Appendix.

A. Definite integrals. For the convenience of the reader, we provide the definite integrals required to compute the residual level function $F(\gamma, \omega)$ from Equation (3.15), the gradient ∇F from Equations (3.19) and (3.20), and the approximation of the Hessian, given in Equation (4.8). For the sake of generality we denote poles as ω_1 and ω_2, without conjugation although some formulation require the conjugation or sign change of ω_1 or ω_2. Recall that the functions $\mathfrak{G}(t)$ and $h(t)$ are piecewise constant and determine the integral boundaries, written as a and b.

The calculation of $W_{k,\ell}$, $\overline{W}^{\Pi}_{k,\ell}$, $X_{k,\ell}$, $\overline{X}^{\Pi}_{k,\ell}$, $Y_{k,\ell}$, and $Z_{k,\ell}$ requires the following integral:

$$\int_a^b \frac{dt}{(t-\omega_1)(t-\omega_2)} = \qquad \frac{1}{\omega_1-\omega_2}\left[\ln\frac{(b-\omega_1)}{(a-\omega_1)} + \ln\frac{(a-\omega_2)}{(b-\omega_2)}\right]$$

For the 'diagonal' terms $W_{k,k}$ and $\overline{X}^{\Pi}_{k,k}$ it holds than $\omega_1 = \omega_2$. In this case we can re-write the expression without logarithms. This integral is also required for $\nabla\overline{\theta}_k$:

$$\int_a^b \frac{dt}{(t-\omega_1)^2} = \frac{b-a}{(b-\omega_1)(a-\omega_1)}$$

Note that this is not the case for the X, Z, and Y, where the signs and/or conjugations of ω_1 and ω_2 differ.

For θ_k and $\overline{\theta}^{\Pi}_k$ we require the following integral:

$$\int_a^b \frac{dt}{t-\omega_1} = \ln\frac{(b-\omega_1)}{(a-\omega_1)}$$

The calculation of $\nabla\overline{W}_{k,\ell}$, $\nabla X_{k,\ell}$, $\nabla\overline{Y}_{k,\ell}$, and $\nabla Z_{k,\ell}$ requires the integral:

$$\int_a^b \frac{dt}{(t-\omega_1)^2(t-\omega_2)} = \frac{b-a}{(\omega_1-\omega_2)(b-\omega_1)(a-\omega_1)} - \frac{\ln\frac{(b-\omega_1)}{(a-\omega_1)}}{(\omega_1-\omega_2)^2} + \frac{\ln\frac{(b-\omega_2)}{(a-\omega_2)}}{(\omega_1-\omega_2)^2}$$

The 'diagonal' term $\nabla\overline{W}_{k,k}$ can again be expressed without logarithm:

$$\int_a^b \frac{dt}{(t-\omega_1)^3} = (b-a)\left[\frac{b+a-2\omega_1}{(b-\omega_1)^2(a-\omega_1)^2}\right]$$

In the approximation of the Hessian the terms $\nabla\nabla\overline{W}_{i,j}$, $\overline{\nabla}\nabla X_{i,j}$, $\nabla\nabla\overline{Y}_{i,j}$, and $\overline{\nabla}\nabla Z_{i,j}$ require yet another integral. This one is given by:

$$\begin{aligned}\int_a^b \frac{dt}{(t-\omega_1)^2(t-\omega_2)^2} = {} & \frac{a-b}{(\omega_1-\omega_2)^2}\left[\frac{1}{(a-\omega_1)(-b+\omega_1)} + \frac{1}{(a-\omega_2)(-b+\omega_2)}\right] \\ & + \frac{2}{(\omega_1-\omega_2)^3}\left[\ln\frac{a-\omega_1}{b-\omega_1} + \ln\frac{b-\omega_2}{a-\omega_2}\right]\end{aligned}$$

Here the 'diagonal' term is given by:

$$\int_a^b \frac{dt}{(t-\omega_1)^4} = \frac{1}{3}\left[\frac{1}{(a-\omega_1)^3} + \frac{1}{(-b+\omega_1)^3}\right]$$

B. Obtaining benchmark problems from a matrix. In this short appendix we discuss a method to generate a large set of intervals $\mathcal{I}$ from a given eigenproblem corresponding to an Hermitian matrix M. The objective is to obtain a benchmark test set for filtered subspace iteration, as a means to compare filters without having to solve each problem separately.

Interior eigensolvers based on filtered subspace iteration are sensitive to the spectrum of M near the search interval $\mathcal{I}$. Thus, the choice of the search intervals is critical for testing an interior

eigensolver. Users often select the search interval motivated by some structure in the spectrum of M, such as the HOMO-LUMO gap in Density Functional Theory methods. Our aim is to create a large set of intervals that represent a variety of real-world use-cases, especially those based on the selection of intervals influenced by the structure of the spectrum. To this end, we propose a method to obtain search intervals by exploiting the distribution of the eigenvalues of M. Instead of choosing the intervals directly, we compute a set of endpoints $E = \{a_1, a_2, ...\}$, and then form intervals $\mathcal{I}_{i,j} = [a_i, a_j]$ with $a_i, a_j \in E$ and $a_i < a_j$.

These endpoints should represent the neighborhood of an identifiable spectral feature, such as a spectral gap or a cluster; we refer to them as "feature points". Given a function $\phi \in C^2$ that approximates the eigenvalue density, we consider feature points to be the real zeros of the first and second derivative of ϕ. A good candidate for ϕ can be obtained, for example, via the Kernel Polynomial Method (KPM) which constructs a polynomial of degree m that approximates the spectral density. The zeros of ϕ' are stationary points; assuming that ϕ is a good approximation of the eigenvalue density, a local maximum may indicate a high density of eigenvalues or even a cluster. Conversely, a local minimum corresponds to a potential gap in the spectrum. Inflection points of ϕ may indicate a change in the increase or decrease of the (approximated) eigenvalue density, signaling a relatively small spectral gap. By choosing a large $m \approx 45$ it is possible to obtain thousands of intervals for a single matrix M. The large problem set can be used to obtain the subspace convergence rates and the condition numbers of the linear system solves without solving each eigenproblem in the benchmark set. Such an approach is limited by the availability of the entire spectrum of M, and thus is only applicable for small to medium sized eigenproblems.

For instance, the subspace convergence rate (see Equation (2.4)) depends only on the filter value for the eigenvalues. If the entire spectrum of M is available, we can obtain the convergence rate by computing these values directly. In the same fashion, each filtered interval results in a number of shifted linear system solves of the form $M - Iz_i$ (see Equation (2.1)). When Krylov-based methods are used to solve these systems, the overall performance depends on the condition number. Since M is Hermitian, the shifted matrix is still normal. As a result the singular values of $M - Iz_i$ are given by $\sigma_i = |\lambda_i - z_i|$, where λ_i is the i^{th} eigenvalue of M. By proceeding in this manner for all the poles z_i, we can inexpensively obtain the condition number for all the system solves. Usually, the solve with the highest condition number dictates the overall performance, which is what we used to motivate our analysis of box constraints in Section 4.4.2.

C. List of filters. The filters presented in Section 2 are provided here. For a given filter function

$$f(t, w, \gamma) = \sum_{k=1}^{q} \left[\frac{\gamma_k}{t - w_k} + \frac{\overline{\gamma}_k}{t - \overline{w}_k} - \frac{\gamma_k}{t + w_k} - \frac{\overline{\gamma}_k}{t + \overline{w}_k} \right]$$

we provide only the poles in the "first quadrant" of the complex plane—that is, the poles with positive real and imaginary parts—and the corresponding coefficients γ.

$\lvert t\rvert \in$	$[0, .95)$	$[.95, 1.05)$	$[1.05, 1.4)$	$[1.4, 5)$	$[5, \infty)$
$\mathfrak{G}_{\gamma\text{-SLiSe}}(t)$	1	.01	10	20	0

$\lvert t\rvert \in$	$[0, .95)$	$[.95, .995)$	$[.995, 1.005)$	$[1.005, 1.05)$	$[1.05, 1.1)$
$\mathfrak{G}_{\eta\text{-SLiSe}}(t)$	1	4	0.5	4	0.6
$\mathfrak{G}_{\text{Box-SLiSe}}(t)$	1	4	2	4	0.6

$\lvert t\rvert \in$	$[1.1, 1.3)$	$[1.3, 1.8)$	$[1.8, 3)$	$[3, \infty)$
$\mathfrak{G}_{\eta\text{-SLiSe}}(t)$ (cont'd)	1	0.3	0.1	0
$\mathfrak{G}_{\text{Box-SLiSe}}(t)$ (cont'd)	1	0.3	0.1	0

Table C.1: Weight functions for filters discussed in this paper.

Poles ω	Coefficients γ
$0.999687712591797 + 0.0117367635577924i$	$0.006050012497458 - 0.000227036554136i$
$0.991596517222374 + 0.093208856178882i$	$0.021484299350510 - 0.003666847993474i$
$0.903848148311606 + 0.327740045699974i$	$0.055938387061383 - 0.032384567818443i$
$0.440319857798568 + 0.732970137475905i$	$0.054079510922005 - 0.122837701171251i$

Table C.2: γ-SLiSe, a filter meant to replace the Gauss filter

Poles ω	Coefficients γ
$0.999986323489133 + 0.002453510792541i$	$0.00110213725846833 - 7.98515806042 \cdot 10^{-6}i$
$0.999401600189637 + 0.024159213959740i$	$0.00768847889630669 - 0.000333793441122i$
$0.983469964312691 + 0.160816338804574i$	$0.04592568742949140 - 0.008858565519222i$
$0.628559997051189 + 0.718255201795431i$	$0.11139119375850727 - 0.147357486573937i$

Table C.3: η-SLiSe, a filter meant to replace the Elliptic filter. Obtained with a penalty parameter of $c = -1.3 \cdot 10^{-6}$

Poles ω	Coefficients γ
$0.999517437449349 + 0.0011346403206723i$	$0.000600799688893 - 0.0001380523176106i$
$0.996122208058289 + 0.0169588203859498i$	$0.006148402177611 - 0.0012349951550185i$
$0.971590779276380 + 0.1314326323772290i$	$0.039288214664051 - 0.0095779819369782i$
$0.632009932807876 + 0.6589465004506030i$	$0.111997418650841 - 0.1454789177402232i$

Table C.4: ζ-SLiSe, a filter meant to replace the Elliptic filter that violates Guideline 1

Poles ω	Coefficients γ
$0.999997864241235 + 0.002199301304944i$	$0.000920693720425 - 2.62225614322 \cdot 10^{-6}i$
$0.999817083735386 + 0.019255687759249i$	$0.006309876577596 - 0.000106451402529i$
$0.993359173161426 + 0.135785991680408i$	$0.040771709455088 - 0.003927802442013i$
$0.694622923894908 + 0.732441949783473i$	$0.135841041422952 - 0.150283641573960i$

Table C.5: κ-SLiSe, a filter meant to replace the Elliptic filter with Krylov solvers that violates Guideline 1. Obtained with a box constraint of `lb` $= 0.0022$