Due Thu June 2.

Here's an integrative project based on a technique developed by Eric Polizzi ("Density-matrix-based algorithm for solving eigenvalue problems", Phys. Rev. B, 115112, 2009).

Goal: Suppose you have a real symmetric matrix **A** that is too big to write, but for which you could solve $\mathbf{Ax} = \mathbf{b}$, e.g., by using multigrid. For such a matrix, finding its eigenvalues and eigenvector with a method like QR is impractical: you can't even create the necessary data structures because of the size of the problem. For this matrix, we wish to find all eigenvalues in a given range $(\lambda_{\min}, \lambda_{\max})$ and the associated eigenvectors.

The technique presented here uses lots of other methods

- An iterative approach like multigrid to solve $\mathbf{Ax} = \mathbf{b}$.
- The QR method for eigenvalues
- Gauss-Legendre quadrature
- Gram-Schmidt orthogonalization

and the theory makes nice use of the Cauchy integral formula!

Some theory

Fact 1: In the complex plane, Cauchy's integral formula says

$$\frac{1}{2\pi i}\oint_C \frac{dz}{z-a} = \begin{cases} 0 & a \notin C \\ 1 & a \in C. \end{cases}$$

That is, if complex point a lies in the closed contour C in the complex plane, the integral is 1. Otherwise, it is zero.

Fact 2: Real symmetric matrix **A** is diagonalizable,

$$\mathbf{A} = \mathbf{X}\mathbf{\Lambda}\mathbf{X}^{-1}$$

and its eigenvectors can be made orthornormal so the decomposition

$$\mathbf{A} = \mathbf{X}\mathbf{\Lambda}\mathbf{X}^{T}$$

exists. $\mathbf{\Lambda}$ is diagonal and real, and **X** is $N \times N$ and real.

The matrix function

$$\mathbf{F}(\mu) = \mathbf{A} - \mu\mathbf{I} = \mathbf{X}(\mathbf{\Lambda} - \mu\mathbf{I})\mathbf{X}^T$$

is real and symmetric, and matrix $(\mathbf{\Lambda} - \mu\mathbf{I})$ is diagonal. and

$$\mathbf{F}(\mu)^{-1} = \mathbf{X}\mathrm{diag}\left((\lambda_1 - \mu)^{-1},\ (\lambda_2 - \mu)^{-1},\ ...\right)\mathbf{X}^T$$

Fact 3: Think of $\mathbf{F}^{-1}_{ij}$ as being

$$\mathbf{F}^{-1}_{ij} = \sum_k \mathbf{X}_{ik}(\lambda_k - \mu)^{-1}\mathbf{X}^T_{kj} = \sum_k \mathbf{X}_{ik}(\lambda_k - \mu)^{-1}\mathbf{X}_{jk}$$

or, if column k of matrix $\mathbf{X}$ is eigenvector $\mathbf{x}_k$, then

$$\mathbf{F}^{-1} = \sum_k \mathbf{x}_k(\lambda_k - \mu)^{-1}\mathbf{x}^T_k.$$

Fact 4: Combine facts 1 and 3:

$$-\frac{1}{2\pi i}\int_C d\mu\mathbf{F}(\mu)^{-1} = \sum_{k|\lambda_k\in C} \mathbf{x}_k\mathbf{x}^T_k.$$

Fact 5: The matrix $\sum_{\sum_{k|\lambda_k\in C}\mathbf{x}_k\mathbf{x}^T_k}$ has rank equal to the number of eigenvalues in the contour C. Suppose that number was M. Then, if $\mathbf{Y}$ were an $N\times M$ random matrix,

$$\mathbf{Q} = -\frac{1}{2\pi i}\int_C d\mu\mathbf{F}(\mu)^{-1}\mathbf{Y}$$

would create an $N\times M$ matrix $\mathbf{Q}$ in the column space of the M eigenvectors of $\mathbf{A}$ associated with eigenvalues in the contour C.

Fact 6: Symmetry can be used to express the contour integral as an integral over only half the contour:

$$\mathbf{Q} = -\frac{1}{2\pi i}\oint d\mu \mathbf{F}(\mu)^{-1}\mathbf{Y}$$
$$\mu = \bar{\lambda} + re^{i\theta}$$
$$\mathbf{Q} = -\frac{1}{2\pi i}\int_{-\pi}^{+\pi} d\theta rie^{i\theta}\mathbf{F}(\bar{\lambda} + re^{i\theta})^{-1}\mathbf{Y}$$
$$\mathbf{Q} = -\frac{1}{2\pi i}\int_{0}^{+\pi} d\theta rie^{i\theta}\mathbf{F}(\bar{\lambda} + re^{i\theta})^{-1}\mathbf{Y} - \underbrace{\frac{1}{2\pi i}\int_{-\pi}^{0} d\theta rie^{i\theta}\mathbf{F}(\bar{\lambda} + re^{i\theta})^{-1}\mathbf{Y}}_{\text{let } \theta=-\phi}$$
$$= -\frac{1}{2\pi i}\int_{0}^{\pi} d\theta rie^{i\theta}\mathbf{F}(\bar{\lambda} + re^{i\theta})^{-1}\mathbf{Y} + \frac{1}{2\pi i}\int_{0}^{\pi} d\phi r(-i)e^{-i\phi}\mathbf{F}(\bar{\lambda} + re^{-i\phi})^{-1}\mathbf{Y}$$

the integrands are complex conjugates of one another, so, we have for example $[\Re(z) + i\Im(z)] - [\Re(z) - i\Im(z)] = 2i\Im(z)$ where z is the integrand:

$$\mathbf{Q} = -\frac{1}{\pi}\int_{0}^{\pi} d\theta \Im\left[rie^{i\theta}\mathbf{F}(\bar{\lambda} + re^{i\theta})^{-1}\mathbf{Y}\right]$$

Fact 7: Recall Gauss-Legendre quadrature:

$$\int_{-1}^{+1} f(s)ds = \sum_{j=1}^{n} w_j f(x_j).$$

Then, change variables in from $\theta \in [0, \pi]$ to $s \in [-1, 1]$ via $\theta = (\pi/2)[s + 1]$,

$$\mathbf{Q} = -\frac{1}{2}\int_{0}^{\pi} ds \Im\left[rie^{i\pi(s+1)/2}\mathbf{F}(\bar{\lambda} + re^{i\pi(s+1)/2})^{-1}\mathbf{Y}\right]$$
$$= \frac{1}{2}\int_{0}^{\pi} ds \Im\left[re^{i\pi s/2}\mathbf{F}(\bar{\lambda} + rie^{i\pi s/2})^{-1}\mathbf{Y}\right]$$
$$= \frac{1}{2}\sum_{j=1}^{n} w_j \Im\left[re^{i\pi s_j/2}\mathbf{F}(\bar{\lambda} + rie^{i\pi s_j/2})^{-1}\mathbf{Y}\right]$$

This will involve complex math, but the final result should be real.

Fact 8: If $\mathbf{z}$ is an eigenvector of $\mathbf{Q}^T\mathbf{A}\mathbf{Q}$, then $\mathbf{x} = \mathbf{Q}\mathbf{z}$ is an eigenvector of $\mathbf{A}$.

An algorithm

With this in mind, here's an algorithm to accomplish the stated goal.

- Given $\lambda_{\min}$ and $\lambda_{\max}$ calculate $\bar{\lambda}$ and r.
- Make $\mathbf{Q}$:
  - choose a random $\mathbf{y}$.
  - for each quadrature point $j = 1, ..., n$, solve $\mathbf{F}(\bar{\lambda} + rie^{i\pi x_j/2})$ solve $\mathbf{F}\mathbf{p}_j = \mathbf{y}$ for $\mathbf{p}_j$. If $\mathbf{A}$, hence $\mathbf{F}$, is titanic, then a technique like multigrid would be used that does not require $N \times N$ storage.
  - $\mathbf{q} = \frac{r}{2} \sum_j w_j \Im \left(\mathbf{p}_j e^{i\pi x_j/2} \right)$
  - make the new $\mathbf{q}$ orthogonal to the already-determined columns of $\mathbf{Q}$ using Gram-Schmidt orthogonalization.
  - when no new $\mathbf{q}$ can be found, the number of columns of $\mathbf{Q}$ is M – the number of eigenvalues in the range.
- Make $\mathbf{A}_Q = \mathbf{Q}^T\mathbf{A}\mathbf{Q}$, a matrix of size $N \times N$. Note: this does not require ever writing $\mathbf{A}$. One need only be able to evaluate the matrix-vector product $\mathbf{A}\mathbf{q}$.
- Apply QR to find the eigenvalues and eigenvectors of $\mathbf{A}_Q$.
- The eigenvectors of $\mathbf{A}_Q$ can be converted to eigenvectors of $\mathbf{A}$.

The homework questions

(1) Polizzi does not use Gram-Schmidt orthogonalization. How could one make this work with the pseudoinverse $\mathbf{Q}^\dagger$? Note: the issue here is not just the similarity transform $\mathbf{Q}^T\mathbf{A}\mathbf{Q}$, but also the determination of M.

(2) Show how this works by calculating the two eigenvalues and eigenvectors of

$$\mathbf{A} = \begin{pmatrix} -2 & 1 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 0 & 1 & -2 & 1 \\ 0 & 0 & 1 & -2 \end{pmatrix}$$

that lie in $[-2, 0]$. Note: I expect this to be a pencil and paper exercise, with matlab assistance (for example). No need to write an elaborate code.