J Glob Optim (2007) 39:235–245
DOI 10.1007/s10898-007-9135-3

ORIGINAL PAPER

A projection based multiscale optimization method for eigenvalue problems

Peter A. Graf · Wesley B. Jones

Received: 6 January 2006 / Accepted: 4 January 2007 / Published online: 12 April 2007

Abstract We present a projection based multiscale optimization method for eigenvalue problems. In multiscale optimization, optimization steps using approximations at a coarse scale alternate with corrections by occasional calculations at a finer scale. We study an example in the context of electronic structure optimization. Theoretical analysis and numerical experiments provide estimates of the expected efficiency and guidelines for parameter selection.

Keywords Reduced-order modeling · Approximation optimization · Multiscale optimization · Galerkin projection

1 Introduction

Casting design in science and engineering as global optimization presents a tantalizing prospect; we can formulate designs as optimizations, we have the tools to perform the computations, but the computational complexity makes the solution of the optimization problems intractable. Our work is a step toward solution of this problem. To solve large global optimization problems involving expensive objective functions and large search spaces, we apply the following general method: Given a potential solution, derive a reduced-order approximation of the objective function at a larger scale, or, equivalently, in a smaller space. Using this approximation, perform some number of optimization steps in the computationally more tractable setting of the larger scale/smaller space. Now calculate the objective function of the improved solution once more with high accuracy, enabling a new approximation to be derived that is better in the region of the current best solution. Repeat this process until satisfied. This method is a marriage of two techniques, reduced-order modeling and global

P. A. Graf (✉) · W. B. Jones
Scientific Computing Center, National Renewable Energy Laboratory, 1617 Cole Blvd mailstop 1608, Golden, CO 80401, USA
e-mail: peter_graf@nrel.gov

optimization. Alone, they fail in the face of large problems. Together, each compensates for the shortcomings of the other.

Uses of approximation in optimization are not uncommon within the context of traditional mathematical programming [19]. Examples with a true multiscale flavor—including approximations and ongoing correction of approximations—are more rare, but include: use of proper orthogonal decomposition to derive reduced-order models in flow control problems [15]; "sequential reduced system programming" in the context of global climate models [9]; multigrid based optimization schemes for nonlinear programming [8] and VLSI layout [14]; refinement of approximations during search based on balance between quality of approximation and success of optimization [18]; optimization of model reduction for ordinary differential equations [7].

The most important concept in this paper is the *repeated* regeneration and use of approximations as an optimization progresses. The most concrete contribution is the development of the technical machinery necessary to implement this concept for optimization problems whose objective functions involve eigenvalue problems.

In Sect. 2 we describe our projection based multiscale optimization method. In Sect. 3 we describe a physical problem of interest and the application of our method to it. In Sect. 4 we present analysis and numerical studies aimed at understanding and tuning the method for maximum effectiveness.

2 Method

Here we describe the projection based optimization method for eigenvalue problems. We begin by stating the type of problem to which we apply our method. We then present the important notions of choosing and projecting onto a subspace. Next we describe the search method employed. We then present the overall projection based multiscale algorithm.

We consider problems of the form

$$\max_{\xi \in \mathcal{D}} F(\xi),$$

where evaluation of $F(\xi)$ involves solution of an eigenvalue problem

$$A(\xi)v = \lambda v,$$

where $v \in \mathcal{R}^N$, and A is an N-by-N matrix. The search domain $\mathcal{D}$ we deliberately leave rather abstract; our method works as well for constrained as for unconstrained problems, and there is no assumption that ξ is a vector in any particular space (in our application below, it is a binary string). We assume that $A = A(\xi)$ depends on the free variables ξ and that F is, for example, a function $F(\{v\}, \{\lambda\})$ of the eigenvalues and eigenvectors of A. The dependence of A (and thus F) on ξ may be highly indirect. For example, in Sect. 3 an atomic configuration ξ determines the potential energy V, which contributes to the Hamiltonian H, whose discretization is A.

For large N, the eigenvalue problem is an expensive calculation we wish to perform as infrequently as possible. Therefore, we will apply the following approach: Given ξ, we solve $A(\xi)v = \lambda v$ with high accuracy (throughout, "high accuracy" will refer to solving an equation in $\mathcal{R}^N$). From the results of this calculation, given $M < N$, we choose an M dimensional subspace $P \subset \mathcal{R}^N$ upon which we will "project" our eigenvalue problem. The subspace P is meant to be a good approximation, in the

neighborhood of ξ, of the part of the spectrum of $A(\xi)$ from which F is computed. That is, using a calculation only at ξ, we choose P such that for all ξ' *near* ξ, the eigenvectors (and thus the eigenvalues) of interest of $A(\xi')$ are well approximated by vectors in the subspace P.

The choice of P is problem specific; the important part of the spectrum for computing F obviously depends on F. For definiteness, in the case considered in this paper, P is chosen as follows: For a given ξ, find the M eigenvectors v of $A(\xi)$ whose eigenvalues are closest to a given reference eigenvalue λ_{ref} that defines the part of the eigenvalue spectrum in which we are interested (for us this will be the center of the "band gap" in the spectrum; see Sect. 3). Let P be the N-by-M matrix whose columns are those M eigenvectors. Such a P always exists in our case because the matrix A is symmetric (see below) so there exists a basis of eigenvectors. (Construction of P in the case that A is not diagonalizable will not be considered here.) We use "P" to refer to both the matrix and the subspace it represents (the columns of the matrix form a basis of the subspace), and we may write $P(\xi)$ to emphasize the dependence of P on ξ.

By evaluating the objective function $F(\xi)$ "in the space P," we mean the following: Instead of evaluating $F(\xi)$, we evaluate an approximate function $\hat{F}$ given by the approximate eigenvalues as calculated in P. To do so, we form the *Galerkin projection* [5] onto P, defined as follows. Let the M-vector w be the coordinates with respect to P of a solution of the projected eigenproblem, and let its corresponding eigenvalue be μ. This vector can be written Pw. The Galerkin projection onto P of the eigenvalue problem $Av = \lambda v$ is defined by the condition that the residual $A(Pw) - \mu(Pw)$ is orthogonal to P. That is,

$$P^T(APw - \mu Pw) = 0.$$

Because the columns of P are orthonormal, this becomes

$$P^T APw = \mu w.$$

Thus the approximate matrix whose eigenvalues we will use to calculate F is given by

$$\hat{A}(\xi) = P^T A(\xi) P.$$

$\hat{A}(\xi)$ is, importantly, an M-by-M matrix. Thus we can compute the eigenvalues of $\hat{A}$ much more quickly than those of A. $\hat{F}$ is defined to be the same function of the eigenvectors and eigenvalues of $\hat{A}$ as F is of A. That is, to evaluate $\hat{F}(\xi)$, we solve

$$\hat{A}w = \mu w,$$

and let $\hat{F}(\xi) = F(\{Pw\}, \{\mu\})$. We present an explicit example in Sect. 3.

The subspace size M is a critical parameter to the method. If it is too near N, we do not gain much efficiency. If it is too small, $\hat{F}$ is not close enough to F to perform effective optimization. This issue is explored in Sect. 4.

We will couple this notion of Galerkin projection with a global optimization method. Though we could use a variety of methods, we will use simulated annealing [4] here. This iterative method consists of a search in which starting from some ξ_0, ξ_{k+1} is derived from ξ_k as follows: a test solution ξ_{k+1} is formed from ξ_k in a problem specific way (see below, for example, where the new test solution is formed from the current one by randomly swapping two atoms), and the new solution is accepted with probability

1: Initialize ξ_0, λ_{ref}, N, M (s.t. $M < N$), and m.
2: **for** $k = 0$ to k_{max} **do**
3: Solve $A(\xi_k)v = \lambda v$.
4: Let P equal the M v's with corresponding eigenvalues closest to λ_{ref}.
5: Let $\xi_k^0 = \xi_k$.
6: **for** $j = 0$ to m **do**
7: Let $\hat{A}(\xi_k^j) = P^T A(\xi_k^j)P$.
8: Solve $\hat{A}(\xi_k^j)w = \mu w$.
9: Use the result to calculate $\hat{F}(\xi_k^j)$.
10: Using $\hat{F}(\xi_k^j)$, take search step to generate ξ_k^{j+1} from ξ_k^j.
11: **end for**
12: Let $\xi_{k+1} = \xi_k^m$.
13: **end for**

Fig. 1 The projection based multiscale eigenvalue optimization algorithm. Note the structure: an inner "optimization" loop (lines 6 through 11) nested within an outer "subspace refinement" loop (lines 2 through 13)

$$\begin{cases} 1 & \text{if } F(\xi_{k+1}) > F(\xi_k), \\ e^{-(F(\xi_k)-F(\xi_{k+1}))/T} & \text{otherwise.} \end{cases}$$

T, the "temperature," is a parameter that is slowly "cooled" to zero during the course of an optimization. The value of T determines the balance between global and local search.

With these ingredients in hand, we can now state our algorithm. Projection based multiscale optimization for eigenvalue problems consists of two levels of iteration, an inner "optimization" loop contained within an outer "subspace refinement" loop, and proceeds as follows: Given the parameters N, M, m, and an initial ξ_0, we first form P as described above. Now, for $k = 0, 1, \ldots$, we let $\xi_k^0 = \xi_k$ and perform m optimization steps using the objective function $\hat{F}$, the projection of F onto P, searching through configurations ξ_k^j, for $1 \leq j \leq m$, using simulated annealing. This is the inner "optimization" loop. As ξ_k^j diverges from ξ_k^0, the quality of our approximation degrades. Therefore, after m inner iterations, we let $\xi_{k+1} = \xi_k^m$, then reform P from ξ_{k+1} by solving $A(\xi_{k+1})v = \lambda v$ with high accuracy. This forms the outer "subspace refinement" loop. We repeat both loops until satisfied with our solution.

Note that it is the *repeated* recalculation of the subspace P that gives the method its power. Approximation in optimization can be useful [17,19], but its effectiveness is limited by the tendency of approximations to only be valid in a small part of the search domain. Here we explicitly address this limitation by continually adjusting the space P, thus the approximation $\hat{F}$, to reflect the current status of the search.

Figure 1 illustrates the algorithm. In the next section we present a physical example. In Sect. 4 we study the choice of m and M.

3 Example

Eigenvalue optimization problems are at the heart of attempts to design semiconductor materials with specific electronic properties. In particular, we now explore our method within the context of an extended Kronig–Penney model [12]. This test application is a first step toward problems of practical interest.

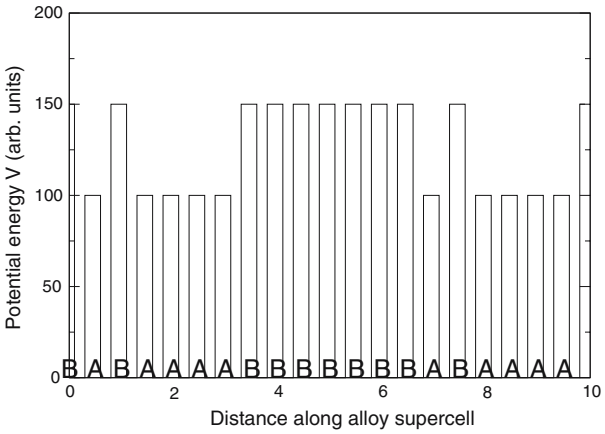

Fig. 2 The atomic identities ("A" or "B") and potential along the one dimensional model lattice for a particular atomic configuration σ

Recent advances in the field of electronic structure calculation [1,10,20] have led to attempts to automatically design materials with desired properties by casting the entire process as a global optimization problem [2,6,11]. In such a formulation we solve a problem of the form

$$\max F(\sigma),$$

where the free variable σ represents a configuration of atoms and $F(\sigma)$ is the electronic property of interest such as semiconductor band gap. Although successful in limited settings, this approach founders on the complexity of the computations as the number of atoms in σ grows. Our method is targeted toward application to such problems.

Consider a one-dimensional lattice of atoms of type "A" and "B" represented by a vector σ. The electronic energy eigenstates of a configuration σ are found by solving the Schrödinger equation

$$H\psi = \epsilon\psi,$$

where

$$H = -\nabla^2 + V(x).$$

(Throughout this paper we assume all physical constants are set to unity.) $V(x)$ is the potential energy. In our model it is a square wave whose height is determined by which atom, A or B, lies nearest to x. This amounts to an extension of the classic Kronig–Penney model, in which all the atoms were of only one type. Note that H depends on σ through $V = V(\sigma)$. Figure 2 illustrates a particular configuration σ and its potential $V(\sigma)$. We assume periodic boundary conditions throughout, so the structure is assumed to repeat infinitely along the line.

To solve this equation, we discretize the domain into N grids of size h. The eigenfunction ψ becomes an N-dimensional vector, and the operator H is represented by an N-by-N matrix we will also call H. For our problem H has a particularly simple form. With periodic boundary conditions, a first-order discretization of the second derivative, and $V_i \equiv V(x_i) \equiv V(i * h)$, we may write

$$H = \begin{pmatrix} \frac{2}{h^2} + V_1 & \frac{-1}{h^2} & 0 & \cdots & \frac{-1}{h^2} \\ \frac{-1}{h^2} & \frac{2}{h^2} + V_2 & \frac{-1}{h^2} & 0 & \cdots \\ 0 & \frac{-1}{h^2} & \frac{2}{h^2} + V_3 & \frac{-1}{h^2} & \cdots \\ \cdots & & & & \\ \frac{-1}{h^2} & 0 & \cdots & \frac{-1}{h^2} & \frac{2}{h^2} + V_N, \end{pmatrix}$$

then solve the resulting eigenproblem.

The spectrum of such a matrix has *bands* of eigenvalues separated by a *band gap* [12]. The band gap of the material is derived by subtracting the lowest eigenvalue above the band gap from the highest eigenvalue below the band gap (typically we know approximately where the band gap is, so it is not difficult to identify these states). It is of practical interest to control this gap. Therefore, our prototypical design problem is to find the arrangement of A and B atoms that maximizes the band gap.

This problem is of the form presented in Sect. 2, with the discretized H as the matrix A, and F defined to be the difference between the two eigenvalues above and below the band gap. We begin with a random atomic configuration σ of a certain composition (number of "A" versus "B" atoms), and the parameters N, M, m, and ϵ_{ref}. We solve $H(\sigma)\psi = \epsilon\Psi$ as describe above, on a grid of size N, and let P be the M eigenvectors whose eigenvalues are closest to ϵ_{ref}. We now take m search steps "in P," that is, by approximating the band gap F computed from H by the approximate band gap $\hat{F}$ computed from $\hat{H} = P^T HP$. After these m steps, the resulting new configuration σ is used to generate a new P, and we repeat the process as long as desired.

Figure 3 shows a comparison between optimizations for a 20 atom problem with and without multiscale approximation. The figure shows the mean optimization paths for a set of five runs, with five different random seeds. The average time to reach the maximum for $M = 30$ was 0.966 s. Without approximation, it was 4.512 s, a factor of five efficiency gain.

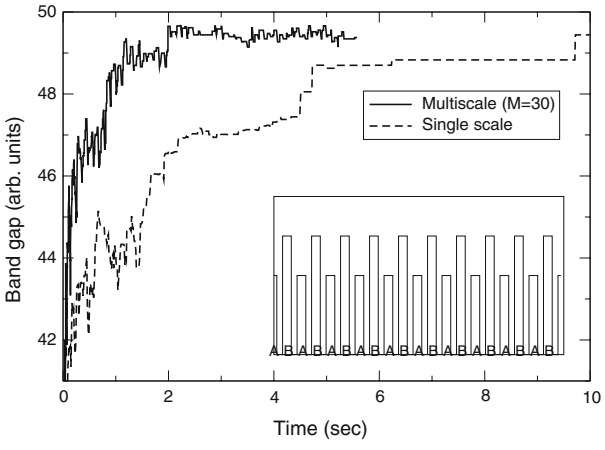

Fig. 3 Results and comparison of optimization of the extended Kronig–Penney model with and without approximation. The band gap units are arbitrary. The results are for a 20-atom system. $N = 100$, $M = 30$, and $m = 20$. The observed time reduction factor is 0.20. Our underlying solver has computational complexity $O(N^3)$, so the lower bound on the time reduction is approximately $(N^3 + MN^2 + m(M^3 + M^2N))/(mN^3) = 0.18$, as discussed in Sect. 4.1. The inset shows the optimal configuration and its potential

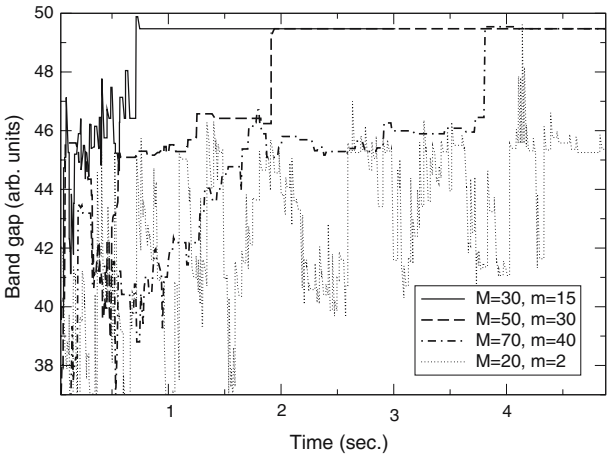

Fig. 4 Comparison of different choices of the size M of the subspace P. For each value we have plotted the mean path of 5 runs from different initial conditions for the best value of m obtained from a scan of values from $m = 1$–50

As a first step toward scaling the algorithm to realistic problems, we have run the one-dimensional model on problems with up to 80 atoms. In all cases the multiscale algorithm increases the efficiency of the search by roughly a factor of two or more.

To apply this method to a problem of practical interest will require specialized implementation of the ingredients of the algorithm described in Sect. 2. For efficiency, an electronic structure code able to compute only the M eigenvalues in a certain "energy window" near the reference energy should be utilized; such codes exist [1]. This will allow us to efficienty compute the subspace P we use for approximation. To implement the projection $\hat{H} = P^T HP$ of the Hamiltonian, however, will require customization of the electronic structure program. In particular, such codes do not store H as a matrix but simply implement the matrix-vector product $H\psi$. We will have to reimplement this part of the code to instead compute $P^T HP\psi$. Such work is beyond the scope of the current paper.

4 Analysis

To begin our study of the method, consider Fig. 4. Here we have run optimizations from five random seeds for a variety of values of subspace size M and inner optimization iterations m. For each M, the plot shows the band gap's mean path (over the five seeds) for the value of m with the best performance. As expected, we observe increased efficiency as M decreases until a point where the subspace is too small for our approximations to be accurate enough to perform effective search. We pursue this situation in more detail for the rest of this section.

Analysis of global optimization methods is inherently problematic, but in our case it is made more difficult by the balance between the efficiency gain due to approximate (hence faster) objective function evaluation and the fact that error due to approximation can lead to search steps "in the wrong direction." We consider each of these issues in turn.

4.1 Complexity

Assume that we are using an eigensolver such as QR-iteration that has computational complexity of $O(N^3)$ [3,16]. (Our example, of course, has a very special form that would admit of specialized, more efficient solution, but we are interested in the general case.) In this case, the computational cost of forming the subspace of eigenvectors P is $O(N^3)$. Naively, the cost of forming the approximation $\hat{H}(\sigma) = P^T H P$ for a subsequent σ is $O(M^2N + MN^2)$. However, we can do better by separating the generation of the projection of the momentum term of H, i.e., the discretization of ∇^2, which does not depend on σ, from that of the potential term, which *does* depend on σ. In this case, we incur an additional $O(MN^2)$ cost when we generate P, but subsequent computation of $\hat{H}(\sigma)$ only requires $O(M^2N)$ operations. The solution of the M-dimensional eigenproblem for $\hat{H}$ is $O(M^3)$. So one total outer iteration, which includes generation of P and m optimization steps, requires

$$O(N^3) + O(MN^2) + m[O(M^3) + O(M^2N)],$$

operations.

Now, if we do not approximate, the same number of optimization steps requires $mO(N^3)$ operations. So the relative efficiency, per optimization step, is

$$\frac{N^3 + MN^2 + m(M^3 + M^2N)}{mN^3}.$$

If m is large, this reduces to $\sim mM^2N/mN^3 = M^2/N^2$. For our runs, typical numbers are $N = 100$, $M = 30$, $m = 20$, which results in operation count ratios of ~ 0.18. This number represents a lower bound on the time reduction attributable to the multiscale approach. We can only expect to approach this bound if our approximation is good enough that we do not require extra search steps. We now consider this issue.

4.2 Optimal choice of parameters

The cost per iteration is only one part of the total cost of the search. The other ingredient is the total number of iterations necessary. We expect that as the cost per iteration goes down (by decreasing M), the total number of iterations goes up (because of increasing error due to approximation). What is the optimal tradeoff?

Because we have no way of directly calculating how many search steps are necessary, we reason as follows: The decisions to accept or reject a move are based on comparisons between two values that may be in error. The larger this error, the more likely it is that our steps are incorrect. Thus we assume our question of how many steps are necessary, or, more precisely, the change in the number of steps with and without approximation, can be reduced to the question of objective function error as a function of different choices of subspace size M and inner iterations m.

Let h be the "Hamming distance" between two configurations:

$$\begin{aligned} h &\equiv ||\sigma_2 - \sigma_1||_H \\ &\equiv \text{the number of atomic sites at which } \sigma_1 \text{ and } \sigma_2 \text{ differ.} \end{aligned}$$

Consider Fig. 5, where we show the dependence of the band gap error $|F(\sigma_2) - \hat{F}(\sigma_2)|$ on M and h. Note that $\hat{F}(\sigma_2)$ is the approximation computed in a space $P = P(\sigma_1)$ for some $\sigma_1 \neq \sigma_2$. This graph has an intriguing shape. First, it is highly nonlinear; we have

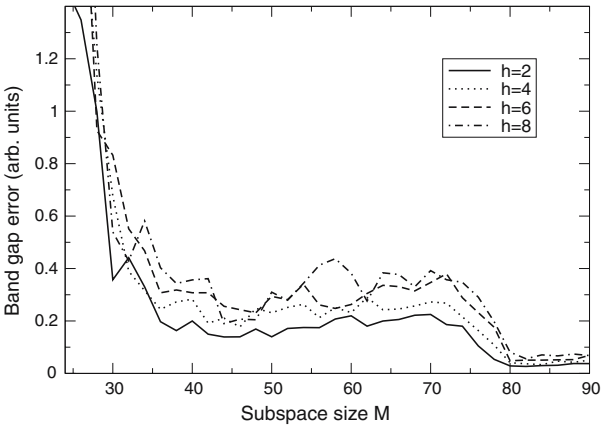

Fig. 5 Eigenvalue error versus degree of approximation (size of subspace M) and Hamming distance h from solution that generated the subspace in which we are computing. The performance of the algorithm (see Fig. 4) is highly correlated with this error

no theoretical explanation for it. Second, and more practically, it has a plateau of relatively low error (the actual band gaps are ~ 40) for a wide range of M, roughly $30 < M < 70$. And there is some, but weaker, dependence on h, which is a reflection of m. We may interpret these data as suggesting that for $30 < M < 70$, the optimization steps we take are equally effective. Thus we should use an M at the low end of this range.

Furthermore, the relatively insignificant dependence of approximation error on m suggests we should use a large value; in this way we increase the ratio of inexpensive inner iterations to expensive outer iterations with little cost in accuracy. These data support the intuition based on both the basic principle of the method, and on Fig. 4, where we find $M = 30$ to be optimal, and a wide range of rather large values of m to be the best.

That is, in Fig. 4, we observe a point near $M = 30$ below which reliability of the search plummets. This is the same point below which the band gap error is seen to increase dramatically in Fig. 5. This suggests a strong correlation between eigenvalue error and the success of the search, supporting the above assumption that the number of search steps was roughly proportional to the error. Furthermore, within the range of M in which the band gap error is small, we can assume our search steps are reliable; thus in this range the complexity estimates in Sect. 4.1 should accurately assess the expected efficiency gain of the method. This is indeed the case, as both the theoretical complexity estimate and actual data give an approximately 5-fold reduction in search time.

To suggest analytically where these results come from, consider the configurations σ_1, with corresponding Hamiltonian H_1, from which we have generated our subspace P_1, and σ_2, with Hamiltonian H_2 and subspace P_2 we would generate if we solved H_2 with high accuracy. Assume it is σ_2 whose approximate eigenvalues we are now computing by projecting onto P_1. If we were to project perfectly onto the eigenvectors corresponding to the "first" (in sense of proximity to ϵ_{ref}) eigenvalues of H_2, we would solve for $H_2^{(2)} = P_2^T H_2 P_2$. Instead we solve for $H_2^{(1)} = P_1^T H_2 P_1$. $H_2^{(2)}$ is a diagonal matrix D_2 of the closest eigenvalues of H_2 to ϵ_{ref}. Similarly, $H_1^{(1)}$ is a diagonal matrix D_1 of the closest eigenvalues of H_1 to ϵ_{ref}. Now, consider H_2 as a perturbation of H_1. The discretization of the momentum operator does not change. And the potential term V, which is always *diagonal* in our real-space discretization, changes by some amount $dV(\sigma_1, \sigma_2)$ only to reflect some number of atom swaps. Then what we compute is (for this equation let $P \equiv P_1$, and let us address the elements of the matrix P as P_{ij})

$$\begin{aligned} P^T H_2 P &= P^T H_1 P + P^T H_2 P - P^T H_1 P \\ &= D_1 + P^T (H_2 - H_1) P \\ &= D_1 + P^T dV P \\ &= D_1 + \left[\sum_l P_{il} P_{jl} dV_{ll} \right], \end{aligned} \tag{1}$$

where we have used the fact that dV is diagonal, and the notation "$[a_{ij}]$" means the matrix whose i,jth element is a_{ij}. This equation allows us to see how M and m affect what we compute. The Hamming distance, which is a function of m, enters only dV, whereas the size of the subspace M enters only in P. As $M \to N$, P becomes a change of coordinates in $\mathcal{R}^N$. As $h \to 0$, $dV \to 0$. In either limit our projected computation becomes equal to the highly accurate one. On the other hand, suppose $M = 1$. In this case $P^T H_2 P$ is an estimate of a single eigenvalue due to the perturbation dV, and Eq. (1) is just the result of first-order time-independent perturbation theory of quantum mechanics [13].

Using the results in [16], estimates of the eigenvalue error in terms of the norms of P_1, P_2, and H_2 may be established, but they are far from sharp, so we will not pursue them here. It is an open problem to discover sharp bounds on the eigenvalue error of a projection based on the subspace size.

Note, though, that even in the absence of sharp theoretical error estimates, the selection of the critical subspace size parameter M can be automated. We can simply perform sample calculations for a variety of M to find the point at which the approximation error grows large. Now choose M just above this threshold. This procedure is not computationally prohibitive because the algorithm already involves occasional highly accurate calculations, so a few more calculations to start the run will not dominate the overall cost. Also, this procedure will be effective, as the evidence suggests that the threshold M is relatively independent of the configuration σ used to generate P, thus independent of the σ we could use to establish M.

Interestingly, it may be possible that we can tolerate considerable error within a simulated annealing optimization, because such error acts like raising the temperature (it increases the likelihood of a new solution being accepted even though its objective function value is worse than the previous solution). This suggests that as the optimization progresses we could adjust M just as we adjust T. Such considerations, however, are beyond the scope of the current paper.

5 Conclusion

We have presented the multiscale optimization concept in which optimization steps in a small space are alternated with approximation refinement steps in a larger space. We have explained in detail a novel algorithm for the application of this concept to eigenvalue optimizations. The technique has been illustrated for an extended Kronig–Penney model in which we solve the problem of finding the configuration of atoms that maximizes the electronic band gap. The utility of the method has been demonstrated by comparison with optimization in the absence of approximation. The crucial parameters of subspace size M and number of optimization steps per refinement step m have been studied. We discover that search performance is highly correlated with band gap error. In the regime of small error—hence reliable search steps—we see steady improvement as M decreases. Below a certain threshold M where the error

due to approximation increases precipitously we observe the concomitant failure of the search. Potential applications of this method to computational materials science abound. In particular, we believe a multiscale approach will be an essential ingredient in the complex problem of optimizing structures for nanoscale applications.

Acknowledgements We gratefully acknowledge discussions with K. Kim and J. Langou. This work used computing resources at the NREL Computational Sciences Center and is supported by the US DOE-SC-ASCR-Mathematical, Information and Computational Sciences program under Lab03-17 Theory Modeling in Nanoscience initiative under Contract No. DE-AC36-99GO10337.

References

1. Canning, A., Wang, L.-W., Williamson, A., Zunger, A.: Parallel empirical pseudopotential electronic structure calculations for million atom systems. J. Comput. Phys. **160**, 29–41 (2000)
2. Cwik, T., Klimeck, G.: Genetically engineered microelectronic infrared filters. In: Proceedings of the First NASA/DoD Workshop on Evolvable Hardware, IEEE, pp. 242–246 (1999)
3. Demmel, J.: Applied Numerical Linear Algebra. SIAM, Philadelphia (1997)
4. Eiben, A.E., Smith, J.E.: Introduct Evol Comput. Springer, Berlin (2003)
5. Eriksson, K., Estep, D., Hansbo, P., Johnson, C.: Computational Differential Equations. Cambridge University Press, Cambridge (1996)
6. Franceschetti, A., Zunger, A.: The inverse band-structure problem of finding an atomic configuration with given electronic properties. Nature **402**, 60–63 (1999)
7. Graf, P.: Optimization of model reduction for linear ordinary differential equations. PhD Thesis, University of California at Berkeley, Department of Mathematics (2003)
8. Hager W.: Minimizing a quadratic over a sphere. SIAM J. Optimiz. **12**, 188–208 (2001)
9. Janssen, M.: Sequential optimization of integrated climate change models. Math. Comput. Simulat. **54**, 477–489 (2001)
10. Kim J.: Materials Computation Center. University of Illinois (2003)
11. Kim, K., Graf, P., Jones, W.: A genetic algorithm based inverse band structure method for semiconductor alloys. J. Comput. Phys. **208**, 735 (2005)
12. Kittel, C.: Introduction to Solid State Physics. Wiley, New York (1996)
13. Landshoff, P., Metherell, A., Rees, G.: Essential Quantum Mechanics. Cambridge University Press, Cambridge (1997)
14. Lewis, R., Nash, S.G. : Practical aspects of multiscale optimization methods for VLSI-CAD. In: Cong, J., Shinnerl, J.R. (eds.) Multiscale Optimization and VLSI/CAD, pp. 265–291. Kluwer, Boston (2002)
15. Ravindran, S.S.: Adaptive reduced-order controllers for a thermal flow system using proper orthogonal decomposition. SIAM J. Sci. Comput. **23**(6), 1924–1942 (2002)
16. Saad, Y.: Numerical Methods for Large Eigenvalue Problems. Manchester University Press, Manchester (1992)
17. Siah, E.S., Sasena, M., Volakis, J.L., Papalambros, P.Y., Wiese R.W.: Fast parameter optimization of large-scale electromagnetic objects using DIRECT with kriging metamodeling. IEEE Trans. Microwave Theory Tech. **52**(1) (2004)
18. Torczon V., Trosset, M.W.: Using approximations to accelerate engineering design optimizations. In: Proceedings of the 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, St. Louis, Missouri (1998)
19. Vazirani, V.V.: Approximation Algorithms. Springer, Berlin (2003)
20. Wang, L.-W., Zunger, A.: Solving Schrödinger's equation around a desired energy: application to silicon quantum dots. J. Chem. Phys. **100**, 2394 (1994)