SIAM J. Sci. Comput.
Vol. 36, No. 6, pp. A2929–A2951

POLE EXPANSION FOR SOLVING A TYPE OF PARAMETRIZED LINEAR SYSTEMS IN ELECTRONIC STRUCTURE CALCULATIONS[*]

ANIL DAMLE[†], LIN LIN[‡], AND LEXING YING[§]

Abstract. We present a new method for solving parametrized linear systems. Under certain assumptions on the parametrization, solutions to the linear systems for all parameters can be accurately approximated by linear combinations of solutions to linear systems for a small set of fixed parameters. Combined with either direct solvers or preconditioned iterative solvers for each linear system with a fixed parameter, the method is particularly suitable for situations when solutions to a large number of distinct parameters or a large number of right-hand sides are required. The method is also simple to parallelize. We demonstrate the applicability of the method to the calculation of the response functions in electronic structure theory. We demonstrate the numerical performance of the method using a benzene molecule and a DNA molecule.

Key words. pole expansion, approximation theory, parametrized linear systems, electronic structure calculation

AMS subject classifications. 65F30, 65D30, 65Z05

DOI. 10.1137/130944825

1. Introduction.

Consider the linear system

$$(H - zS)u = b, \tag{1.1}$$

where H and S are $N \times N$ Hermitian matrices, S is positive definite, u and b are vectors of length N, and $z \in \mathbb{C}$ with $\operatorname{Re} z \le 0$.

Often (1.1) needs to be solved a large number of times, both for a large number of distinct parameters z and for multiple right-hand sides b. This type of calculation arises in a number of applications, such as time-independent density functional perturbation theory (DFPT) [5, 18, 4], many body perturbation theory using the GW method [20, 2, 11, 39, 34, 16], and the random phase approximation (RPA) of the electron correlation energy [25, 26, 14, 31]. The connection between (1.1) and these applications will be illustrated in section 5.

[*] Submitted to the journal's Methods and Algorithms for Scientific Computing section November 11, 2013; accepted for publication (in revised form) September 2, 2014; published electronically December 18, 2014.
http://www.siam.org/journals/sisc/36-6/94482.html

[†] Institute for Computational and Mathematical Engineering, Stanford University, Stanford, CA 94305 (damle@stanford.edu). This author's work was supported by NSF fellowship DGE-1147470 and was partially supported by the National Science Foundation grant DMS-0846501.

[‡] Department of Mathematics, University of California Berkeley and Computational Research Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720 (linlin@math.berkeley.edu). This author's work was partially supported by the Laboratory Directed Research and Development Program of Lawrence Berkeley National Laboratory under the U.S. Department of Energy contract DE-AC02-05CH11231 and by the Scientific Discovery through Advanced Computing (SciDAC) program funded by U.S. Department of Energy, Office of Science, Advanced Scientific Computing Research and Basic Energy Sciences.

[§] Department of Mathematics and Institute for Computational and Mathematical Engineering, Stanford University, Stanford, CA 94305 (lexing@math.stanford.edu). This author's work was partially supported by National Science Foundation under award DMS-0846501 and by the Mathematical Multifaceted Integrated Capability Centers (MMICCs) effort within the Applied Mathematics activity of the U.S. Department of Energy's Advanced Scientific Computing Research program, under award DE-SC0009409.

1.1. Previous work. From an algorithmic point of view, solving linear systems with multiple shifts has been widely explored in the literature [12, 13, 37, 6, 30, 15, 3, 9, 36]. These methods are often based on the Lanczos method. The basic idea here is that when $S = I$, the Krylov subspace constructed in the Lanczos procedure is invariant with respect to the shifts z. Therefore, the same set of Lanczos vectors can be used simultaneously to solve (1.1) with multiple z. When S is not the identity matrix, S should be factorized, such as by the Cholesky factorization, and triangular solves with the Cholesky factors must be computed during every iteration, which can significantly increase the computational cost.

In its simplest form, the Lanczos method as described in section 2 precludes the use of a preconditioner. However, there are strategies, some of which are discussed in the aforementioned references, for using a small number of preconditioners and perhaps multiple Krylov spaces to find solutions for all of the desired parameters. Even when using preconditioners, it may be unclear how to pick which Krylov spaces to use, or what the shifts for the preconditioners should be; see, e.g., [30]. Another specific class of methods are the so-called recycled Krylov methods; see, e.g., [33]. In such methods, the work done to solve one problem, potentially with a preconditioner, is reused to try and accelerate the convergence of related problems.

Another class of popular methods, especially in the context of model reduction, is based on the Padé approximation, which does not target the computation of all the entries of $u(z) = (H - zS)^{-1}b$ as in (1.1), but rather a linear functional of $u(z)$ in the form of $f(z) = l^T u(z)$. $f(z)$ is a scalar function of z, which can be stably expanded in the Padé approximation via the Lanczos procedure [9]. However, it is not obvious how to obtain a Padé approximation for all entries of $u(z)$ directly, and with an approximation that has a uniformly bounded error for all $\operatorname{Re} z \leq 0$.

1.2. Contribution. In this paper we develop a new approach for solving parametrized linear systems using a pole expansion. The pole expansion used here is a rational approximation to all entries of $u(z)$ simultaneously. The main idea behind this pole expansion comes directly from the work of Hale, Higham, and Trefethen [19], which finds a nearly optimal quadrature rule of the contour integral representation for $u(z)$ in the complex plane. The idea of the work in [19] has also been adapted in the context of approximating the spectral projection operator and the Fermi–Dirac operator (a "smeared" spectral projection operator) for ground state electronic structure calculation [27].

Such a scheme directly expresses $u(z)$, and thus the solution to (1.1) for any z, as the linear combination of solutions to a small set of linear systems, each of which has a fixed parameter. Each linear system with a fixed parameter may be solved either with a preconditioned iterative method or a direct method. Furthermore, the construction of the pole expansion depends only on the largest and smallest positive generalized eigenvalues of the matrix pencil and the number of poles. The number of poles used in the expansion yields control over the accuracy of the scheme. In fact, the pole expansion converges exponentially with respect to the number of approximating terms across the range $\operatorname{Re} z \leq 0$. Finally, the pole expansion allows the same treatment for general S as in the case of $S = I$, without directly using its Cholesky factor or S^{-1} at every iteration. As a result, the pole expansion addresses the disadvantages of both the Lanczos method and the Padé approximation.

1.3. Notation. In this paper we use the following notation. Let $\{\lambda_i\}, \{\psi_i\}$ be the generalized eigenvalues and eigenvectors of the matrix pencil (H, S) which satisfy

$$H\psi_i = S\psi_i\lambda_i, \quad i = 1, \ldots, N, \tag{1.2}$$

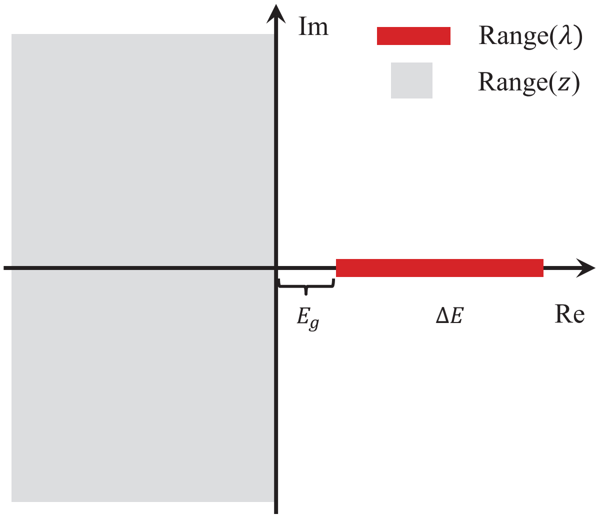


FIG. 1. *A schematic view of the range of the spectrum of the (H, S) pencil (thick red line on the positive real axis) and the range of the parameter z (light gray area), separated by a positive distance E_g.*

with $\{\lambda_i\}$ labeled in nonincreasing order. Such an eigen decomposition is only used for deriving the pole expansion and is not performed in practical calculations. We initially assume that all the eigenvalues λ_i are positive, and denote by $E_g \equiv \lambda_N > 0$. However, we later relax this assumption and allow for a small number of negative eigenvalues. We also denote by ΔE the spectrum width of the (H, S) pencil, i.e., $\Delta E = \lambda_1 - \lambda_N$. We refer to Figure 1 for an illustration of the relative position of the spectrum and the range of the parameter z in the complex plane.

Furthermore, A^* and A^T denote the conjugate transpose and the transpose of a matrix A, respectively. We let R be an upper triangular matrix that denotes the Cholesky factorization of S, i.e.,

$$S = R^* R.$$

We use the notation

$$\mathcal{K}_n(A, b) = \operatorname{span}\left(b, Ab, \ldots, A^{n-1}b\right)$$

to denote the nth Krylov subspace associated with the matrix A and the vector b. Finally, we let N_z be the number of distinct complex shifts z for which we are interested in solving (1.1).

1.4. Outline. The rest of the manuscript is organized as follows. We first discuss the standard Lanczos method for solving linear systems with multiple shifts in section 2. We introduce the pole expansion method for solving (1.1) and analyze the accuracy and complexity of the approach in section 3. In general, the matrix pencil (H, S) may not always have all positive eigenvalues, and the case where there are positive and negative eigenvalues is discussed in section 4. The connection between (1.1) and electronic structure calculations is given in section 5. The numerical results with applications to density functional perturbation theory calculations are given in section 6. We conclude in section 7.

2. Lanczos method for parametrized linear systems. We briefly describe a basic variant of the Lanczos method for solving parametrized linear systems of the form (1.1), where the matrix pencil (H, S) satisfies the conditions given in the introduction. Using the Cholesky factorization of S, we may transform (1.1) in a manner such that we instead solve

$$(R^{-*}HR^{-1} - zI)\tilde{u} = \tilde{b}, \tag{2.1}$$

where $u = R^{-1}\tilde{u}$ and $\tilde{b} = R^{-*}b$. Since $R^{-*}HR^{-1}$ is Hermitian positive definite, we now briefly describe the Lanczos method for parametrized systems under the assumption that our equation is of the form

$$(A - zI)x = \tilde{b}$$

for some Hermitian positive definite matrix A, right-hand side $\tilde{b}$, and complex shift z such that Re $z \leq 0$.

Given these systems, we note that

$$\mathcal{K}_k(A, \tilde{b}) = \mathcal{K}_k(A - zI, \tilde{b})$$

for any complex scalar z. Based on this observation, the well-known Lanczos process, see, e.g., [17] for details and [24] for a historical perspective, may be slightly modified such that at each iteration, approximate solutions, each of which satisfy the Conjugate Gradient (CG) error criteria, are produced for each shift. However, in this basic formulation, using the invariance of the Krylov subspaces precludes the use of preconditioners for the problems.

Remark 1. We note that the restriction Re $z \leq 0$ may be relaxed if we instead use a method analogous to MINRES to simultaneously solve the shifted systems; for details of MINRES, see, e.g., [32]. Such an algorithm is only slightly more expensive than the CG style algorithm given here and scales asymptotically in the same manner.

We observe that the computational cost of using this method breaks down into two distinct components. First, there is the cost of the Lanczos procedure, which only has to be computed once regardless of the number of shifts. The dominant cost of this procedure is a single matrix vector multiplication at each iteration. In the case where $S \neq I$, the method actually requires a single application of $R^{-*}HR^{-1}$, which is accomplished via two triangular solves and one matrix vector product with H. Second, for each shift z, a $k \times k$ tridiagonal system, denoted T_k, must be solved. However, this may be done very cheaply by maintaining an LDL^* factorization of T_k for each shift. (See, e.g., [17] for a detailed description of the case without a shift.) This means that the computational costs at each iteration for solving the set of subproblems scales linearly with respect to both the number of shifts and the problem size. Thus, the dominant factor in the computation is often the cost of the Lanczos process.

Furthermore, properties of the Lanczos process and the structure of the LDL^* factorization imply that if we neglect the cost of storing A, the memory costs of the algorithm scale linearly in both the number of shifts and the problem size. At the core of this memory scaling is the fact that not all of the Lanczos vectors must be stored to update the solution; see, e.g., [17]. In fact, for large problems that take many iterations, storing all of the vectors would be infeasible. However, this does impact the use of the algorithm for multiple shifts. Specifically, the set of shifts for which solutions are desired must be decided upon before the algorithm is run so that all of the necessary factorizations of T_k may be built and updated at each iteration. If a solution is needed for a new shift, the algorithm would have to be run again from the start unless all of the Lanczos vectors were stored.

3. Pole expansion for parametrized linear systems.

We now describe a method based on the work of Hale, Higham, and Trefethen [19] to simultaneously solve systems of the form

$$(H - z_l S)u_l = b, \tag{3.1}$$

where the subscript l explicitly denotes the set of N_z distinct shifts. Here, the matrix pencil (H, S) and the shifts z_l satisfy the same properties as in the introduction.

3.1. Constructing a pole expansion.

First, we provide a brief overview of the method for computing functions of matrices presented in [19]. Given an analytic function f, a Hermitian matrix A, and a closed contour Γ that only encloses the analytic region of f and winds once around the spectrum of A in a counter clockwise manner, $f(A)$ may be represented via contour integration as

$$f(A) = \frac{1}{2\pi i}\int_\Gamma f(\xi)(\xi I - A)^{-1} d\xi. \tag{3.2}$$

The authors in [19] provide a method via a sequence of conformal mappings to generate an efficient quadrature scheme for (3.2) based on the trapezoidal rule. Specifically, in [19] a map from the region $S = [-K, K] \times [0, K']$ to the upper half plane of $\Omega = \mathbb{C}\backslash((-\infty, 0] \cup [\lambda_{\min}(A), \lambda_{\max}(A)])$ is constructed via

$$z = \sqrt{\lambda_{\min}(A)\lambda_{\max}(A)}\left(\frac{k^{-1}+u}{k^{-1}-u}\right),$$
$$u = sn(t) = sn(t|k),$$
$$k = \frac{\sqrt{\lambda_{\max}(A)/\lambda_{\min}(A)}-1}{\sqrt{\lambda_{\max}(A)/\lambda_{\min}(A)}+1},$$

where $t \in S$ and z is in the upper half plane of Ω. The constants K and K' are complete elliptic integrals, and $sn(t)$ is one of the Jacobi elliptic functions. Application of the trapezoidal rule in S using the P equally spaced points

$$t_j = -K + \frac{iK'}{2} + 2\frac{(j-\frac{1}{2})K}{P}, \quad 1 \le j \le P,$$

yields a quadrature rule for computing (3.2). In [19] the assumption made is that the only nonanalytic region of f lies on the negative real axis.

Here, we instead consider the case where the nonanalytic region of f may be anywhere in the negative half plane. Therefore, for our purposes a modification of the transform in [19] must be used. Specifically, we use the construction of the quadrature presented in section 2.1 of [27], where now an additional transform of the form $\xi = \sqrt{z}$ is used to get the quadrature nodes. We note that there is a slight difference between the contour used here and the one in [27]. Because we are assuming the matrix is Hermitian positive definite we only need to consider a single branch of the square root function in defining the nodes ξ_j, in this case the positive one.

The procedure outlined may be used to generate a P term pole expansion for a function $f(A)$ denoted

$$f_P(A) \approx \sum_{k=1}^{P} w_k f(\xi_k)\left(\xi_k I - A\right)^{-1}, \tag{3.3}$$

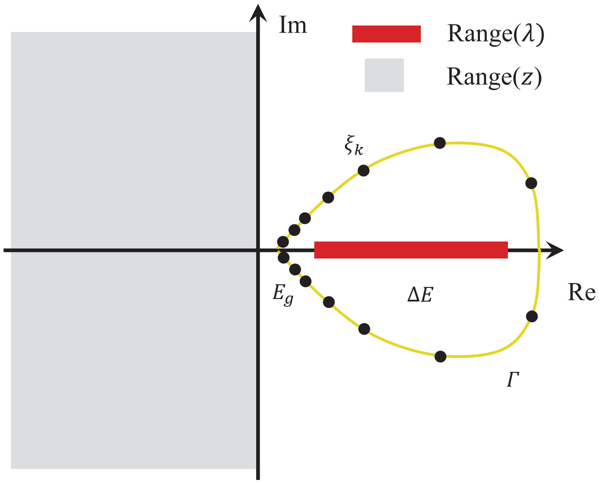


FIG. 2. *A schematic view of the range of the spectrum of the* (H, S) *pencil (thick red line on the positive real axis) and the range of the parameter* z *(light gray area), separated by a positive distance* E_g*, together of the Cauchy contour (yellow line) surrounding the positive part of the spectrum and the discretized contour integration points (poles, black dots).*

where ξ_k and w_k, $k = 1, \ldots, P$ are the quadrature nodes and weights, respectively. Both the nodes and the weights depend on P, $\lambda_{\max}(A)$, and $\lambda_{\min}(A)$. In [19], asymptotic results are given for the error $\|f(A) - f_P(A)\|$. For a Hermitian positive definite matrix A, the asymptotic error in the expansion of the form (3.3), see [19, Theorem 2.1] and [27, section 2], behaves as

$$\|f(A) - f_P(A)\| = \mathcal{O}\left(e^{-CP/\log(\lambda_{\max}(A)/\lambda_{\min}(A))}\right), \tag{3.4}$$

where C is a constant independent of $P, \lambda_{\max}(A)$, and $\lambda_{\min}(A)$.

Remark 2. We make special note of the fact that in the framework where we have all positive generalized eigenvalues, the constant in the exponent does not depend on where the function f has poles in the left half plane. Specifically, the initial use of the $z = \xi^2$ mapping maps the poles on the imaginary axis to the negative real axis; see [27] for details. All of the poles with strictly negative real part get mapped to a distinct sheet of the Riemann surface from the one that contains the positive real axis and the poles that were initially on the imaginary axis. As noted previously, the only sheet we need to consider building a contour on is the one containing the spectrum of A, and thus C does not depend on the locations of the poles.

Figure 2 illustrates the type of contours computed by this scheme and used throughout the remainder of the paper. To facilitate the computation of certain quantities necessary in the generation of these contours we used a Schwarz–Christoffel toolbox [7, 8].

3.2. Solving shifted linear systems. We now address the use of a pole expansion to solve problems of the form (3.1). Denote by $\Psi = [\psi_1, \ldots, \psi_N]$ and $\Lambda = \text{diag}[\lambda_1, \ldots, \lambda_N]$; from the generalized eigenvalue problem (1.2) we then have

$$\Psi^* H \Psi = \Lambda, \quad \Psi^* S \Psi = I.$$

We emphasize that the eigen decomposition is only used in the derivation of the method and is not performed in practical calculations. For the moment, we assume that $\lambda_1, \ldots, \lambda_N$ are positive and ordered from largest to smallest. From (1.1) we have for each shift z_l

$$\begin{aligned} u_l &= (H - z_l S)^{-1} b \\ &= \left(\Psi^{-*}\Lambda\Psi^{-1} - z_l \Psi^{-*}\Psi^{-1}\right)^{-1} b \\ &= \Psi \left(\Lambda - z_l\right)^{-1} \Psi^* b. \end{aligned}$$

Since Re $z \leq 0$, we may now use a pole expansion for $f^l(A) = 1/(A - z_l I)$ generated by the procedure in [27] to write

$$\begin{aligned} u_l^P &= \Psi \sum_{k=1}^{P} \frac{\omega_k}{\xi_k - z_l} \left(\Lambda - \xi_k\right)^{-1} \Psi^* b \\ &= \sum_{k=1}^{P} \frac{\omega_k}{\xi_k - z_l} \left(\Psi^{-*}\Lambda\Psi^{-1} - \xi_k \Psi^{-*}\Psi^{-1}\right)^{-1} b \qquad (3.5) \\ &= \sum_{k=1}^{P} \frac{\omega_k}{\xi_k - z_l} \left(H - \xi_k S\right)^{-1} b, \end{aligned}$$

which yields an approximate solution, u_l^P, to the true solution u_l.

Remark 3. Using the pole expansion method to solve the parametrized systems (1.1) motivated our choice of f. However, the use of the pole expansion only places mild requirements on f, so the method presented here may potentially be used to solve systems with different types of parametrization.

To simplify the notation, let us define h_k as the solutions to the set of problems

$$(H - \xi_k S)\, h_k = b, \tag{3.6}$$

such that our approximate solution is now simply formed as

$$u_l^P = \sum_{k=1}^{P} \frac{\omega_k}{\xi_k - z_l} h_k. \tag{3.7}$$

If we assume that h_k is computed exactly, the solutions to (3.1) formed via (3.7) satisfy the asymptotic, $P \to \infty$, error bound:

$$\frac{\|u_l - u_l^P\|_S}{\|\Psi^* b\|_2} \leq \mathcal{O}\left(e^{-CP/\log(\lambda_1/\lambda_N)}\right).$$

Given that the h_k are computed exactly, we may write

$$\begin{aligned} \|u_l - u_l^P\|_S &= \|\Psi f^l(\Lambda)\Psi^* b - \Psi f_P^l(\Lambda)\Psi^* b\|_S \\ &= \|f^l(\Lambda)\Psi^* b - f_P^l(\Lambda)\Psi^* b\|_2 \qquad (3.8) \\ &\leq \|f^l(\Lambda) - f_P^l(\Lambda)\|_2 \|\Psi^* b\|_2. \end{aligned}$$

Finally, using the error approximation (3.4) in conjunction with (3.8) yields the desired result. Furthermore, in the special case where $S = I$, the error bound reduces to

$$\frac{\|u_l - u_l^P\|_2}{\|b\|_2} \leq \mathcal{O}\left(e^{-CP/\log(\lambda_1/\lambda_N)}\right).$$

These bounds show that asymptotically the error decreases exponentially with respect to the number of poles used in the expansion for $f^l(A)$. However, there are additional errors introduced since h_k is computed inexactly. Thus, the overall error will often be dominated by the error in the computation of h_k. More specifically, let us define

$$\tilde{u}_l^P = \sum_{k=1}^{P} \frac{\omega_k}{\xi_k - z_l} \tilde{h}_k, \tag{3.9}$$

where $\tilde{h}_k$ represents an approximation of h_k. We assume that $\tilde{h}_k$ satisfies the error bound

$$\frac{\|\Psi^*(b - (H - \xi_k S)\tilde{h}_k)\|_2}{\|\Psi^* b\|_2} \leq \epsilon \tag{3.10}$$

and define $r_k = b - (H - \xi_k S)\tilde{h}_k$. In this case, the solutions to (3.1) formed via (3.9) satisfy the asymptotic, $P \to \infty$, error bound:

$$\frac{\|u_l - \tilde{u}_l^P\|_S}{\|\Psi^* b\|_2} \leq \mathcal{O}\left(e^{-CP/\log(\lambda_1/\lambda_N)}\right) + \epsilon \left(\max_{i=1,\ldots,N} \frac{1}{|\lambda_i - z_l|} \right).$$

Under the assumptions about the inexact solutions made here we may write

$$\begin{aligned} \tilde{u}_l^P &= \sum_{k=1}^{P} \frac{\omega_k}{\xi_k - z_l} \tilde{h}_k \\ &= \sum_{k=1}^{P} \frac{\omega_k}{\xi_k - z_l} \left(h_k - (H - \xi_k S)^{-1} r_k \right) \\ &= u_l^P - \sum_{k=1}^{P} \frac{\omega_k}{\xi_k - z_l} (H - \xi_k S)^{-1} r_k. \end{aligned}$$

In conjunction with the error bound (3.10), this implies that

$$\begin{aligned} \frac{\|u_l - \tilde{u}_l^P\|_S}{\|\Psi^* b\|_2} &= \frac{\|u_l - u_l^P + \Psi f_P^l(\Lambda)\Psi^* r_k\|_S}{\|\Psi^* b\|_2} \\ &\leq \frac{\|u_l - u_l^P\|_S}{\|\Psi^* b\|_2} + \frac{\|f_P^l(\Lambda)\Psi^* r_k\|_2}{\|\Psi^* b\|_2} \\ &\leq \frac{\|u_l - u_l^P\|_S}{\|\Psi^* b\|_2} + \epsilon \|f_P^l(\Lambda)\|_2 \\ &\leq \frac{\|u_l - u_l^P\|_S}{\|\Psi^* b\|_2} + \epsilon \left(\|f_P^l(\Lambda) - f^l(\Lambda)\|_2 + \|f^l(\Lambda)\|_2 \right) \\ &\leq \frac{\|u_l - u_l^P\|_S}{\|\Psi^* b\|_2} + \epsilon \|f_P^l(\Lambda) - f^l(\Lambda)\|_2 + \epsilon \left(\max_{i=1,\ldots,N} \frac{1}{|\lambda_i - z_l|} \right). \end{aligned} \tag{3.11}$$

Finally, using the estimate (3.4) along with (3.11) yields the desired result. Once again, in the case where $S = I$, the error bound simplifies to

$$\frac{\|u_l - \tilde{u}_l^P\|_2}{\|b\|_2} \leq \mathcal{O}\left(e^{-CP/\log(\lambda_1/\lambda_N)}\right) + \epsilon \left(\max_{i=1,\ldots,N} \frac{1}{|\lambda_i - z_l|} \right).$$

The error bound shows us that the error may be dominated by either the error in the pole expansion or the error in the solutions of (3.6). Since the error in the pole expansion decays exponentially, it is often best to control the overall error via the relative error requested when solving (3.6).

Given that we are interested in solving systems of the form (3.1) for a large number of shifts, the key observation in (3.5) is that the vectors h_k are independent of the shifts z_l because the ξ_k are independent of z_l. Therefore, this method parametrizes the solutions to N_z linear systems of the form (3.1) on the solutions of P independent linear systems of the form (3.1). In fact, once this parametrization has been done, the method is completely flexible and any method may be used to solve the subproblems.

Remark 4. Because the ξ_k appear in complex conjugate pairs, if H, S, and b are real, then so do the solutions to the systems (3.6). Therefore, in this situation we only have to solve $P/2$ systems, where for simplicity we assume P is even.

The bulk of the computational cost in this method is the necessity of solving P systems of the form (3.6), after which the vectors h_k may be combined with different weights to yield approximation solutions for as many distinct z_l as desired. In fact, because the systems are completely independent, this method can be easily parallelized with up to P (or $P/2$ depending on whether symmetry is used) machines. Once the subproblems have been solved, computing a solution for all N_z shifts costs $\mathcal{O}(PN_zN)$. Furthermore, as long as the h_k are saved, solutions for new shifts may be computed as needed with negligible computational cost. This is in contrast to the Lanczos method, where, as discussed in section 2, computing a solution for a new shift generally requires running the algorithm again from the start unless all $\{v_i\}$ are stored.

In some situations, the set of systems (3.6) may even be simultaneously solved using existing Krylov based methods for simultaneously solving shifted systems; see, e.g., [12, 13, 37, 6, 30, 15, 3, 9, 36]. Perhaps the simplest example would be to use the MINRES variation of the method outlined in section 2. In this situation the same methods may be applicable to the original systems; however, the pole expansion reduces the number of shifts that have to be solved for from N_z to P. Similarly, if a good preconditioner is known for each of the P distinct systems, then each system may be independently solved via an iterative method.

If the systems are amenable to the use of a direct method, e.g., LU factorization or factorizations as in [29], then the N_z shifted systems may be solved by computing factorizations of the matrices $(H-\xi_k S)$ and using those factorizations to solve the subproblems. If a direct method is used, the procedure also allows for efficiently solving shifted systems with multiple right-hand sides. The specific solution methodologies we used for the subproblems of the pole expansion method will be discussed in section 6.

4. Indefinite systems.

Up until this point we have considered Hermitian matrix pencils (H, S) for which the generalized eigenvalues are all positive. Motivated by the applications we discuss in section 5, we now discuss the case where there are both positive and negative generalized eigenvalues and we seek a solution of a specific form. We still require that H and S are Hermitian and that S is positive definite. We assume that the generalized eigenvalues of (H, S) are ordered such that

$$\lambda_1 \geq \cdots \geq \lambda_M > 0 > \lambda_{M+1} \geq \cdots \geq \lambda_N.$$

Let $\Psi_+ = [\psi_1, \ldots, \psi_M]$ and $\Psi_- = [\psi_{M+1}, \ldots, \psi_N]$. Similarly, let $\Lambda_+ = \mathrm{diag}[\lambda_1, \ldots, \lambda_M]$ and $\Lambda_- = \mathrm{diag}[\lambda_{M+1}, \ldots, \lambda_N]$. While the notation used here mirrors that earlier in the paper, the generalized eigenvectors and eigenvalues here are distinct from the rest of the paper.

4.1. Lanczos method. For the Lanczos-based method used here, there are two considerations that have to be made with respect to indefinite systems. For the purposes of this section, we make the simplification, as in section 2, that we first transform the problem in a manner such that $S = I$. Under this assumption we are interested in solving systems for which the right-hand side $\tilde{b}$ satisfies $\Psi_-^*\tilde{b} = 0$ and the solution satisfies $\Psi_-^* x^l = 0$. We note that in exact arithmetic, $\Psi_-^*\tilde{b} = 0$ implies that $\Psi_-^* x^l = 0$. However, we must ensure that the numerical method used to solve these systems maintains this property. Specifically, we need to ensure that $\Psi_-^* x_k^l \approx 0$, and, if we are using a CG style method, we must ensure that the subproblems remain nonsingular. Both of these conditions are reliant upon the mutual orthogonality between the Lanczos vectors and the columns of Ψ_-.

In practice, we have observed that there is no excessive build up of components in the Ψ_- directions among the Lanczos vectors and thus good orthogonality is maintained between the solutions we compute using the Lanczos-based methods and the negative generalized eigenspace. Furthermore, in practice if we assume that Ψ_- is known, then at each step of the Lanczos process we may project out any components of the Lanczos vectors that lie in the negative generalized eigenspace. Such a procedure enforces orthogonality, up to the numerical error in the projection operation, between the computed solution x_k and Ψ_-.

4.2. Pole expansion. The expansion we used in (3.5) is valid for Hermitian positive definite matrices. For the case where the diagonal matrix of generalized eigenvalues Λ has both positive and negative entries, the pole expansion does not directly give accurate results. However, the pole expansion is still applicable in the case where we wish to solve the systems projected onto the positive generalized eigenspace. Section 5 provides the motivation for considering such problems. Similar to before, we are interested in solving systems for which the right-hand side b satisfies $\Psi_-^* b = 0$ and the solution satisfies $\Psi_-^* S u_l = 0$. To accomplish this, we use a pole expansion using λ_1 and λ_M as the bounds of the spectrum. We note that in exact arithmetic, $\Psi_-^* b = 0$ implies that $\Psi_-^* S u_l = 0$. However, we must ensure that our numerical methods retain this property. Earlier in this section we discussed the impact of this generalization on the Lanczos-based solver. Here we restrict our discussion to the impact of solving an indefinite system on the pole expansion method and provide an argument for why we do not observe difficulty in this regime.

If we once again assume that $\tilde{h}_k$ satisfies (3.10), we may conclude that

$$\frac{\|u_l - \tilde{u}_l^P\|_S}{\|\Psi_+^* b\|_2} \leq \mathcal{O}\left(e^{-CP/\log(\lambda_1/\lambda_M)}\right) + \mathcal{O}\left(\epsilon\right)\left(\max_{i=1,\dots,M}\frac{1}{|\lambda_i - z_l|}\right).$$

To argue such a bound, in a manner similar to before, we may write

$$\begin{aligned}
\frac{\|u_l - \tilde{u}_l^P\|_S}{\|\Psi_+^* b\|_2} &= \frac{\|u_l - u_l^P + \Psi f_P^l(\Lambda)\Psi^* r_k\|_S}{\|\Psi^* b\|_2} \\
&\leq \frac{\|u_l - u_l^P\|_S}{\|\Psi^* b\|_2} + \frac{\|f_P^l(\Lambda)\Psi^* r_k\|_2}{\|\Psi_+^* b\|_2} \\
&\leq \frac{\|u_l - u_l^P\|_S}{\|\Psi^* b\|_2} + \frac{\|f_P^l(\Lambda_+)\Psi_+^* r_k\|_2 + \|f_P^l(\Lambda_-)\Psi_-^* r_k\|_2}{\|\Psi_+^* b\|_2} \\
&\leq \frac{\|u_l - u_l^P\|_S}{\|\Psi^* b\|_2} + \epsilon\|f_P^l(\Lambda_+)\|_2 + \frac{\|f_P^l(\Lambda_-)\Psi_-^* r_k\|_2}{\|\Psi_+^* b\|_2} \\
&\leq \frac{\|u_l - u_l^P\|_S}{\|\Psi^* b\|_2} + \epsilon\left(\|f_P^l(\Lambda_+) - f^l(\Lambda_+)\|_2 + \|f^l(\Lambda_+)\|_2\right) + \epsilon\|f_P^l(\Lambda_-)\|_2.
\end{aligned}$$

Here we observe that the rational function f_P^l is well behaved on the negative real axis in a manner dependent on the closest poles, which are of order λ_M away. Furthermore, if the residuals r_k are orthogonal to the negative general eigenspace, this term vanishes. Also, the construction of f_P^l once again implies a dependence on the distance between the shifts and the generalized eigenvalues of the system. When using the pole expansion method on an indefinite system, as long as the subproblems are appropriately solved the use of the expansion basically maintains the accuracy of the overall solution. Furthermore, as long as the overall solution is computed accurately enough, and as long as S is reasonably conditioned, we cannot have large components of the computed solution in the negative generalized eigenspace. Furthermore, if necessary we may simply project out the components of $\tilde{h}_k$ in the negative generalized eigenspace to ensure that the overall solution error is not impacted by the subproblem solution method.

5. Connection with electronic structure calculation. In this section we discuss the connection between the problem with multiple shifts in (1.1) and several aspects of the electronic structure theory, which are based on perturbative treatment of Kohn–Sham density functional theory [21, 23] (KSDFT). KSDFT is the most widely used electronic structure theory for describing the ground state electronic properties of molecules, solids, and other nano structures. To simplify our discussion, we assume the computational domain is $\Omega = [0, L]^3$ with periodic boundary conditions. We use linear algebra notation, and we do not distinguish integral operators from their kernels. For example, we may simply denote $\hat{f}(\mathbf{r}) = \hat{A}[\hat{g}](\mathbf{r}) \equiv \int \hat{A}(\mathbf{r}, \mathbf{r}')\hat{g}(\mathbf{r}')\,\mathrm{d}\mathbf{r}'$ by $\hat{f} = \hat{A}\hat{g}$ and represent the operator $\hat{A}$ by its kernel $\hat{A}(\mathbf{r}, \mathbf{r}')$. All quantities represented in the real space are given in the form such as $\hat{H}(\mathbf{r}, \mathbf{r}')$ and $\hat{f}(\mathbf{r})$, and the corresponding matrix or vector coefficients represented in a finite dimensional basis set is given in a form such as H and f.

The Kohn–Sham equation defines a nonlinear eigenvalue problem

$$\begin{aligned} &\hat{H}[\hat{\rho}]\hat{\psi}_i = \varepsilon_i\hat{\psi}_i, \\ &\hat{\rho}(\mathbf{r}) = \sum_{i=1}^{N_e}|\hat{\psi}_i(\mathbf{r})|^2, \qquad \int \hat{\psi}_i^*(\mathbf{r})\hat{\psi}_j(\mathbf{r})\,\mathrm{d}\mathbf{r} = \delta_{ij}, \end{aligned} \tag{5.1}$$

where N_e is the number of electrons. (Spin degeneracy is omitted here for simplicity.) The eigenvalues $\{\varepsilon_i\}$ are ordered nondecreasingly. The lowest N_e eigenvalues $\{\varepsilon_i\}_{i=1}^{N_e}$ are called the occupied state energies, and $\{\varepsilon_i\}_{j>N_e}$ are called the unoccupied state energies. We assume $\varepsilon_{N_e+1} - \varepsilon_{N_e} > 0$, i.e., the system is an insulating system [28]. The eigenfunctions $\{\hat{\psi}_i\}_{i=1}^{N_e}$ define the electron density $\hat{\rho}(\mathbf{r})$, which in turn defines the Kohn–Sham Hamiltonian

$$\hat{H}[\hat{\rho}] = -\frac{1}{2}\Delta + \hat{V}_c[\hat{\rho}] + \hat{V}_{\mathrm{xc}}[\hat{\rho}] + \hat{V}_{\mathrm{ion}}. \tag{5.2}$$

Here, Δ is the Laplacian operator for characterizing the kinetic energy of electrons.

$$\hat{V}_c[\hat{\rho}](\mathbf{r}) \equiv \int \frac{\hat{\rho}(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|}\,\mathrm{d}\mathbf{r}'$$

is the Coulomb potential which is linear with respect to the electron density $\hat{\rho}$. $\hat{V}_{\mathrm{xc}}[\hat{\rho}]$ is a nonlinear functional of $\hat{\rho}$, characterizing the many body exchange and correlation effect. $\hat{V}_{\mathrm{ion}}$ is the electron-ion interaction potential and is independent of $\hat{\rho}$. Because the eigenvalue problem (5.1) is nonlinear, it is often solved iteratively by a class of algorithms called self-consistent field iterations (SCF) [28], until (5.2) reaches self-consistency.

When the self-consistent solution of the Kohn–Sham equation is obtained, one may perform post Kohn–Sham calculations for properties within and beyond the ground state properties of the system. Examples of such calculations include the density functional perturbation theory (DFPT) [5, 18, 4], the GW theory [20, 2, 11, 39, 34, 16], and the random phase approximation (RPA) of the electron correlation energy [25, 26, 14, 31]. In these theories, a key quantity is the so-called independent particle polarizability matrix, often denoted by $\hat{\chi}_0(\omega)$. The independent particle polarizability matrix characterizes the first order *non-self-consistent* response of the electron density $\delta\hat{\rho}_0(\omega)$ with respect to the time dependent external perturbation potential $\delta\hat{V}(\omega)$, where ω is the frequency of the time dependent perturbation potential. ω can be chosen to be 0, characterizing the static linear response of the electron density with respect to the static external perturbation potential.

In the density functional perturbation theory, the first order *self-consistent* static response (i.e., the physical response) of the electron density $\delta\hat{\rho}(\mathbf{r})$ with respect to the static external perturbation potential $\delta\hat{V}$ can be computed as

$$\delta\hat{\rho} = \hat{\chi}(0)\delta\hat{V},$$

where the operator $\hat{\chi}(0)$ is directly related to $\hat{\chi}_0(0)$ as

$$\hat{\chi}(0) = \left[I - \hat{\chi}_0(0)\left(\hat{V}_c + \frac{\delta\hat{V}_{\mathrm{xc}}}{\delta\hat{\rho}}\right)\right]^{-1}\hat{\chi}_0(0).$$

Many body perturbation theories such as the GW theory computes the quasi-particle energy which characterizes the excited state energy spectrum of the system. The key step for calculating the quasi-particle energy is the computation of the screened Coulomb operator, which is defined as

$$\hat{W}(i\omega) = \left(I - \hat{V}_c\hat{\chi}_0(i\omega)\right)^{-1}\hat{V}_c.$$

Here, $\chi_0(i\omega)$ should be computed on a large set of frequencies on the imaginary axis $i\omega$.

The RPA of the electron correlation energy improves the accuracy of many existing exchange-correlation functional in ground state electronic structure calculation. Using the adiabatic connection formula [25, 26], the correlation energy can be expressed as

$$E_c = -\frac{1}{2\pi}\int_0^1\int_0^\infty \mathrm{Tr}\left\{\hat{V}_c[\hat{\chi}_\lambda(i\omega) - \hat{\chi}_0(i\omega)]\right\}\,\mathrm{d}\omega\,\mathrm{d}\lambda,$$

and $\hat{\chi}_\lambda(i\omega)$ can be computed from $\hat{\chi}_0(i\omega)$ as

$$\hat{\chi}_\lambda(i\omega) = \hat{\chi}_0(i\omega) + \lambda\hat{\chi}_0(i\omega)\hat{V}_c\hat{\chi}_\lambda(i\omega).$$

In all the examples above, computing $\hat{\chi}_0(i\omega)$ is usually the bottleneck. For simplicity, we consider the case when $\hat{H}$ is real and therefore the eigenfunctions $\hat{\psi}_i(\mathbf{r})$

are also real. In such a case, the kernel $\hat{\chi}_0(i\omega)(\mathbf{r},\mathbf{r}')$ can be computed from the Adler–Wiser formula [1, 40] as

$$\hat{\chi}_0(\mathbf{r},\mathbf{r}',i\omega) = 2\mathrm{Re} \sum_{i\le N_e, j>N_e} \frac{\hat{\psi}_i(\mathbf{r})\hat{\psi}_j(\mathbf{r})\hat{\psi}_j(\mathbf{r}')\hat{\psi}_i(\mathbf{r}')}{\varepsilon_i-\varepsilon_j+i\omega}.$$

The summation $\sum_{j>N_e}$ requires the computation of a large number of eigenstates, which is usually prohibitively expansive. Recent techniques [39, 34, 16] have allowed one to avoid the direct computation of $\{\hat{\psi}_j\}_{j>N_e}$ when $\hat{\chi}_0(i\omega)$ is multiplied to an arbitrary vector $\hat{g}$ as

$$\begin{aligned}\hat{\chi}_0(i\omega)[\hat{g}](\mathbf{r}) &= 2\mathrm{Re} \sum_{i\le N_e} \hat{\psi}_i(\mathbf{r}) \int \sum_{j>N_e} \frac{\hat{\psi}_j(\mathbf{r})\hat{\psi}_j(\mathbf{r}')}{\varepsilon_i-\varepsilon_j+i\omega}\hat{\psi}_i(\mathbf{r}')\hat{g}(\mathbf{r}')\,\mathrm{d}\mathbf{r}' \\ &\equiv 2\mathrm{Re} \sum_{i\le N_e} \hat{\psi}_i(\mathbf{r})\hat{u}_i(\mathbf{r}).\end{aligned} \tag{5.3}$$

Here, $\hat{u}_i(\mathbf{r})$ can be solved through the equation

$$\hat{Q}\left[\hat{H}-(\varepsilon_i+i\omega)\right]\hat{Q}\hat{u}_i = -\hat{Q}[\hat{\psi}_i\odot\hat{g}]. \tag{5.4}$$

The operator $\hat{Q}(\mathbf{r},\mathbf{r}')=\delta(\mathbf{r},\mathbf{r}')-\sum_{i\le N_e}\hat{\psi}_i(\mathbf{r})\hat{\psi}_i^*(\mathbf{r}')$ is a projection operator onto the occupied states (noting $\hat{\psi}_i^*(\mathbf{r}')=\hat{\psi}_i(\mathbf{r}')$ is real), and $[\hat{\psi}_i\odot\hat{g}](\mathbf{r})\equiv\hat{\psi}_i(\mathbf{r})\hat{g}(\mathbf{r})$ is the elementwise product between two vectors.

Without loss of generality we may set the largest occupied state energy $\varepsilon_{N_e}=0$. Equation (5.4) can be reduced to the form

$$(\hat{H}-z)\hat{u}=\hat{f} \tag{5.5}$$

for multiple shifts $z=\varepsilon_i+i\omega$ ($\mathrm{Re}\, z\le 0$) and multiple right-hand sides $\hat{f}$. Furthermore, $\hat{u},\hat{f}\in\mathrm{Ran}(\hat{Q})$, where Ran is the range of the operator $\hat{Q}$. Equation (5.5) can be solved in practice using a finite dimensional basis set, such as finite element, plane waves, or more complicated basis functions such as numerical atomic orbitals [38]. We denote the basis set by a collection of column vectors as $\hat{\Phi}=[\hat{\varphi}_1(\mathbf{r}),\ldots,\hat{\varphi}_N(\mathbf{r})]$. The overlap matrix associated with the basis set $\hat{\Phi}$ is

$$S=\hat{\Phi}^*\hat{\Phi},$$

and the projected Hamiltonian matrix in the basis $\hat{\Phi}$ is

$$H=\hat{\Phi}^*\hat{H}\hat{\Phi}.$$

Using the ansatz that both the solution and the right-hand side can be represented using the basis set $\hat{\Phi}$ as

$$\hat{u}=\hat{\Phi}u,\quad \hat{f}=\hat{\Phi}f, \tag{5.6}$$

(5.5) becomes

$$(H-zS)u=\hat{\Phi}^*\hat{f}=Sf,$$

which is (1.1) with $b = Sf$. Using the eigen decomposition of the matrix pencil (H, S) as in (1.2), each eigenfunction in the real space $\hat{\psi}_i(\mathbf{r})$ is given using the basis set $\hat{\Phi}$ as

$$\hat{\psi}_i = \hat{\Phi}\psi_i. \tag{5.7}$$

Combining (5.7) and (5.6), the condition $\hat{u} \in \operatorname{Ran}(\hat{Q})$ becomes

$$\hat{\psi}_i^* \hat{u} = \psi_i^* S u = 0 \quad \forall i \leq N_e, \tag{5.8}$$

and similarly $\hat{f} \in \operatorname{Ran}(\hat{Q})$ becomes

$$\psi_i^* S f = 0 \quad \forall i \leq N_e. \tag{5.9}$$

In practice, the conditions (5.8) and (5.9) can be satisfied by a projection procedure as described in section 4.

6. Numerical results.

First, we illustrate the accuracy and the scaling of the pole expansion method. We then present two distinct numerical examples based on the general method presented here. In one example we consider the case where $S = I$ and an iterative method is used to solve the subproblems, and in the second example we consider the case where $S \neq I$ and a direct method is used to solve the subproblems. In each case, the method presented here is compared with the Lanczos style method described in section 2.

All of the numerical experiments were run in MATLAB on a Linux machine with four 2.0GHz eight core CPUs and 256GB of RAM.

6.1. Accuracy of the pole expansion.

Before presenting the examples motivated by the preceding section, we first demonstrate the behavior of the pole expansion method for approximating

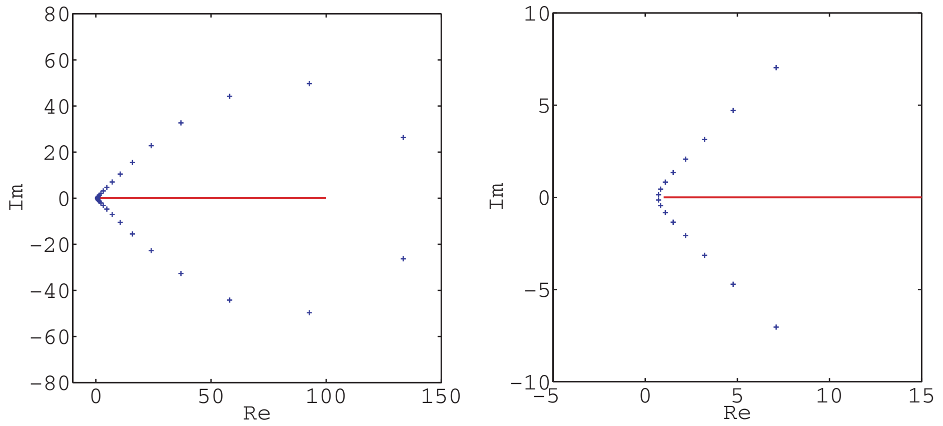
$$f(x; z) = \frac{1}{x - z}$$

on some interval of the positive real axis. Figure 3 shows an example contour as computed via the method in section 3 when the region of interest is $[1, 100]$.

First, we consider approximating $f(x; z)$ for a fixed value $z = i$ on the interval $[1, 1000]$. Figure 4(a) shows the $\|\cdot\|_\infty$ error of approximating $f(x; i)$ by $f_P(x; i)$, computed via sampling at 10,000 equally spaced points in $[1, 1000]$ along the x direction, as the number of poles P increases. We observe the exponential decay in error as the number of poles increases, which is aligned with the error analysis presented earlier.

Next, we illustrate the number of poles required to reach fixed accuracy as x approaches 0, i.e., as $E_g \to 0$. We consider approximating $f(x; i)$ via $f_P(x; i)$ on the interval $[\sigma, 10]$ for 20 values of $\sigma \in [10^{-4}, 1]$ that are equally spaced on the logarithmic scale. Figure 4(b) shows the number of poles required such that $\|f_P(x; i) - f(x; i)\|_\infty$ is less than 10^{-8} over the interval $[\sigma, 10]$.

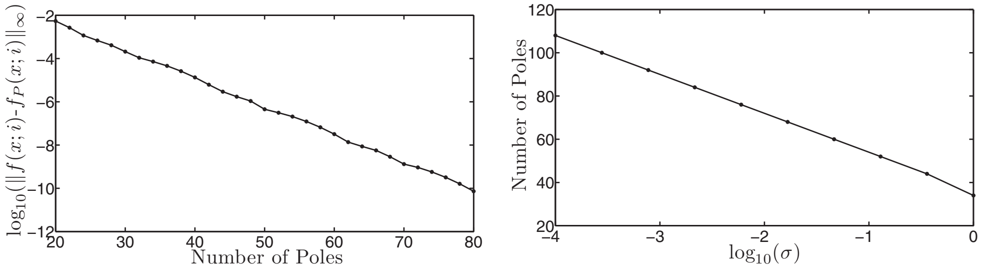
Finally, we consider the approximation of $f(x; z)$ for a wide range of z with Re $z \leq 0$ and demonstrate how $\|f_P(x; z) - f(x; z)\|_\infty$ varies as z changes using the same pole expansion. We fixed the number of poles used in the approximation to be 60. Figure 5 shows that high accuracy is maintained for all z in the left half plane. Here we kept the region of interest as $x \in [1, 1000]$ and consider the accuracy at 5000 distinct z evenly distributed with Re $z \in [-50, 0]$ and Im $z \in [-50, 50]$.



(a) View of the entire contour.

(b) View near the origin.

FIG. 3. *Example of quadrature nodes (blue +'s) used when the spectrum of A (red line) lies in the interval* $[1, 100]$.



(a)

(b)

FIG. 4. *Behavior of the pole expansion.* (a) $\|f_P(x;i) - f(x;i)\|_\infty$*, on a* $\log_{10}$ *scale, for the interval* $x \in [1, 1000]$ *as the number of poles,* P*, is increased.* (b) *Number of poles required to achieve* $\|f_P(x;i) - f(x;i)\|_\infty \leq 10^{-8}$ *for* $x \in [\sigma, 10]$.

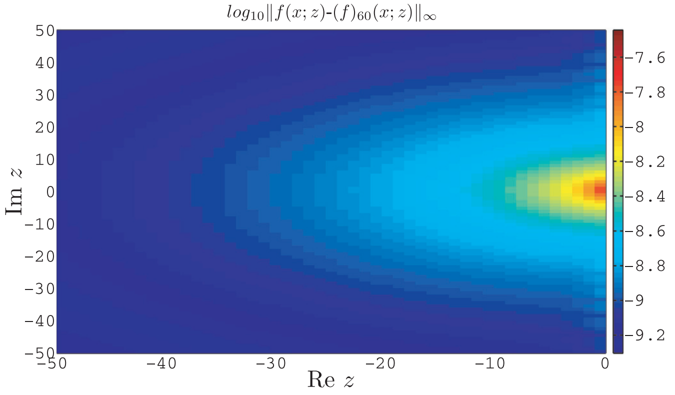


FIG. 5. $\|f_P(x;z) - f(x;z)\|_\infty$ *on a* $\log_{10}$ *scale for the interval* $x \in [1, 1000]$ *as* z *is varied.*

6.2. Orthogonal basis functions: $S = I$. First, we consider the case where orthogonal basis functions are used, and thus in the notation here $S = I$. The Hamiltonian matrix H takes the form

$$H = -\frac{1}{2}\Delta + V,$$

and the operator V is obtained by solving the Kohn–Sham density functional theory problem for a benzene molecule using the KSSOLV package [41], which is a MATLAB toolbox for solving Kohn–Sham equations for small molecules and solids implemented entirely in MATLAB m-files. The benzene molecule has 30 electrons and 15 occupied states (spin degeneracy of 2 is counted here). The atomic configuration of the benzene molecule is given in Figure 8(a).

The computational domain is $[0, 20) \times [0, 20) \times [0, 10)$ along the x, y, z directions, respectively. The computational domain is discretized into $64 \times 64 \times 32 = 131072$ points. The Laplacian operator is discretized using the plane wave basis set, and this set of grid corresponds to the usage of the kinetic energy cutoff at 50.5 Hartree. The Laplacian operator is applied using spectral method and is done efficiently using the Fast Fourier Transform (FFT). The 15 negative eigenvalues (occupied states) and the corresponding eigenvectors are computed using the locally optimal block preconditioned CG (LOBPCG) method [22], with a preconditioner of the form $(-\frac{1}{2}\Delta + .001)^{-1}$. We only use these computed eigenvectors to ensure that the right-hand side for the set of equations we solve is orthogonal to the negative eigenspace of the (H, S) pencil, as required in section 4 for using the pole expansion for indefinite systems.

We now solve the set of problems

$$(H - i\eta_l I)x_l = b \tag{6.1}$$

for 101 equispaced η_l in the interval $[-10, 10]$. Because H and b are real, some systems are in essence solved redundantly. However, here we are interested in the performance given the number of shifts and not the specific solutions.

We monitor the behavior of our method and the Lanczos method as the condition number of H increases. To this end we sequentially refine the number of discretization points in each direction by a factor of 2 and generate a potential function V via Fourier interpolation. The grid size is denoted by $N \times N \times N/2$, and the largest problem we consider is discretized on a $256 \times 256 \times 128$ grid. For these large problems, the eigenvectors associated with negative eigenvalues are approximated by Fourier interpolates of the computed eigenvectors for the smallest problem.

To compare the methods, we solved the set of problems (6.1) via the pole expansion method using a number of poles that depended on the size of the problem. To combat the slight loss of accuracy that occurs for a fixed number of poles as the condition number of the matrix grows, we increased the number of poles as the problem size grew. During this step, preconditioned GMRES [35] in MATLAB was used to solve the subproblems associated with the pole expansion. The preconditioner used is of the form

$$\left(-\frac{1}{2}\Delta + \xi_k\right)^{-1},$$

and similarly to H it was efficiently applied via the FFT. The requested accuracy of the GMRES routine is that the relative residual is less than 10^{-7}. Since H and b are real, only $P/2$ subproblems had to be solved. We remark that though the solution for different poles can be straightforwardly parallelized, here we performed the calculations sequentially in order to compare with the sequential implementation of the Lanczos method. Let $\tilde{x}_l^P$ denote the approximate solutions computed using the pole expansion. The relative error metric

$$r_l^P = \frac{\|b - (H - i\eta_l I)\tilde{x}_l^P\|_2}{\|b\|_2}$$

is then computed for each shift after the approximate solutions have been computed. Finally, the Lanczos procedure described in section 2 is called with a requested error tolerance of

$$\frac{\|b-(H-i\eta_l I)\hat{x}_l\|_2}{\|b\|_2} \leq \max_l r_l^P, \tag{6.2}$$

where $\hat{x}_l$ denotes the approximate solution computed via the Lanczos method. Since the Lanczos method reveals the residual for each shift at each iteration, this stopping criteria is cheap computationally. Thus, the Lanczos method was run until all of the approximate solutions met the stopping criteria (6.2). The Lanczos method used here takes advantage of the cheap updates briefly described in section 2, and to further save on computational time, once a solution for a given η_l was accurate enough, the implementation stopped updating that solution. For comparison purposes we also solved the problem with a version of the Lanczos algorithm that uses Householder reflectors to maintain orthogonality amongst the Lanczos vectors [17]. For the smallest size problem used here, and without any shifts, the version from section 2 took 460 iterations to converge to 10^{-8} accuracy while the version that maintained full orthogonality took 456 iterations to converge to the same accuracy. However, the method that maintained full orthogonality took 65 times longer to run and would be prohibitively expensive for the larger problems given the increased problem size and iteration count, so for all the comparisons here we use the CG style method outlined in section 2.

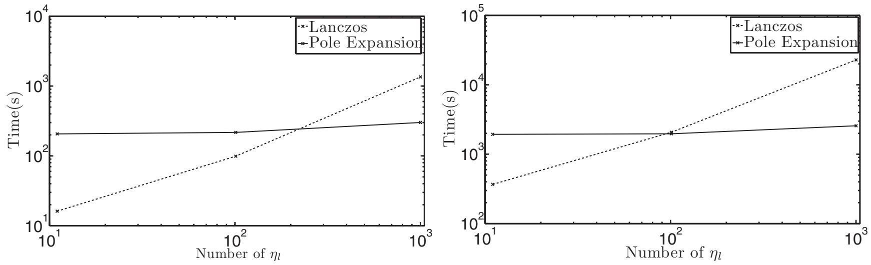
The spectrum of the operator H grows as $\mathcal{O}(N^2)$, and we observe that the number of iterations required for the Lanczos method to converge grow roughly as $\mathcal{O}(N)$. We do not expect to see such growth of the number of iterations in the pole expansion method, given the preconditioner used in solving the subproblems. Furthermore, even though the cost per iteration of solving for multiple shifts scales linearly in the Lanczos method, the time required is dependent on the number of iterations required to converge. In fact, in this case where H may be applied very efficiently for a very large number of shifts, the cost at each iteration may be dominated by the additional computational cost associated with each shift. In contrast, once the subproblems have been solved, the pole expansion method has a fixed cost for computing the solutions for all the η_l, which only depends on the number of shifts and the number of poles. If $|\eta_l|$ is large, the Lanczos method will converge very quickly for this specific problem, so in the case where only shifts with large magnitude are considered the Lanczos method may perform better. However, section 5 motivates our use of shifts spaced out along a portion of the imaginary axis that includes 0.

Table 1 reports the results of the pole expansion method and the Lanczos method for the problem described above. For the pole expansion method, the number of iterations reported is the total number of iterations required to solve all of the subproblems. In both cases the error reported is the maximum computed relative residual over the shifts. Finally, the total time taken to solve the problems is reported for each method.

Table 1

Comparison of the pole expansion method and the Lanczos method for solving $(H-i\eta_l I)x_l = b$.

	Lanczos			Pole expansion			
Problem size	Iter.	Time(s)	Error	P	Iter.	Time(s)	Error
$64 \times 64 \times 32$	297	97.23	9.83×10^{-7}	70	1005	181.99	9.85×10^{-7}
$128 \times 128 \times 64$	911	2073.89	6.91×10^{-6}	80	969	1920.75	6.93×10^{-6}
$256 \times 256 \times 128$	1746	35129.79	6.83×10^{-6}	90	959	17127.93	6.85×10^{-6}



(a) $64 \times 64 \times 32$

(b) $128 \times 128 \times 64$

FIG. 6. *Time taken to solve problems for a varying number of* η_l *equispaced in* $[-10, 10]$.

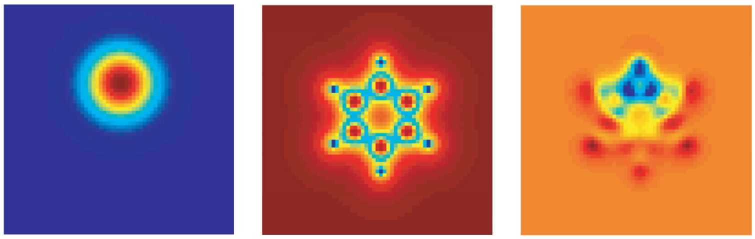
We observe that, as expected, the Lanczos method performs better for the smallest-sized problem. However, once we reach the mid-sized problem the methods perform comparably, and the pole expansion method is actually a bit faster even though it takes a few more iterations overall. What is important to notice is that the overall iteration count of solving all the subproblems associated with the pole expansion method remains relatively constant even with the increased number of poles. By the time we reach the largest problem, the pole expansion method outperforms the Lanczos method, taking about half the time to solve the set of problems.

Remark 5. Because the GMRES method used here uses Householder transforms to maintain orthogonality among the Krylov basis, it is not very memory efficient. However, we also ran this example using preconditioned TFQMR [10] in place of GMRES, and while the iteration count more than doubled for the pole expansion method, it was actually faster since the applications of the Householder matrices in GMRES is very expensive. This comparison actually helps demonstrate the flexibility of the method with respect to the solver used. If a fast method was not available for applying H, then GMRES may be preferable; however, in the case where H may be applied very efficiently methods such as TFQMR may be better suited to the problem.

To further demonstrate the scaling of the methods with respect to the number of shifts, we ran the problem at a fixed size and varied the number of shifts. As before, preconditioned GMRES was used in the pole expansion method and no parallelism was used when solving the subproblems. The same strategy as above was used to ensure that the Lanczos method stopped once it had solved all the problems as accurately as the least accurate solution found using the pole expansion. Figure 6(a) shows the time taken to solve the problems of size $64 \times 64 \times 32$ for a varying number of η_l equispaced in $[-10, 10]$. In all cases, the largest relative residual was on the order of 7×10^{-7}. Similarly, Figure 6(b) shows the time taken to solve the problems of size $128 \times 128 \times 64$ for a varying number of η_l equispaced in $[-10, 10]$. In all cases, the largest relative residual was on the order of 7×10^{-6}. Here we observe that in both cases, the pole expansion method scales very well as the number of shifts increases, especially in comparison to the Lanczos method. For example, even in the case where the problem is $64 \times 64 \times 32$ and the Lanczos method takes around a third of the iterations of the pole expansion method, if you take the number of shifts to be large enough the pole expansion method becomes considerably faster.

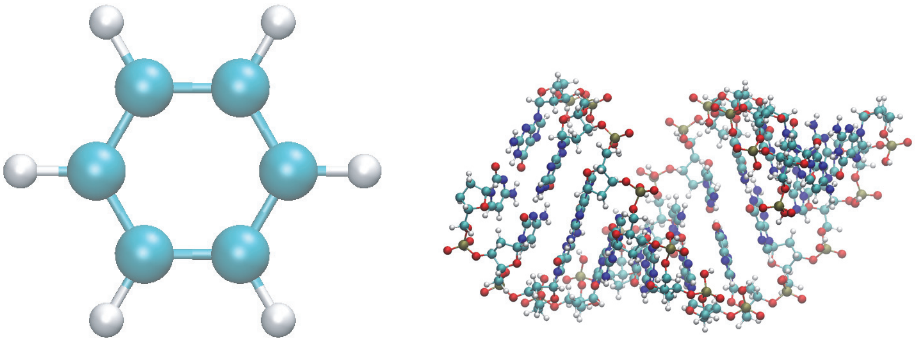
6.3. Computation of $\hat{\chi}_0(i\omega)[\hat{g}](\mathbf{r})$.

Motivated by our discussion in section 5, we used our technique to compute $\hat{\chi}_0(i\omega)[\hat{g}](\mathbf{r})$ as defined in (5.3), where we only need $\hat{\psi}_i(\mathbf{r})$ for $i \le N_e$. In order to compute $\hat{\chi}_0(i\omega)[\hat{g}](\mathbf{r})$ for a large number of $i\omega$, we



(a) Slice of $\hat{g}$. (b) Slice of V. (c) Slice of $\hat{\chi}_0(0)[\hat{g}](\mathbf{r})$.

FIG. 7. *Slices* $\hat{g}$ *and* $\hat{\chi}_0(0)[\hat{g}](\mathbf{r})$ *at the vertical midpoint of domain.*



(a) (b)

FIG. 8. (a) *The atomic configuration of a benzene molecule with* 12 *atoms.* (b) *The atomic configuration of a DNA molecule with* 715 *atoms.*

had to compute $\hat{u}_i(\mathbf{r})$ via (5.4). Physically, this corresponds to the calculation of the response of the electron density with respect to external perturbation potential $\hat{g}(\mathbf{r})$.

We constructed $\hat{g}$ as a Gaussian centered at one of the carbon atoms in the benzene molecule. Figure 7(a) shows a slice along the z direction of the function $\hat{g}$, and 7(b) shows the corresponding slice of potential function V for the benzene molecule. The pole expansion for computing the required quantities in (5.4) took 4400 seconds. This time encompasses solving 15 sets of systems, each with 200 shifts of the form $\varepsilon_i + i\eta_l$, and each ε_i corresponds to a distinct right-hand side. Computing the 15 smallest eigenvectors took 46.86 seconds. Conversely, to use the alternative formula in (5.4) for computing $\hat{\chi}_0(i\omega)[\hat{g}](\mathbf{r})$ requires computing a large number of additional eigenvectors. Even just computing the first 1000 eigenvectors took 6563 seconds using LOBPCG. Computing the first 2000 eigenvectors took 82448 seconds. Figure 7(c) shows an example of the solution in a slice of the domain for $\omega = 0$.

6.4. Nonorthogonal basis functions: $\boldsymbol{S \neq I}$. We now consider the case where nonorthogonal basis functions are used in the formulation of the problems as described in section 5. The matrices H and S are obtained by solving the Kohn–Sham density functional theory problem for a DNA molecule with 715 atoms using the SIESTA package [38] using the atomic orbital basis. The atomic configuration of the DNA molecule is given in Figure 8(b). The number of electrons is 2442, and the number of occupied states $N_e = 1221$ (spin degeneracy of 2 is counted here). The 1221 negative eigenvalues and the corresponding eigenvectors were directly computed and were only used to ensure that the right-hand sides for the set of equations we solve are orthogonal to the negative eigenspace of the (H, S) pencil.

In this case, the basis functions are nonorthogonal, and therefore $S \neq I$. This means that we are now interested in solving problems of the form

$$(H - i\eta_l S)x_l = b \tag{6.3}$$

for a large number of equispaced η_l in the interval $[-10, 10]$. Furthermore, we are interested in solving the system for the same set of η_l for multiple right-hand sides b. In this case, both H and S are sparse and of size 7752×7752. Therefore, the problem lends itself to the use of a direct method when solving the subproblems for the pole expansion method as discussed in 3. Since once again the problem is real, this means that we have to compute $P/2$ LU factorizations of matrices of the form

$$(H - \xi_k S). \tag{6.4}$$

After this step has been completed, the pole expansion method may be used to quickly solve problems of the form (6.3). First, we may compute the set of weights for each η_l. Then, we may use the LU factorizations of (6.4) to find the vectors $\tilde{h}_k$. Finally, we may compute the solutions to the set of problems (6.3) for a fixed b using the computed weights. For each additional right-hand side, we just need to compute a new set of $\tilde{h}_k$ and then combine them using the same weights as before. The marginal cost for each additional right-hand side is $P/2$ forward and backward substitutions plus the cost of combining P vectors with N_z distinct sets of weights. Comparatively, using the Lanczos method to solve for multiple right-hand sides requires starting over because the Krylov subspace is dependent on b. It is important to note that if the number of shifts is less than the number of poles used, it would be more efficient to just factor the N_z shifted systems. However, we are interested in the case where there are more shifts than poles even though some cases in the example do not reflect this situation.

This solution strategy is not dependent on computing LU factorizations of (6.4). Any direct method may be used that allows for rapid computation of solutions given a new right-hand side. Therefore, the pole expansion algorithm used in this manner has two distinct parts. There is the factorization step, where P factorizations are computed, and there is the solving step, where the factorizations are used to solve P subproblems for each right-hand side and then the solutions are combined for all the desired η_l. For a small number of shifts and for very few right-hand sides, we still expect the Lanczos method to potentially outperform the pole expansion method. However, as soon as we have a large number of shifts or there are enough right-hand sides to make the use of the direct methods favorable to an iterative method, we expect the pole expansion method to take much less time than the Lanczos method.

Based on the splitting of the work for the pole expansion between a factorization step and a solve step, we present the results for this example slightly differently than before. The problems were solved for a varying number of η_l, denoted N_z. For each set of η_l, the problem was solved using both the pole expansion method and the Lanczos method for N_{rhs} distinct right-hand sides via the transformation in (2.1), and we report the average time to solve the problem for a single right-hand side along with the time for computing the Cholesky factorization of S, denoted T_c. Table 2 shows the results of using the Lanczos method for solving the set of problems. We observe that the number of iterations is consistent regardless of the number of shifts, and because the problem is small, the method scales well as the number of shifts grows. However, for each right-hand side, the Lanczos method must start from scratch.

Table 2
Lanczos method for solving $(H - i\eta_l S)x_l = b$.

		Lanczos			
N_z	N_{rhs}	Avg. iter.	$T_c(s)$	Avg. solve time(s) per b	Max error
3	20	171	1.33	102	3.88 $\times 10^{-9}$
11	20	173	1.33	103.8	4.03 $\times 10^{-9}$
101	20	173	1.33	114.8	4.25 $\times 10^{-9}$
1001	20	171	1.33	223.6	4.94 $\times 10^{-9}$

Table 3
Pole expansion method for solving $(H - i\eta_l S)x_l = b$.

		Pole expansion			
N_z	N_{rhs}	P	Factor time(s)	Avg. solve rime(s) per b	Max error
3	20	60	149.4	3.95	5.52 $\times 10^{-10}$
11	20	60	149.4	3.98	5.46 $\times 10^{-10}$
101	20	60	149.4	5.14	5.45 $\times 10^{-10}$
1001	20	60	149.4	16.2	5.56 $\times 10^{-10}$

Table 3 shows the time taken to factor the $P/2$ subproblems of the pole expansion method and then the cost for computing a solution for all the η_l per right-hand side. Similar to before, we did not take advantage of the parallelism in the method and simply computed the factorizations sequentially. Here, we observe that the bulk of the computation time is in the factorization step, which is expected since this now behaves like a direct method. However, once the factorizations have been computed, the marginal cost of forming the solutions for all of the desired shifts with a new right-hand side is minimal.

When a direct method is an option for solving problems of the form (6.3) and there are a large number of shifts, the direct method may be combined with the pole expansion method to essentially parametrize the factorizations of N_z distinct matrices on the factorizations of P distinct matrices. If the number of shifts is much larger than the number of poles required for the desired accuracy, this reduction in the number of required factorizations turns out to be very beneficial computationally. Furthermore, if memory is an issue, it is possible to only ever store one factorization at a time. Specifically, once a factorization is computed for a given pole, the vectors $\widetilde{h}_k$ may be computed for each right-hand side and then the factorization may be discarded. Overall, the combined use of the pole expansion and an efficient direct method appears to be a very efficient method for solving sets of parametrized linear systems for multiple right-hand sides, or, in some cases, even for a single right-hand side.

7. Conclusion.

We have presented a new method for efficiently solving a type of parametrized linear systems, motivated from electronic structure calculations. By building a quadrature scheme based on the ideas in [19], we are able to represent the solutions of the parametrized shifted systems as weighted linear combinations of solutions to a set of fixed problems, where the weights vary based on the parameter of the system. This method scales well as the number of distinct parameters for which we want to solve (1.1) grows. Furthermore, because the solutions to the parametrized equations are based on a fixed set of subproblems, there is flexibility in how the subproblems are solved. We presented examples using both iterative and direct solvers within the framework for solving the shifted systems. The method presented here can be more favorable compared to a Lanczos based method, especially when solutions to a large number of parameters or a large number of right-hand sides are required.

Acknowledgments. We would like to thank Lenya Ryzhik for providing computing resources. We are grateful to Alberto Garcia and Georg Huhs for providing the atomic configuration for the DNA molecule. L. L. thanks Dario Rocca and Chao Yang for helpful discussions, and thanks the hospitality of Stanford University, where the idea of this paper started.

REFERENCES

[1] S. L. Adler, *Quantum theory of the dielectric constant in real solids*, Phys. Rev., 126 (1962), pp. 413–420.

[2] F. Aryasetiawan and O. Gunnarsson, *The GW method*, Rep. Progr. Phys., 61 (1998), p. 237.

[3] Z. Bai and R. W. Freund, *A partial Padé-via-Lanczos method for reduced-order modeling*, Linear Algebra Appl., 332 (2001), pp. 139–164.

[4] S. Baroni, S. de Gironcoli, A. Dal Corso, and P. Giannozzi, *Phonons and related crystal properties from density-functional perturbation theory*, Rev. Modern Phys., 73 (2001), pp. 515–562.

[5] S. Baroni, P. Giannozzi, and A. Testa, *Green's-function approach to linear response in solids*, Phys. Rev. Lett., 58 (1987), pp. 1861–1864.

[6] B. N. Datta and Y. Saad, *Arnoldi methods for large Sylvester-like observer matrix equations, and an associated algorithm for partial spectrum assignment*, Linear Algebra Appl., 154 (1991), pp. 225–244.

[7] T. A. Driscoll, *Algorithm* 756: *A MATLAB toolbox for Schwarz–Christoffel mapping*, ACM Trans. Math. Software, 22 (1996), pp. 168–186.

[8] T. A. Driscoll, *Algorithm* 843: *Improvements to the Schwarz-Christoffel toolbox for MATLAB*, ACM Trans. Math. Software, 31 (2005), pp. 239–251.

[9] P. Feldmann and R. W. Freund, *Efficient linear circuit analysis by Padé approximation via the Lanczos process*, IEEE Trans. Comput. Aid D., 14 (1995), pp. 639–649.

[10] R. Freund, *A transpose-free quasi-minimal residual algorithm for non-Hermitian linear systems*, SIAM J. Sci. Comput., 14 (1993), pp. 470–482.

[11] C. Friedrich and A. Schindlmayr, *Many-body perturbation theory: The GW approximation*, NIC Series, 31 (2006), p. 335.

[12] A. Frommer, *BiCGStab(l) for families of shifted linear systems*, Computing, 70 (2003), pp. 87–109.

[13] A. Frommer and U. Glässner, *Restarted GMRES for shifted linear systems*, SIAM J. Sci. Comput., 19 (1998), pp. 15–26.

[14] F. Furche, *Molecular tests of the random phase approximation to the exchange-correlation energy functional*, Phys. Rev. B, 64 (2001), 195120.

[15] K. Gallivan, G. Grimme, and P. Van Dooren, *A rational Lanczos algorithm for model reduction*, Numer. Algorithms, 12 (1996), pp. 33–63.

[16] F. Giustino, M. L. Cohen, and S. G. Louie, *GW method with the self-consistent sternheimer equation*, Phys. Rev. B, 81 (2010), 115105.

[17] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., Johns Hopkins University Press, Baltimore, MD, 1996.

[18] X. Gonze, D. C. Allan, and M. P. Teter, *Dielectric tensor, effective charges, and phonons in α-quartz by variational density-functional perturbation theory*, Phys. Rev. Lett., 68 (1992), p. 3603.

[19] N. Hale, N. J. Higham, and L. N. Trefethen, *Computing A^{α}, $\log(A)$, and related matrix functions by contour integrals*, SIAM J. Numer. Anal., 46 (2008), pp. 2505–2523.

[20] L. Hedin, *New method for calculating the one-particle Green's function with application to the electron-gas problem*, Phys. Rev., 139 (1965), p. A796.

[21] P. Hohenberg and W. Kohn, *Inhomogeneous electron gas*, Phys. Rev., 136 (1964), pp. B864–B871.

[22] A. V. Knyazev, *Toward the optimal preconditioned eigensolver: Locally optimal block preconditioned conjugate gradient method*, SIAM J. Sci. Comput., 23 (2001), p. 517.

[23] W. Kohn and L. Sham, *Self-consistent equations including exchange and correlation effects*, Phys. Rev., 140 (1965), pp. A1133–A1138.

[24] C. Lanczos, *An iteration method for the solution of the eigenvalue problem of linear differential and integral operators*, J. Res. Nat. Bur. Stand., 45 (1950), pp. 255–282.

[25] D. C. Langreth and J. P. Perdew, *The exchange-correlation energy of a metallic surface*, Solid State Commun., 17 (1975), pp. 1425–1429.

[26] D. C. Langreth and J. P. Perdew, *Exchange-correlation energy of a metallic surface: Wavevector analysis*, Phys. Rev. B, 15 (1977), 2884.

[27] L. Lin, J. Lu, L. Ying, and W. E, *Pole-based approximation of the Fermi–Dirac function*, Chin. Ann. Math. Ser. B, 30B (2009), p. 729.

[28] R. Martin, *Electronic Structure—Basic Theory and Practical Methods*, Cambridge University Press, West Nyack, NY, 2004.

[29] P. G. Martinsson and V. Rokhlin, *A fast direct solver for boundary integral equations in two dimensions*, J. Comput. Phys., 205 (2005), pp. 1–23.

[30] K. Meerbergen, *The solution of parametrized symmetric linear systems*, SIAM J. Matrix Anal. Appl., 24 (2003), pp. 1038–1059.

[31] H.-V. Nguyen and S. de Gironcoli, *Efficient calculation of exact exchange and RPA correlation energies in the adiabatic-connection fluctuation-dissipation theory*, Phys. Rev. B, 79 (2009), 205114.

[32] C. Paige and M. Saunders, *Solution of sparse indefinite systems of linear equations*, SIAM J. Numer. Anal., 12 (1975), pp. 617–629.

[33] M. Parks, E. de Sturler, G. Mackey, D. Johnson, and S. Maiti, *Recycling Krylov subspaces for sequences of linear systems*, SIAM J. Sci. Comput., 28 (2006), pp. 1651–1674.

[34] Y. Ping, D. Rocca, and G. Galli, *Electronic excitations in light absorbers for photoelectrochemical energy conversion: First principles calculations based on many body perturbation theory*, Chem. Soc. Rev., 42 (2013), pp. 2437–2469.

[35] Y. Saad and M. Schultz, *Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Stat. Comput., 7 (1986), pp. 856–869.

[36] A. Saibaba, T. Bakhos, and P. Kitanidis, *A flexible Krylov solver for shifted systems with application to oscillatory hydraulic tomography*, SIAM J. Sci. Comput., 35 (2013), pp. A3001–A3023.

[37] V. Simoncini and D. B. Szyld, *Recent computational developments in Krylov subspace methods for linear systems*, Numer. Linear Algebra Appl., 14 (2007), pp. 1–59.

[38] J. M. Soler, E. Artacho, J. D. Gale, A. García, J. Junquera, P. Ordejón, and D. Sánchez-Portal, *The SIESTA method for ab initio order-N materials simulation*, J. Phys. Condens. Matter, 14 (2002), pp. 2745–2779.

[39] P. Umari, G. Stenuit, and S. Baroni, *GW quasiparticle spectra from occupied states only*, Phys. Rev. B, 81 (2010), 115104.

[40] N. Wiser, *Dielectric constant with local field effects included*, Phys. Rev., 129 (1963), pp. 62–69.

[41] C. Yang, J. C. Meza, B. Lee, and L. W. Wang, *KSSOLV—A MATLAB toolbox for solving the Kohn–Sham equations*, ACM Trans. Math. Software, 36 (2009), p. 10.