic band structure calculations, one often needs to solve the same eigenvalue problem many times for different sets of parameters. Often only a relatively small part of the matrix varies with these parameter changes. We have recently developed a method where, after the eigenvalue problem is solved once, the remaining cases could be computed much faster (150 times in a typical calculation). Prototype calculations using 2D photonic band structures as examples have verified this concept.

Keywords: computation, eigenvalue, band structure, low-rank update

1. Introduction

In nanodevices, we take advantage of quantum mechanical properties arising from having device feature sizes comparable to electron deBroglie wavelengths. In nanophotonics, we exploit optical properties of nanofabricated dielectric structures with artificial feature sizes on the order of the wavelengths of electromagnetic radiation of interest. Many of today's advanced optoelectronic devices incorporate both quantum device and photonic structure design principles, examples include: photonic crystal defect mode laser (Painter *et al.* 1999), quantum well infrared photodetectors (Gunapala *et al.* 2000), and photonic crystal distributed feedback lasers (Vurgaftman and Meyer 2001). The studies of nanodevices and nanophotonics often involve costly or time-consuming advanced growth, fabrication, and characterization techniques. Therefore modeling is often used to help in the device design process, and also to provide physical understanding of device properties. Typical modeling activities involve solving Schrödinger or Maxwell wave equations to compute energy or frequency spectrum, wave functions or mode profiles, as well as wave propagation characteristics. As the complexity of nanodevices and photonic structures grows, device modeling also becomes more challenging. Traditional numerical techniques with run times proportional to the cube of computational domain size are no longer sufficient, and are replaced by advanced modern numeric algorithms. These algorithms tend to be specialized, and their performance can vary greatly depending on their compatibility with the application.

In this paper we consider band structure calculations where the same eigenvalue problem needs to be solved many times for different sets of parameters, but only a relatively small part of the matrix varies with these parameter changes. We have developed a method where, after the eigenvalue problem is solved once, the remaining cases could be computed much faster (150 times in a typical calculation). The rest of the paper is organized as follows. Section 2 describes the method. Section 3 demonstrates its effectiveness in a calculation of 2D photonic band structures. Section 4 summarizes.

2. Low-Rank Update Eigensolver for Band Structure Calculations

We use photonic band structure calculations as an example in the discussion of our method. In photonic band structure calculations we find normal frequency modes as functions of wave vectors (**k**) for two- or three-dimensional periodic dielectric structures. A straightforward approach involves the application of Bloch's (Floquet's) Theorem to limited the real-space computation domain to a single unit cell, and then discretizing Maxwell equations using a finite differencing scheme. This results in a matrix representation of the problem; diagonalization of the matrix yields the normal modes. To compute band structure, one repeats the same computation for a set of, typically 100 or more, wave vectors. A key characteristic of this matrix is that only the matrix elements involving the discretized points on the surface of the unit cell are **k**-dependent. So from run to run, only a small fraction (essentially ratio between the number of dicretized surface points and number volume points) of the matrix elements are updated. This is the characteristic that we exploit in the low-rank update eigensolver.

In two spatial dimensions, the discretized Schoedinger equation can be written in the following form

$$S = A + H(q),$$

where A is a real symmetric matrix and $H(q)$ is a low rank Hermitian matrix that contains all information regarding parameter changes. For the 64-by-64 2-D grid, A is 4096-by-4096 while $H(q)$ is a matrix of rank at most 58.

Our approach is to first compute the eigendecomposition

$$A = QDQ' \text{ to get}$$
$$S = Q(D + W)Q',$$

where $W = Q'H(q)\ Q$ is a matrix of the same rank as $H(q)$. The eigenvalues of S are now the eigenvalues of $D + W$, where D is a diagonal matrix. We have developed a specialized method to rapidly compute the eigenvalues of interest in this new and structured matrix. Our approach is based on bisection accelerated by Chebyshev approximation.

Specifically, let d be an approximation to an eigenvalue we are interested in. The Inertia of the matrix $D + W - dI$ is defined to be the triple (p, q, n) where p, q, n are the numbers of positive, zero, and negative eigenvalues of the matrix $D + W - dI$. We will talk about efficient inertia computation later. For now suppose there are t eigenvalues of $D + W$ within the interval (a, b), and a, b themselves are not eigenvalues of $D + W$, then there are exactly t more positive eigenvalues in the matrix $D + W - aI$ than in the matrix $D + W - bI$. Hence we can compute the number t without knowing the eigenvalues of $D + W$ themselves by computing the inertias of $D + W - aI$ and $D + W - bI$. Let $d = (a + b)/2$. We can further compute the numbers of eigenvalues in the intervals (a, d) and (d, b) by computing the inertia for the matrix $D + W - dI$. The bisection method computes all the eigenvalues of $D + W$ in the interval (a, b) by repeatedly dividing the interval into smaller and smaller intervals and computing the inertias at the end points. Those intervals that do not contain any eigenvalues of $D + W$ are immediately ignored. The bisection method converges when all the subintervals in (a, b) that contain eigenvalues of $D + W$ have lengths that are sufficiently small. The mid-points of these subintervals are then taken as approximate eigenvalues of $D + W$. Further discussion on the inertia and the bisection method can be found in Demmel (1997) and Golub and van Loan (1983).

To compute the inertia, we rewrite W as $W = FTF'$, where T is r-dimensional Hermitian matrix with r being the rank of W. The following theorem forms the basis of our inertia computation. For any matrix M, let In(M) be its inertia.

Theorem. $\text{In}\,(-T-1)+\text{In}(D+W-dI) = \text{In}\,(D-dI)+ \text{In}(-(T-1+F'(D-dI)-1F))$.

Hence we compute the inertia of $D + W - dI$ via the computation of the other three inertias in the above theorem. Since—$T - 1$ is independent of d, its inertia can computed once and for all d. Since D is diagonal, the inertia of $D-dI$ is just a matter of counting positive, zero, and negative diagonal entries. Hence the inertia computation for the matrix $D + W - dI$ is basically the inertia computation for $T - 1 + F'(D - dI) - 1F$, which is an r-dimensional Hermitian matrix, despite its complex look.

Since r is much smaller than the dimension of D, most of the computation for the inertia of $T - 1 + F'$ $(D-dI)-1F$ is in computing the entries of this matrix itself for any given value of d. For each d, the cost for this computation is $O(n\ r2)$ operations, where n is the dimension of D.

To further improve efficiency in this computation, we note that most of the eigenvalues of A are far away

from the eigenvalues that are of interest to us. In other words, the majority of diagonal entries in D are far away from the interval (a, b). To take advantage of this fact, we rewrite D as $D = \mathrm{diag}(D1, D2)$, where diagonal entries in $D1$ are far away from (a, b) but those of $D2$ are close, and we rewrite F accordingly as $F' = (F1'F2)$. It follows that

$$F'(D - dI) - 1F = F1'(D1 - dI) - 1F1 + F2'(D2 - dI) - 1F2. \quad (1)$$

Usually there are only around $O(r)$ terms in the second expression on the above right hand side, the rest are all in the first term. Indeed, our numerical experiments indicate that there are very few terms in $D2$ (about 20–30) and all the rest can be put in $D1$. The cost for evaluating the second term for each d is $O(r3)$ operations, whereas the cost for the first term remains $O(n\,r2)$ operations.

Since all the diagonal entries of $D1$ are away from d, the first term on the above right hand side can be approximated by a low degree matrix polynomial in d using Chebyshev approximations. This approximation can be computed in $O(n\,r2)$ operations, and this is done only once for all d. With this approximation, the cost for evaluating the both terms becomes $O(r3)$ operations for each value of d, as opposed to $O(n\,r2)$ operations. The savings are significant because r is much smaller than n and the expressions in (1) must be evaluated for a large number of d values in order for bisection method to converge.

3. Applications to Photonic Band Structure Calculations

As an example, we have applied the low-rank update technique to the band structure calculation of a 2D photonic crystal. The resulting band structure is shown in Fig. 1. Table 1 summarizes timing results on the application of our method. Grid sizes of 16×16, 32×32, and 64×64 are used in these calculations. As expected, the amount of time required for low rank updates is significantly less than the initial set up cost. Also, the benefit increases considerably with increasing grid size, as a consequence of decreasing "surface-to-volume" ratio. For the case of the 64×64 grid (order 4096), the computational cost of each of the remaining points is only 1/158 of that for the first point. This means that once the first **k** point is computed, the remaining points of the band structure can be obtained essentially for free. Furthermore, the table also shows that the cost of computing the first point is less than the cost of a single point computed using a standard Lapack eigensolver. We estimate that for a typical 100-point band structure, this results in over 250-fold performance gain over dense eigensolver routines from the standard Lapack library. To be fair, there are available methods that are faster than the standard Lapack dense matrix solvers (e.g., iterative solvers). On the other hand, we are also using a dense matrix eigendecomposition routine in the initial step of this prototype low-rank update

Table 1. Timing comparison between Low-rank update method and Lapack eigensolver.

	Low-rank update method		
Matrix size	Time for computing the first case (sec.)	Time for computing each subsequent case (sec.)	Lapack Eigensolver time per case (sec.)
256	1.5	0.48	0.4
1024	39.5	3.2	55.6
4096	3664.8	23.2	15241.0

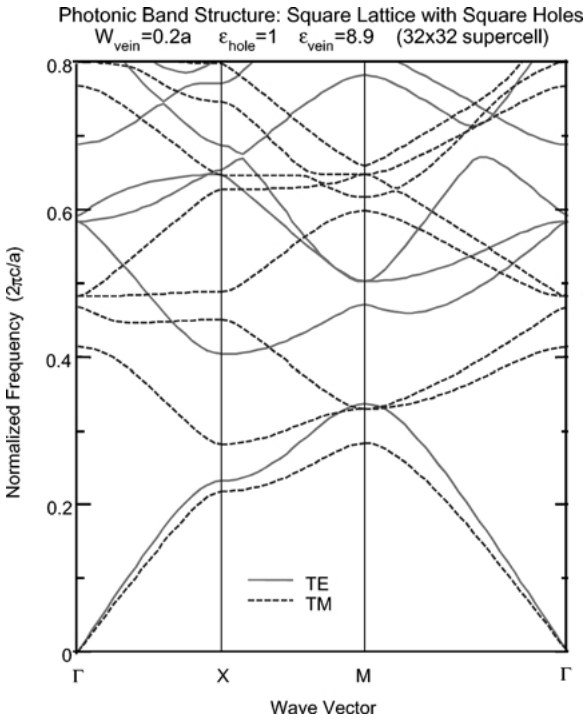

Figure 1. Calculated photonic band structure showing frequency vs. wave vectors.

eigensolver. Here we mainly wish to illustrate the numerical advantages of the low-rank update method. We are in the process of developing improved methods where only a Jacobi-Davidson style iterative method is used to find the relevant eigenvalue information for the initial step, thereby completely eliminating the need for dense solvers.

4. Summary

We have recently developed a low-rank update method where for accelerating band structure calculations. The method relies on the fact that in band structure calculations one often needs to solve the same eigenvalue problem many times for different sets of parameters. Often only a relatively small part of the matrix varies with these parameter changes. Our techniques allows us to solve the eigenvalue problem once using a special technique, and then compute the remaining a a much lower computational cost. Prototype calculations using 2D photonic band structures as examples have verified this concept.

Acknowledgment

A part of the work described in this paper was performed by DZT at the Jet Propulsion Laboratory, California Institute of Technology, and was sponsored by the NASA Advanced Concepts and Technology Innovation Program and by the Caltech President's Fund. MG was partially supported by an NSF career award CCR-9702866 and by Alfred Sloan Research Fellowship BR-3720.

References

Demmel J. 1997. Applied Numerical Linear Algebra. SIAM, Philadelphia, PA.

Golub G. and van Loan C. 1983. Matrix Computations. Johns Hopkins University Press, Baltimore, MD.

Gunapala S.D., Bandara S.V., Liu J.K., Luong E.M., Rafol S.B., Mumolo J.M., Ting D.Z., Bock J.J., Ressler M.E., Werner M.W., LeVan P.D., Chehayeb R., Kukkonen C.A., Levy M., LeVan P., and Fauci M.A. 2000. Sensor Mater. 12: 327.

Painter O.J., Husain A., Scherer A., O'Brien J.D., Kim I., and Dapkus P.D. 1999. J. Lightwave Technol. 17: 2082.

Vurgaftman I. and Meyer J.R. 2001. Appl. Phys. Lett. 78: 1475.