1 Introduction

Let us recall that for given symmetric A, $B \in \mathbb{R}^{n\times n}$ and B positive definite, the ***Rayleigh Quotient*** for the matrix pencil $A - \lambda B$ is defined by

$$\rho(x) = \frac{x^T Ax}{x^T Bx}. \tag{1.1}$$

Denote the eigenvalues of $A - \lambda B$ by λ_1, λ_2, $\ldots$, λ_n in ascending order, i.e.,

$$\lambda_1 \le \lambda_2 \le \cdots \le \lambda_n,$$

and their associated eigenvectors by

$$u_1,\, u_2,\, \ldots,\, u_n, \text{ respectively, and } \|u_i\|_B = 1 \text{ for } i = 1, 2, \ldots, n,$$

where $\|\cdot\|_B$ is defined through the B-inner product

$$\langle x, y\rangle_B \stackrel{\text{def}}{=} \langle Bx, y\rangle \equiv y^T Bx.$$

It is known that

$$\lambda_1 = \min_x \rho(x), \quad u_1 = \operatorname*{argmin}_x \rho(x), \tag{1.2}$$

and in general for $i > 1$,

$$\lambda_i = \min_{x\perp_B u_j,\, 1\le j<i} \rho(x), \quad u_i = \operatorname*{argmin}_{x\perp_B u_j,\, 1\le j<i} \rho(x), \tag{1.3}$$

where by $x\perp_B y$ we mean that x and y are B-orthogonal, i.e., $\langle x, y\rangle_B = 0$. Therefore naturally various optimization techniques can be and have been employed to compute λ_1 or the first few λ_i and their associated eigenvectors.

By replacing A by $-A$, expressions similar to (1.2) and (1.3) can be obtained for λ_n and the last few eigenvalues. These expressions enable the use of optimization techniques for the computation of λ_n or the last few λ_i and their associated eigenvectors. Therefore in what follows, we shall simply focus on the first few eigenvalues and their associated eigenvectors.

Methods in this lecture are based on minimizing the Rayleigh Quotient $\rho(x)$. Two useful quantities for this purpose are the gradient $g(x) = \nabla\rho(x)$ and Hessian $H(x)$ of $\rho(x)$:

$$g(x) = \frac{2}{x^T Bx}[Ax - \rho(x)\, Bx], \tag{1.4}$$

$$H(x) = \frac{2}{x^T Bx}[A - \rho(x)\, B - g(x)(Bx)^T - (Bx)g(x)^T]. \tag{1.5}$$

In minimizing the Rayleigh quotient along the direction of $g(x)$, the scalar $2/x^T Bx$ does not matter, that is that the minimization is equivalently to seek optimal solution along the residual vector

$$r(x) = Ax - \rho(x)\, Bx. \tag{1.6}$$

Observe that $x^T r(x) = 0$.

Much of the development in this lecture evolves around $\inf_t \rho(x + tp)$ to seek a better approximation $x + tp$ to the desired eigenvector and correspondingly a better approximation $\rho(x+tp)$ to the desired eigenvalue, where p is a search direction. So we single it out by devoting an entire section to $\inf_t \rho(x + tp)$ first. In the classical steepest descent method, p is simply taken to be $r(x)$.

Throughout this lecture, all notation assignments in this introduction remain valid. In particular, we emphasize that A, $B \in \mathbb{R}^{n\times n}$ are symmetric and B is positive definite.

2 The problem $\inf_t \rho(x + tp)$

In the rest of this lecture,

$$\inf_{t\in\mathbb{R}} \rho(x + tp) \tag{2.1}$$

has to be frequently solved, where p is the searching direction, selected in hoping that $\inf_t \rho(x+tp)$ is strictly less than $\rho(x)$. When it does, some progress is made towards approximating the smallest eigenvalues of $A - \lambda B$. In the case of the steepest decent method, p is simply taken to be the residual vector (1.6), but it may not be, as in the conjugate gradient methods.

Conventionally in the literature, min is used in (2.1), instead of inf. Rigorously speaking, the conventional use of min is not technically correct because there is a possibility that $\inf_t \rho(x+tp)$ may not be attained for any $t \in \mathbb{R}$.

First if p is unfortunately chosen to be collinear to x, then $\rho(x+tp) \equiv \rho(x)$. No improvement is possible as to approximating the smallest eigenvalues of $A - \lambda B$. In the rest of this section, we assume *x and p are linearly independent.*

Suppose x and p are linearly independent. We have

$$\rho(x + tp) \equiv \frac{(x + tp)^T A(x + tp)}{(x + tp)^T B(x + tp)} = \frac{x^T Ax + 2t\, x^T Ap + t^2\, p^T Ap}{x^T Bx + 2t\, x^T Bp + t^2\, p^T Bp}. \tag{2.2}$$

Since $x + tp \neq 0$ for all $t \in \mathbb{R}$ and B is positive definite, $\rho(x + tp)$ is well-defined over the entire $\mathbb{R}$. Therefore $\inf_t \rho(x + tp)$ is equal to the minimum of the values of $\rho(x + tp)$ evaluated at its critical points (where the derivative of $\rho(x + tp)$ with respect to t vanishes) and

$$\lim_{t\to\infty} \rho(x + tp) = \rho(p). \tag{2.3}$$

Through sketching the graph of $\rho(x + tp)$, one can easily conclude

Lemma 2.1.

1. *If $\rho(x) \neq \rho(p)$, then $\rho(x + tp)$ has at least one critical point;*
2. *If $\rho(x) = \rho(p)$, then either $\rho(x + tp) \equiv \rho(x)$ or it has at least two critical points with at least one positive and at least one negative.*

The derivative of $\rho(x + tp)$ is

$$\frac{d}{dt}\rho(x + tp) = \frac{a\, t^2 + b\, t + c}{\left[(x + tp)^T B(x + tp)\right]^2}, \tag{2.4}$$

where

$$a = (p^T Ap)\,(x^T Bp) - (x^T Ap)\,(p^T Bp), \tag{2.5a}$$
$$b = (p^T Ap)\,(x^T Bx) - (x^T Ax)(p^T Bp), \tag{2.5b}$$
$$c = (x^T Ap)(x^T Bx) - (x^T Ax)(x^T Bp), \tag{2.5c}$$

more usefully

$$a = [x^T\, r(p)](p^T Bp) \tag{2.5a$'$}$$
$$= \left(p^T[Ax - \rho(p)\, Bx]\right)\,(p^T Bp), \tag{2.5a$''$}$$
$$b = [\rho(p) - \rho(x)](p^T Bp)(x^T Bx), \tag{2.5b$'$}$$
$$c = [r(x)^T p](x^T Bx). \tag{2.5c$'$}$$

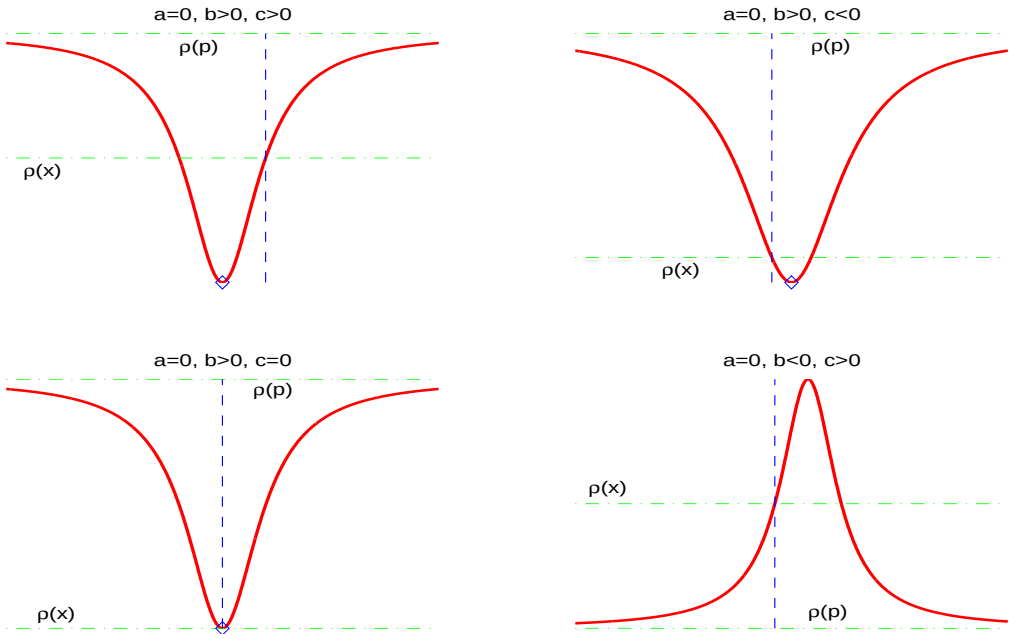

Figure 2.1: $\rho(x+tp)$ for the case $a=0$ with four subcases: two horizonal lines are for $\rho(x)$ and $\rho(p)$, respectively, as marked, the vertical line is for $t=0$, and the curved one is for $\rho(x+tp)$. The points marked by $\diamond$ are the optimal points for each subcases, except the last plot where no point is marked because the optimal values are not attainable but arbitrarily approached as $t \to \pm\infty$.

Now set $\dfrac{d}{dt}\rho(x+tp)$ to 0 to get

$$a\,t^2 + b\,t + c = 0 \tag{2.6}$$

whose solutions are the critical points of $\rho(x+tp)$. There are at most two critical points for $\rho(x+tp)$. They, if any, must be real by Lemma 2.1 because complex roots come in pair.

Lemma 2.2.

1. *Either $a=b=c=0$ which implies $\rho(x+tp) \equiv \rho(x)$ (the converse is also true), or* (2.6) *has only real solutions.*

2. *If $c \neq 0$, then $a^2+b^2>0$. Namely the case $a=b=0$ but $c \neq 0$ cannot occur.*

Proof. If $a^2+b^2+c^2>0$, then for t sufficiently large

$$\frac{d}{dt}\rho(x+tp) \sim \frac{a\,t^2+b\,t+c}{t^4(p^TBp)}$$

which says $\frac{d}{dt}\rho(x+tp) \not\equiv 0$. So $\rho(x+tp)$ is not a constant function, and (2.6) has only real solutions by Lemma 2.1.

Suppose $a=b=0$ but $c \neq 0$. Then $\frac{d}{dt}\rho(x+tp)$ has the same sign as c for all t, which implies $\rho(x+tp)$ moves further and further away from $\rho(x)$ as t moves away from 0. This contradicts (2.3) since $b=0$ and (2.5b$'$) imply $\rho(p)=\rho(x)$. $\square$

If $a \neq 0$, there are two finite real roots to the quadratic equation (2.6):

$$t_\pm = \frac{-b \pm \sqrt{b^2-4ac}}{2a}. \tag{2.7}$$

Notice that

$$\frac{d}{dt}\rho(x+tp)\bigg|_{t=0} = \frac{2}{x^TBx}\,[r(x)^Tp] \tag{2.8}$$

which has the same sign as c by (2.5c′). We now dissect all possible cases in solving the optimization problem (2.1).

1. Suppose $a = 0$.

   (a) If $b = 0$, then $c = 0$ by Lemma 2.2. $\rho(x + tp) \equiv \rho(x)$.

   (b) If $b \neq 0$, there is only one solution to (2.6): $t = -c/b$. $\frac{d}{dt}\rho(x + tp)$ changes its sign at this point.

      i. If $b > 0$, i.e., $\rho(p) > \rho(x)$, $\frac{d}{dt}\rho(x + tp)$ changes from negative to positive. So $\min_t \rho(x+tp)$ is attained at $t = -c/b$, and $\min_t \rho(x+tp) \leq \rho(x)$ and the equality holds if and only if $c = 0$, i.e., $r(x)^T p = 0$.

      ii. If $b < 0$, i.e., $\rho(p) < \rho(x)$, $\frac{d}{dt}\rho(x + tp)$ changes from positive to negative. So $\inf_t \rho(x + tp) = \rho(p)$, and it cannot be attained by any finite t.

   According to the signs of $b \neq 0$, and c, there are all together 6 subcases. Figure 2.1 presents subplots that shows how $\rho(x+tp)$ behaves for 4 of the 6 subcases. The two missing cases are 1) $a = 0$, $b < 0$, and $c < 0$, and 2) $a = 0$, $b < 0$, and $c = 0$. The curves of $\rho(x + tp)$ for these two cases have the same shape as in the last subplot in Figure 2.1, except the tops are to the left of $t = 0$ and on $t = 0$, respectively.

2. Suppose $a > 0$. $\frac{d}{dt}\rho(x + tp)$ is positive for $|t|$ sufficiently large, which means $\rho(x + tp)$ increases from $\rho(p)$ as t moves away from $-\infty$, and $\rho(x+tp)$ increases to $\rho(p)$ as t moves towards $+\infty$. Therefore both $\sup_t \rho(x + tp)$ and $\inf_t \rho(x + tp)$ are attainable and

$$\max_t \rho(x + tp) > \rho(p) > \min_t \rho(x + tp).$$

   Thus $\rho(x+tp)$ has at least two critical points, and in consideration of (2.6) it has exactly two distinct critical points which are $t_\pm$ as given in (2.7). As t goes from $-\infty$ to ∞, $\rho(x + tp)$ increases from $\rho(p)$ until $t = \min t_\pm$ and then decreases until $t = \max t_\pm$ and then increases again towards $\rho(p)$. Therefore the optimal t for (2.1) is $\max t_\pm = t_+$.

   When $t_+ = 0$ which happens when $b > 0$ and $c = 0$, $\min_t \rho(x + tp) = \rho(x)$ and thus no improvement upon $\rho(x)$ can be made.

3. Suppose $a < 0$. Similar arguments leads to that as t goes from $-\infty$ to ∞, $\rho(x + tp)$ decreases from $\rho(p)$ until $t = \min t_\pm$ and then increases until $t = \max t_\pm$ and then decreases again towards $\rho(p)$. Therefore the optimal t for (2.1) is again $\min t_\pm = t_+$.

   Again when $t_+ = 0$ which happens when $b > 0$ and $c = 0$, $\min_t \rho(x + tp) = \rho(x)$.

Figure 2.2 presents representative plots about the behaviors of $\rho(x+tp)$ according to the signs of a and c for the case $a \neq 0$. The sign of b is not distinguished in the subplots, and its effect on the plot determines the positions of the two horizontal lines for $\rho(x)$ and $\rho(p)$ because b has the same sign as $\rho(p) - \rho(x)$ due to (2.5b′), but not the shape of the curves for $\rho(x + tp)$. The case $c = 0$ differs from the case $c \neq 0$ in that whether one of the extreme points marked by $\diamond$ and $\triangle$ is on the vertical line: $t = 0$.

The above analysis enables us to conclude Lemma 2.3 which is not at all obvious and Theorem 2.1.

Lemma 2.3. *If $a \neq 0$, then $b^2 - 4ac > 0$.*

Theorem 2.1. $\inf_t \rho(x + tp) = \rho(x)$ *if and only if one of the following occurs*

1. $a = b = 0$ *(which necessarily implies $c = 0$);*

2. $b > 0$ *and $c = 0$;*

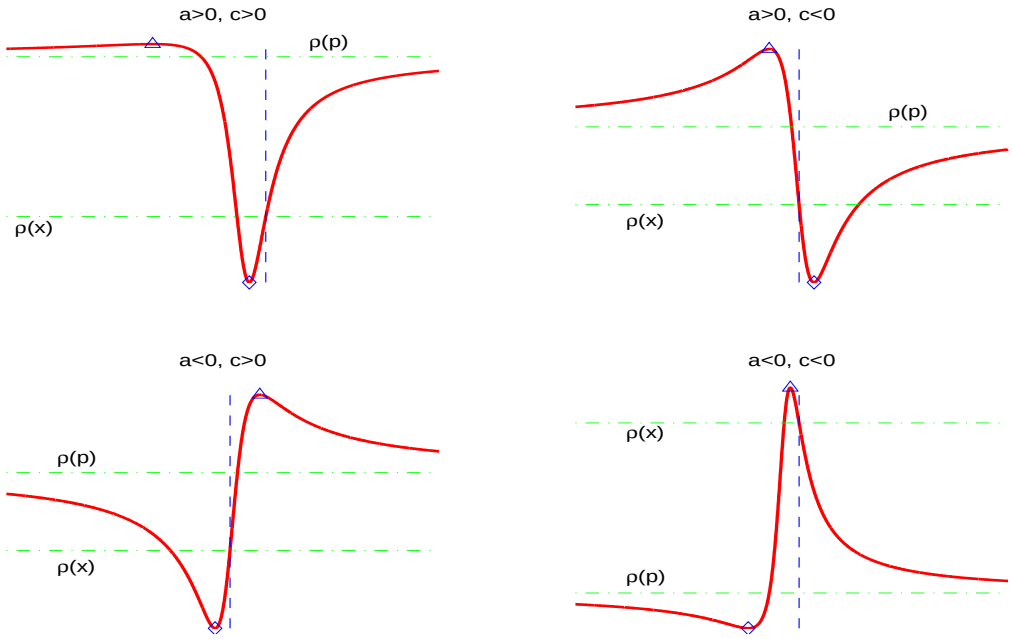

Figure 2.2: $\rho(x+tp)$ for the case $a \neq 0$ with four subcases: two horizonal lines are for $\rho(x)$ and $\rho(p)$, respectively, as marked, the vertical line is for $t = 0$, and the curved one is for $\rho(x+tp)$. The points marked by $\diamond$ are for $\min_t \rho(x+tp)$ while the ones marked by $\triangle$ are for $\max_t \rho(x+tp)$.

If none of the two cases occur, then $\inf_t \rho(x+tp) < \rho(x)$. *Furthermore*

$$\inf_t \rho(x+tp) = \rho(x+t_{\text{opt}}p) \ \textit{at}\ t_{\text{opt}} = \begin{cases} \textit{any value}, & \textit{if } a = b = 0, \\ -c/b, & \textit{if } a = 0,\ b > 0, \\ \infty, & \textit{if } a = 0 \ \textit{and}\ b < 0, \\ t_+, & \textit{if } a \neq 0, \end{cases} \tag{2.9}$$

where $\rho(x+t_{\text{opt}}p)$ *at* $t_{\text{opt}} = \infty$ *is understood as* $\lim_{t\to\infty} \rho(x+tp) = \rho(p)$.

Proof. It can be seen that $\inf_t \rho(x+tp) = \rho(x)$ if and only if either $a = b = 0$ (which necessarily implies $c = 0$, or $a = 0$, $b > 0$, and $c = 0$, or $t_+ = 0$ in the case $a \neq 0$. But $t_+ = 0$ in the case $a \neq 0$ is equivalent to $b > 0$ and $c = 0$. □

Corollary 2.1. $\inf_t \rho(x+tp) = \rho(x)$ *implies* $c = 0$, *i.e.,* $r(x)^T p = 0$.

Define, according to (2.9),

$$y = \begin{cases} x + t_{\text{opt}}\, p, & \text{if } t_{\text{opt}} \neq \infty, \\ p, & \text{otherwise.} \end{cases} \tag{2.10}$$

Similar situations occur later for us to assign a vector $x+tp$ to another vector y with a possibility that t might be infinite. For the ease of presentation, it is understood that $y = x+tp$ is treated differently for $t = \pm\infty$, i.e., simply $y = p$ as in (2.10).

It is hope that y is closer to an eigenvector u_i of $A - \lambda B$ than x is. Typically the closeness of two vectors is measured by the acute angle between them. We see that $\inf_t \rho(x+tp)$ is not always attainable by a finite t_{opt}, for example $t_{\text{opt}} = \infty$ when $a = 0$ and $b < 0$. This is caused by not treating x and p equally as far as the minimization problem is concerned. An equivalent statement of the problem is given in Exercise 2.3.

Exercises

2.1. Prove Lemma 2.1.

2.2. Find an example for which $a = b = c = 0$ and yet x and p are linearly independent, or prove such an example cannot be found.

2.3. Establish the equivalence of the minimization problem (2.1) and the problem

$$y = \xi_{\text{opt}} x + \zeta_{\text{opt}} p, \quad (\xi_{\text{opt}}, \zeta_{\text{opt}}) = \underset{\substack{\xi^2+\zeta^2>0 \\ s,\, t\in\mathbb{R}}}{\operatorname{argmin}} \rho(\xi x + \zeta p) \tag{2.11}$$

by constructing a solution of one from a solution of the other. Rigorously, y by (2.11) is not well-defined because $(\xi_{\text{opt}}, \zeta_{\text{opt}})$ is not unique, but all $\xi_{\text{opt}} x + \zeta_{\text{opt}} p$ are collinear, however. So from the point of view of approximating an eigenvector, any one of them is just as good as others.

2.4. Let $X = [x,\, p] \in \mathbb{R}^{n\times 2}$ and suppose x and p are linearly independent. Let (μ_i, z_i), $i = 1, 2$ be the two eigenpairs of matrix pencil $(X^T AX) - \lambda(X^T BX)$ and $\mu_1 \le \mu_2$.

(a) Express y in (2.11) in terms of X and z_i;

(b) Give an alternative algorithm to solve (2.1).

2.5. Let a, b, and c be defined by (2.5a) – (2.5c). Show that

$$a\,\frac{\|x\|_B}{\|p\|_B} + c\,\frac{\|p\|_B}{\|x\|_B} = b\,\frac{\langle x, p\rangle_B}{\|x\|_B\|p\|_B}. \tag{2.12}$$

Is it necessary to require that x and p be linearly independent for this equality to hold? Use (2.12) to prove that $b^2 - 4ac \ge 0$. Investigate the conditions under which $b^2 - 4ac = 0$.

3 Steepest decent method

3.1 Computing one eigenpair

Given an approximation $\boldsymbol{x}$ to u_1 and $\|\boldsymbol{x}\|_B = 1$, one step of the steepest decent method is simply a line search along the (opposite) direction of the gradient $\nabla\rho(\boldsymbol{x})$, i.e., solve (2.1) with $x = \boldsymbol{x}$ and $p = \boldsymbol{r} \equiv r(\boldsymbol{x})$:

$$\inf_t \rho(\boldsymbol{x} + t\boldsymbol{r}). \tag{3.1}$$

Every development in the previous section holds, because of the choice $p = \boldsymbol{r}$, we now always have

$$c = (\boldsymbol{r}^T\boldsymbol{r})(\boldsymbol{x}^T B\boldsymbol{x}) > 0. \tag{3.2}$$

So $a^2 + b^2 > 0$ always, too. The optimal t_{opt}, according to (2.9) is

$$t_{\text{opt}} = \begin{cases} -c/b, & \text{if } a = 0 \text{ and } b > 0, \\ \infty, & \text{if } a = 0 \text{ and } b < 0, \\ -\frac{b-\sqrt{b^2-4ac}}{2a}, & \text{if } a \ne 0 \text{ and } b \le 0, \\ -\frac{2c}{b+\sqrt{b^2-4ac}}, & \text{if } a \ne 0 \text{ and } b > 0. \end{cases} \tag{3.3}$$

The next approximation to u_1, according to (2.10), is given by $\boldsymbol{y} = \boldsymbol{x} + t_{\text{opt}}\boldsymbol{r}$ with an understanding that $\boldsymbol{y} = \boldsymbol{r}$ when $t_{\text{opt}} = \infty$, and the next approximation $\rho(\boldsymbol{y})$ to λ_1 is less than $\rho(\boldsymbol{x})$ by Theorem 2.1 and by the fact that $c > 0$.

Once $\boldsymbol{y}$ is determined, in actual implementation $\boldsymbol{y}$ is rescaled by, e.g., $\|\boldsymbol{y}\|_B$ and overwrites $\boldsymbol{x}$ and the process is repeated until convergence. A common practice to detect convergence is through checking

$$\text{if} \quad \frac{\|r(\boldsymbol{x})\|}{\|A\boldsymbol{x}\|_2 + |\rho(\boldsymbol{x})|\,\|B\boldsymbol{x}\|_2} < \texttt{rtol}, \tag{3.4}$$

where `rtol` is a given relative tolerance. We now summarize the steepest decent method as follows.

Algorithm 3.1 (Steepest Decent Method)**.** Given an initial approximation x_0 to u_1, and a relative tolerance `rtol`, the algorithm attempts to compute an approximate pair to (λ_1, u_1) with the prescribed `rtol`.

```
x_0 = x_0/||x_0||_B, ρ_0 = x_0^T A x_0, r_0 = A x_0 − ρ_0 B x_0;
for i = 0, 1, ..., do
    if ||r_i||/(||A x_i||_2 + |ρ_i| ||B x_i||_2) ≤ rtol, BREAK;
    Compute a, b, c as in (2.5) with x = x_i and p = r_i;
    Compute t_opt by (3.3);
    x̂ = x_i + t_opt r_i, x_{i+1} = x̂/||x̂||_B;
    ρ_{i+1} = x_{i+1}^T A x_{i+1}, r_{i+1} = A x_{i+1} − ρ_{i+1} B x_{i+1};
end
Return (ρ_i, x_i) as an approximate eigenpair to (λ_1, u_1).
```

A detailed convergence analysis for the case $B = I$ can be found in Faddeev and Faddeeva [3, p.430]. For the case $B \neq I$, it is in [18]. Also for $B = I$, The results of Knyazev and Skorokhodov [11] implies that locally

$$\frac{\rho_{i+1} - \lambda_1}{\rho_i - \lambda_1} \sim \left(\frac{1-\xi}{1+\xi}\right)^2, \quad \xi = \frac{\lambda_2 - \lambda_1}{\lambda_n - \lambda_1}.$$

3.2 Computing several eigenpairs

A natural way to compute the first few eigenpairs for the generalized eigenproblem $A - \lambda B$ is through deflating out the converged eigenpairs and applying Algorithm 3.1. Suppose we already have approximate eigenpairs

$$(\boldsymbol{\lambda}_i, \boldsymbol{u}_i),\ \|\boldsymbol{u}_i\|_B = 1,\ i = 1, 2, \ldots, j-1$$

to (λ_i, u_i), $i = 1, 2, \ldots, j-1$. It is reasonable to expect $\boldsymbol{u}_i^T B \boldsymbol{u}_\ell = 0$ approximately for $i \neq \ell$. Given an approximation $\boldsymbol{x}$ to u_j, we now explain how to compute the next approximation $\boldsymbol{y}$ to u_j. First, we B-orthogonalize $\boldsymbol{x}$ against $\boldsymbol{u}_i$, $i = 1, 2, \ldots, j-1$. This can be done by the Modified Gram-Schmidt process:

$$\text{for } i = 1, 2, \ldots, j-1, \text{ set } s = \boldsymbol{u}_i^T B \boldsymbol{x} \text{ and overwrite } \boldsymbol{x} = \boldsymbol{x} - s\,\boldsymbol{u}_i.$$

Evidently, we still refer to it by the same notation $\boldsymbol{x}$ as doing so will not cause any confusion. Next we B-orthogonalize $\boldsymbol{r} = A\boldsymbol{x} - \rho(\boldsymbol{x})B\boldsymbol{x}$ against $\boldsymbol{u}_i$, $i = 1, 2, \ldots, j-1$, again, by the Modified Gram-Schmidt process:

$$\hat{\boldsymbol{r}} = \boldsymbol{r}; \text{ and for } i = 1, 2, \ldots, j-1, \text{ set } s = \boldsymbol{u}_i^T B \hat{\boldsymbol{r}} \text{ and overwrite } \hat{\boldsymbol{r}} = \hat{\boldsymbol{r}} - s\,\boldsymbol{u}_i.$$

Note, unlike $\boldsymbol{x}$, the assignment of $\boldsymbol{r}$ stays the same before and after this process. Lastly, we solve

$$\inf_t \rho(\boldsymbol{x} + t\hat{\boldsymbol{r}}). \tag{3.5}$$

This is the line search problem (2.1) with $p = \hat{\boldsymbol{r}}$. Upon its solving, the next approximation is given by $\boldsymbol{y} = \boldsymbol{x} + t_{\text{opt}}\,\hat{\boldsymbol{r}}$. Again in actual implementation $\boldsymbol{y}$ is rescaled by, e.g., $\|\boldsymbol{y}\|_B$ and overwrites $\boldsymbol{x}$ and the process is repeated until convergence. We summarize the algorithm as follows.

Algorithm 3.2 (Deflated Steepest Decent Method)**.** Suppose accurate approximations to the first $j-1$ eigenpairs of (λ_i, u_i) are known as $(\boldsymbol{\lambda}_i, \boldsymbol{u}_i)$, and suppose that $\boldsymbol{u}_i$ for $i = 1, 2, \ldots, j-1$ are B-orthonormal. Given an initial approximation x_0 to u_j, and a relative tolerance `rtol`, the algorithm attempts to compute an approximation pair to (λ_j, u_j) with the prescribed `rtol`.

B-orthogonalize x_0 against $\boldsymbol{u}_i$, $i = 1, 2, \ldots, j-1$, and denote the orthogonalized vector still by x_0;
$x_0 = x_0/\|x_0\|_B$, $\rho_0 = x_0^T A x_0$, $r_0 = Ax_0 - \rho_0 B x_0$;
for $i = 0, 1, \ldots,$ do
    if $\|r_i\|/(\|Ax_i\|_2 + |\rho_i|\,\|Bx_i\|_2) \le$ `rtol`, **BREAK**;
    B-orthogonalize r_i against $\boldsymbol{u}_i$, $i = 1, 2, \ldots, j-1$, and denote the orthogonalized vector by $\hat{r}$;
    Compute a, b, c as in (2.5) with $x = x_i$ and $p = \hat{r}$;
    Compute t_{opt} by (2.9);
    $\hat{x} = x_i + t_{\text{opt}}\,\hat{r}$;
    B-orthogonalize $\hat{x}$ against $\boldsymbol{u}_i$, $i = 1, 2, \ldots, j-1$, and denote the orthogonalized vector still by $\hat{x}$;
    $x_{i+1} = \hat{x}/\|\hat{x}\|_B$, $\rho_{i+1} = x_{i+1}^T A x_{i+1}$, $r_{i+1} = Ax_{i+1} - \rho_{i+1} B x_{i+1}$;
end
Return (ρ_i, x_i) as an approximate eigenpair to (λ_j, u_j).

Another method for computing several eigenpairs as a variation to the steepest descent method is the so-called ***Simultaneous Rayleigh Quotient Minimization Method*** (SIRQIT) due to Longsine and McCormick [12]. The idea is to start with k linearly independent vectors whose span is intended to approximate the deflating subspace spanned by u_i, $i = 1, 2, \ldots, k$ and then compute another k linearly independent vectors whose span hopefully approximate the deflating subspace (much) better. This is in contrast to the method we just explained which compute one approximate eigenpair at time in the sequential order. This difference is reminiscent of the difference between the power method and the simultaneous subspace iteration.

Let $\boldsymbol{X} \in \mathbb{R}^{n\times k}$ whose columns are B orthonormal, i.e., $\boldsymbol{X}^T B \boldsymbol{X} = I_k$. The goal is to compute another $\boldsymbol{Y} \in \mathbb{R}^{n\times k}$ whose columns span a subspace that are hopefully closer to the the deflating subspace spanned by u_i, $i = 1, 2, \ldots, k$ than $\boldsymbol{X}$'s columns do. This is done as follows.

1. Project the eigenproblem for $A - \lambda B$ onto the subspace span($\boldsymbol{X}$) to get $\boldsymbol{X}^T A \boldsymbol{X} - \lambda I_k$, a much smaller problem. Compute its eigendecomposition:

$$\boldsymbol{Q}^T(\boldsymbol{X}^T A \boldsymbol{X})\boldsymbol{Q} = \Omega,\ \boldsymbol{Q}^T\boldsymbol{Q} = I_k,\ \Omega = \operatorname{diag}(\omega_1, \ldots, \omega_k)$$

   with $\omega_1 \le \ldots \le \omega_k$. Set $\boldsymbol{Z} = \boldsymbol{X}\boldsymbol{Q}$. Naturally we would pair $(\omega_i, \boldsymbol{Z}_{(:,i)})$ to (λ_i, u_i) as its approximation for each i. Set $\boldsymbol{R} = A\boldsymbol{Z} - B\boldsymbol{Z}\Omega$.

2. We seek a better approximation to u_i than $\boldsymbol{Z}_{(:,i)}$ does: solve

$$\inf_t \rho(\boldsymbol{Z}_{(:,1)} + t\,\boldsymbol{R}_{(:,1)})$$

and set $y_1 = \boldsymbol{Z}_{(:,1)} + t_{\text{opt}}\boldsymbol{R}_{(:,1)}$; and for $i = 2, \ldots, k$, B-orthogonalize $\boldsymbol{R}_{(:,i)}$ against $y_1, \ldots, y_{i-1}$ already computed and solve[1]

$$\inf_t \rho(\boldsymbol{Z}_{(:,i)} + t\,\hat{\boldsymbol{R}}_{(:,i)}),$$

where $\hat{\boldsymbol{R}}_{(:,i)}$ is the B-orthogonalized $\boldsymbol{R}_{(:,i)}$, and then set $\hat{y} = \boldsymbol{Z}_{(:,i)} + t_{\text{opt}}\hat{\boldsymbol{R}}_{(:,i)}$ and B-orthogonlize $\hat{y}$ against $y_1, \ldots, y_{i-1}$ to get y_i.

3. $\boldsymbol{Y} = [y_1, y_2, \ldots, y_k]$.

This is just one step of SIRQIT-G [12]. The complete process overwrites $\boldsymbol{X}$ by $\boldsymbol{Y}$ and repeats. It must also include testing for convergence.

An alternative to SIRQIT-G is SIRQIT-G2, also mentioned in [12]. It is theoretically more powerful, but arguably have the same (or similar) asymptotical speed of convergence. The idea, simply put, is to project the original eigenproblem onto $\text{span}(\boldsymbol{Z}, \boldsymbol{R})$ to lead to a $2k \times 2k$ eigenproblem. Solve the projected problem and select its k smallest eigenpairs. Specifically:

1. Same as Item 1 for SIRQIT-G;
2. B-orthogonalize $\boldsymbol{R}$ against $\boldsymbol{Z}$, calling the resulting matrix $\hat{\boldsymbol{R}}$. The columns of $\boldsymbol{Z}$ and $\hat{\boldsymbol{R}}$ together form a B-orthonormal basis of $\text{span}(\boldsymbol{Z}, \boldsymbol{R})$.
3. Project $A - \lambda B$ onto $\text{span}(\boldsymbol{Z}, \boldsymbol{R})$ to get

$$\begin{bmatrix} \boldsymbol{Z}^T \\ \hat{\boldsymbol{R}}^T \end{bmatrix} A \begin{bmatrix} \boldsymbol{Z} & \hat{\boldsymbol{R}}^T \end{bmatrix} - \lambda \begin{bmatrix} \boldsymbol{Z}^T \\ \hat{\boldsymbol{R}}^T \end{bmatrix} B \begin{bmatrix} \boldsymbol{Z} & \hat{\boldsymbol{R}}^T \end{bmatrix} = \begin{bmatrix} \Omega & \boldsymbol{R}^T\hat{\boldsymbol{R}} \\ \hat{\boldsymbol{R}}^T\boldsymbol{R} & \hat{\boldsymbol{R}}^T A\hat{\boldsymbol{R}} \end{bmatrix} - \lambda I. \tag{3.6}$$

   This is because $\hat{\boldsymbol{R}}^T A\boldsymbol{Z} = \hat{\boldsymbol{R}}^T(\boldsymbol{R} + B\boldsymbol{Z}\Omega) = \hat{\boldsymbol{R}}^T\boldsymbol{R}$.
4. Let $\boldsymbol{W}$ be the eigenvector matrix corresponding to the k smallest eigenvalues of the projected problem (3.6), and set $\boldsymbol{Y} = \begin{bmatrix} \boldsymbol{Z} & \hat{\boldsymbol{R}} \end{bmatrix} \boldsymbol{W}$.

Numerical experiments suggest that SIRQIT-G and SIRQIT-G2 exhibit similar asymptotic behavior. This can also be seen heuristically from (3.6) because at the convergence, $\boldsymbol{R}$ is nearly 0. It is conceivable that at the beginning of calculations, SIRQIT-G2 should perform better. But it is at a cost of solving $2k \times 2k$ eigenproblems, whereas SIRQIT-G always solves $k \times k$ eigenproblems.

3.3 Pre-conditioned steepest decent method

The steepest decent method, while always makes progress in driving the Rayleigh quotient towards a minimum, can be very slow in practice. The pre-conditioned steepest decent method is designed to overcome its slow convergence by modified its search direction $r(x)$. The method can be simply viewed as an application of the *vanilla* steepest decent method after a linear transformation to the Rayleigh quotient.

The kernel of the steepest decent method for $A - \lambda B$ is

$$t_{\text{opt}} = \operatorname*{argmin}_t \rho(x + tr(x)), \quad y = x + t_{\text{opt}}r(x), \tag{3.7}$$

[1] A possible variation is to B-orthogonaliz $\boldsymbol{Z}_{(:,i)}$ against $y_1, \ldots, y_{i-1}$, too, to get $\hat{\boldsymbol{Z}}_{(:,i)}$, and then to solve

$$\inf_t \rho(\hat{\boldsymbol{Z}}_{(:,i)} + t\,\hat{\boldsymbol{R}}_{(:,i)}).$$

where $r(x) = Ax - \rho(x)Bx$. Consider transformation $\widetilde{x} = Lx$, where L is $n \times n$ and nonsingular. For the Rayleigh quotient $\rho(x)$,

$$\rho(x) = \frac{x^T Ax}{x^T Bx} = \frac{\widetilde{x}^T L^{-T} AL^{-1} \widetilde{x}}{\widetilde{x}^T L^{-T} BL^{-1} \widetilde{x}} \tag{3.8}$$

which corresponding to the eigenproblem $L^{-T}AL^{-1} - \lambda L^{-T}BL^{-1}$. Adopt the notational convention that the same symbol with and without a tilde is for $A - \lambda B$ and for $L^{-T}AL^{-1} - \lambda L^{-T}BL^{-1}$, respectively. For example,

$$\widetilde{\rho}(\widetilde{x}) = \frac{\widetilde{x}^T L^{-T} AL^{-1} \widetilde{x}}{\widetilde{x}^T L^{-T} BL^{-1} \widetilde{x}} \equiv \rho(x), \ \widetilde{r}(\widetilde{x}) = L^{-T}AL^{-1}\widetilde{x} - \widetilde{\rho}(\widetilde{x})L^{-T}BL^{-1}\widetilde{x} \equiv L^{-T}r(x). \tag{3.9}$$

The kernel of the steepest decent method for $L^{-T}AL^{-1} - \lambda L^{-T}BL^{-1}$ is

$$\widetilde{t}_{\text{opt}} = \operatorname*{argmin}_{\widetilde{t}} \widetilde{\rho}(\widetilde{x} + \widetilde{t}\,\widetilde{r}(\widetilde{x})), \quad \widetilde{y} = \widetilde{x} + \widetilde{t}_{\text{opt}}\widetilde{r}(\widetilde{x}).$$

Eliminating the tilde variables to get back to the original variables, we have

$$y = x + \widetilde{t}_{\text{opt}}(L^T L)^{-1}\, r(x) \equiv x + \widetilde{t}_{\text{opt}} K\, r(x),$$

where $K = (L^T L)^{-1}$ is the so-called ***pre-conditioner***. Notice that

$$\widetilde{\rho}(\widetilde{x} + \widetilde{t}\widetilde{r}(\widetilde{x})) = \widetilde{\rho}(L(x + \widetilde{t}Kr(x))) = \rho(x + \widetilde{t}Kr(x)).$$

Therefore in terms of variable x and y, the steepest decent method for the transformed eigenproblem can be stated as, after dropping the tildes on the t-parameters,

$$t_{\text{opt}} = \operatorname*{argmin}_{t} \rho(x + tKr(x)), \quad y = x + t_{\text{opt}}Kr(x). \tag{3.10}$$

Comparing (3.7) and (3.10), we see the difference is the modification of the search direction from $r(x)$ to $Kr(x)$ by the selected pre-conditioner K. In view of this, there are pre-conditioned versions of all algorithms previously discussed in this section:

1. For Algorithm 3.1, replace all r_i in Lines 4 and 6 by Kr_i;
2. For Algorithm 3.2, replace all r_i in Line 5 by Kr_i;
3. For SIRQIT-G and SIRQIT-G2, overwrite $\boldsymbol{R}$ by $K\boldsymbol{R}$ immediately after the end of Item 1.

The pre-conditioned SIRQIT-G and SIRQIT-G2 as stated here are rather straightforward once the idea of pre-conditioning is explained, but they were not included in Longsine and McCormick [12].

There are some estimates on convergence rates for the pre-conditioned steepest decent method. These estimates do indeed show that the rates are dramatically improved with suitable pre-conditioners. We present one easily proven estimate here which is essentially due to Samokish [16] who studied the case when $B = I$. The reader is referred to [10, 14] for further reading.

Theorem 3.1 (Samokish)**.** *Let* $t_{\text{opt}} = \operatorname{argmin}_t \rho(x + tKr(x))$ *and* $y = x + tKr(x)$*, and denote*[2] *the smallest positive and largest eigenvalue of* $K(A - \lambda_1 B)$ *by* γ *and* Γ*. If*

$$\tau\left(\sqrt{\Gamma} + \epsilon\right)\epsilon < 1,$$

[2]It is worth emphasizing that $A - \lambda_1 B$ is singular and hence its smallest eigenvalue is 0, and γ is its smallest positive nonzero eigenvalue.

where

$$\epsilon = \sqrt{\|B^{1/2}KB^{1/2}\|_2\,[\rho(x)-\lambda_1]}, \quad \tau = \frac{2}{\gamma+\Gamma},$$

then

$$\rho(y)-\lambda_1 \le \left[\frac{\Delta+\tau\sqrt{\Gamma}\,\epsilon}{1-\tau(\sqrt{\Gamma}+\epsilon)\epsilon}\right]^2 [\rho(x)-\lambda_1], \tag{3.11}$$

where $\kappa = \Gamma/\gamma$, $\Delta = (\kappa-1)/(\kappa+1)$.

Proof. Let $z = x - \tau Kr(x)$. Then $\lambda_1 \le \rho(y) \le \rho(z)$ and thus $\rho(y)-\lambda_1 \le \rho(z)-\lambda_1$. So it suffices to show that $\rho(z)-\lambda_1$ is no bigger than the right-hand side of (3.11).

Note that $A-\lambda_1 B$ is symmetric positive semi-definite. For any vector w (see Exercise 3.3), we have

$$\|w\|^2_{A-\lambda_1 B} = (\rho(w)-\lambda_1)\|w\|^2_B, \tag{3.12}$$

$$\|[I-\tau K(A-\lambda_1 B)]w\|_{A-\lambda_1 B} \le \Delta\|w\|_{A-\lambda_1 B}. \tag{3.13}$$

Write $z = [I-\tau K(A-\lambda_1 B)]x + \tau[\rho(x)-\lambda_1]KBx$. Without loss of generality, we may assume $\|x\|_B = 1$. We have

$$\begin{aligned}
\|z\|_{A-\lambda_1 B} &= \sqrt{\rho(z)-\lambda_1}\|z\|_B,\\
\|z\|_{A-\lambda_1 B} &\le \|[I-\tau K(A-\lambda_1 B)]x\|_{A-\lambda_1 B} + \tau[\rho(x)-\lambda_1]\|KBx\|_{A-\lambda_1 B}\\
&\le \Delta\|x\|_{A-\lambda_1 B} + \tau[\rho(x)-\lambda_1]\sqrt{\Gamma}\|Bx\|_K\\
&\le \Delta\sqrt{\rho(x)-\lambda_1} + \tau[\rho(x)-\lambda_1]\sqrt{\Gamma\,\|B^{1/2}KB^{1/2}\|_2}\\
&= (\Delta+\tau\sqrt{\Gamma}\,\epsilon)\sqrt{\rho(x)-\lambda_1},\\
\|z\|_B &\ge \|x\|_B - \tau\|Kr(x)\|_B\\
&= 1-\tau\|Kr(x)\|_B,\\
\|Kr(x)\|_B &= \|K(A-\lambda_1 B)x - [\rho(x)-\lambda_1]KBx\|_B\\
&\le \|K(A-\lambda_1 B)x\|_B + [\rho(x)-\lambda_1]\|KBx\|_B\\
&\le \sqrt{\|K^{1/2}BK^{1/2}\|_2\Gamma}\|x\|_{A-\lambda_1 B} + [\rho(x)-\lambda_1]\|B^{1/2}KB^{1/2}\|_2\|x\|_B\\
&= \sqrt{\Gamma}\epsilon + \epsilon^2.
\end{aligned}$$

Finally use

$$\rho(z)-\lambda_1 = \frac{\|z\|^2_{A-\lambda_1 B}}{\|z\|^2_B} \le \frac{\|z\|^2_{A-\lambda_1 B}}{[1-\tau\|Kr(x)\|_B]^2}$$

to complete the proof. □

3.4 Discussions on selecting good pre-conditioners

One quick conclusion that can be drawn from Theorem 3.1 is that asymptotically $\rho(y)-\lambda_1$ is reduced by a factor of at least Δ^2 which depends on the conditioning of $K(A-\lambda_1 B)$, after its zero eigenvalues discarded, but not the eigenvalues of $A-\lambda_1 B$.

One important aspect of Theorem 3.1 lies as to what constitutes a good pre-conditioner, namely those making Γ/γ as close to 1 as possible. Since $\Gamma/\gamma = 1$ for $K = (A-\lambda_1 B)^\dagger$, the Moore-Penrose inverse, $K \approx (A-\lambda_1 B)^\dagger$ would be a good pre-conditioner, and $K = (A-\lambda_1 B)^\dagger$ is the best one could hope for although albeit impractical. Naturally this suggests that, for example, to let $K = (L^T L)^{-1}$ where $A-\lambda_1 B \approx L^T L$, an incomplete Cholesky decomposition.

In practice, however, λ_1 is not available to begin with. A remedy would be to estimate a lower bound μ of λ_1 and compute $L^T L \approx A - \mu B$, instead.

In some practical situations, A is also positive definite. In such cases, often simply $\mu = 0$ is chosen [4].

Exercises

3.1. Research Question. The quantitative estimates by Knyazev and Skorokhodov [11] are for $B = I$. Try to obtain some estimates when B is a general symmetric and positive definite matrix.

3.2. Conceivably some of the first k eigenpairs got converged faster by $(\omega_i, \boldsymbol{Z}_{(:,i)})$ than others in SIRQIT-G, SIRQIT-G2 and their pre-conditioned versions. For better efficiency, any converged eigenpairs should not be carried inside $\boldsymbol{X}$ until all k eigenpairs are accurately computed. Design a deflation scheme for the purpose.

3.3. Verify (3.12) and (3.13).

3.4. Prove that the eigenvalues of $K(A-\lambda_1 B)$ are real and nonnegative, where K is symmetric positive definite.

4 Conjugate gradient method

The Conjugate Gradient (CG) method was originally proposed in 1950s by Hestenes and Stiefel [7] for solving linear system $Hx = b$ with symmetric and positive definite H, as an alternative to the Gaussian elimination method, and later was interpreted as an efficient iterative method for large scale linear systems. In the 1960s, it was extended by Fletcher and Reeves [5] as an iterative method for nonlinear optimization problems. The extension is almost verbatim. Because of the optimality properties (1.2) and (1.3) of Rayleigh quotients, it is natural to apply the CG method to compute a few eigenpairs of $A - \lambda B$. For a better understanding of the application, the reader is strongly recommended to carefully study the CG method for linear systems. In [6], an illuminating step-by-step derivation of the method is presented. A different derivation is given in [13]. The connection between the CG method and the Lanczos procedure is presented in [2].

4.1 CG for linear systems

Let H be $n \times n$, symmetric, and positive definite. Define

$$\phi(x) = \frac{1}{2} x^T H x - x^T b. \tag{4.1}$$

It is a quadratic functional in x. It is convex and has a unique local and global minimum at $x = H^{-1} b$. In fact,

$$\begin{aligned}\phi(x) &= \frac{1}{2}(Hx - b)^T H^{-1}(Hx - b) - \frac{1}{2} b^T H^{-1} b \\ &= \frac{1}{2}(x - H^{-1}b)^T H (x - H^{-1}b) - \frac{1}{2} b^T H^{-1} b.\end{aligned}$$

It can be computed that the gradient $\nabla\phi(x) = Hx - b$, and its Hessian matrix is H itself. Define the residual vector

$$r(x) = Hx - b.$$

Given an initial guess x_0 to $H^{-1}b$, the CG method iteratively produces a sequence of approximations x_i and conjugate searching directions p_i, i.e., $p_i^T H p_j = 0$ for $i \neq j$, with $p_0 = r(x_0)$ such that

$$\phi(x_{i+1}) = \min_t \phi(x_i + \alpha p_i).$$

The complete algorithm can be described as follows.

1. Give an initial guess x_0, compute $r_0 = Ax_0 - b$, and set $p_0 = r_0$;
2. For $i = 0, 1, \ldots,$ do

$$\begin{aligned} \alpha_i &= \operatorname*{argmin}_\alpha \phi(x_i + \alpha p_i), & x_{i+1} &= x_i + \alpha_i p_i, \\ r_{i+1} &= r_i + \alpha_i H p_i, & p_{i+1} &= r_{i+1} + \beta_i p_i, \end{aligned}$$

   where β_i is chosen so that $p_{i+1}^T H p_i = 0$.

There are various mathematically equivalent expressions for α_i and β_i:

$$\alpha_i = -\frac{p_i^T r_i}{p_i^T H p_i} \tag{4.2a}$$

$$= -\frac{r_i^T r_i}{p_i^T H p_i} \tag{4.2b}$$

and

$$\beta_i = -\frac{p_i^T H r_{i+1}}{p_i^T H p_i} \tag{4.3a}$$

$$= \frac{r_{i+1}^T r_{i+1}}{r_i^T r_i} \tag{4.3b}$$

$$= \frac{r_{i+1}^T (r_{i+1} - r_i)}{r_i^T r_i}. \tag{4.3c}$$

Commonly used ones are (4.2b), (4.3b), and (4.3c). Their numerical behaviors can be quite different sometimes.

In the absence of roundoff errors, it can be proved that if $r_\ell \neq 0$, then $p_0, p_1, \ldots, p_\ell$ are linearly independent, and

$$r_i^T r_j = 0, \quad \text{for } 0 \leq i,\, j \leq \ell \text{ and } i \neq j, \tag{4.4a}$$

$$\operatorname{span}\{r_0, r_1, \ldots, r_\ell\} = \operatorname{span}\{p_0, p_1, \ldots, p_\ell\} \tag{4.4b}$$

$$= \operatorname{span}\{r_0, Hr_0, \ldots, H^\ell r_0\}, \tag{4.4c}$$

$$p_i^T H p_j = 0, \quad \text{for } 0 \leq i,\, j \leq \ell \text{ and } i \neq j, \tag{4.4d}$$

$$\phi(x_\ell) = \min_{t_0, \ldots, t_\ell} \phi(x_0 + t_0 p_0 + t_1 p_1 + \cdots + t_\ell p_\ell). \tag{4.4e}$$

That the CG method converges in at most n steps is a consequence of these properties.

4.2 Computing one eigenpair

In extending the CG method, the key is to recognize that the residual $r(x)$ in the linear system case plays the role of the gradient direction for $\phi(x)$, the objective function, in (4.1). For the eigenproblem of $A - \lambda B$, the objective function is the Rayleigh quotient

$$\rho(x) = \frac{x^T A x}{x^T B x} \tag{1.1}$$

whose gradient is collinear to

$$r(x) = Ax - \rho(x)\, Bx. \tag{1.6}$$

This observation naturally leads to the following CG method for computing (λ_1, u_1).

Algorithm 4.1 (Conjugate Gradient Method)**.** Given an initial approximation x_0 to u_1, and a relative tolerance `rtol`, the algorithm attempts to compute an approximate pair to (λ_1, u_1) with the prescribed `rtol`.

$x_0 = x_0/\|x_0\|_B$, $\rho_0 = x_0^T A x_0$, $r_0 = Ax_0 - \rho_0 B x_0$, $p_0 = r_0$;
for $i = 0, 1, \ldots,$ do
  if $\|r_i\|/(\|Ax_i\|_2 + |\rho_i|\, \|Bx_i\|_2) \le$ `rtol`, **BREAK**;
  Compute a, b, c as in (2.5) with $x = x_i$ and $p = p_i$;
  Compute $\alpha_i = t_{\text{opt}}$ by (2.9);
  $\hat{x} = x_i + \alpha_i\, p_i$, $x_{i+1} = \hat{x}/\|\hat{x}\|_B$;
  $\rho_{i+1} = x_{i+1}^T A x_{i+1}$, $r_{i+1} = Ax_{i+1} - \rho_{i+1} B x_{i+1}$, $p_{i+1} = r_{i+1} + \beta_i p_i$,
  where β_i is commonly chosen by either (4.3b) or (4.3c);
end
Return (ρ_i, x_i) as an approximate eigenpair to (λ_1, u_1).

No longer the properties listed in (4.4) hold because $\rho(x)$ is not quadratic. But still $p_i^T r_{i+1} = 0$. See Exercise 4.2. No longer the choice of β_i by either (4.3b) or (4.3c) ensures any conjugate relation even among adjacent p_i and p_{i+1}. Other choices have been made to rectify that, for example

$$\beta_i = -\frac{\langle p_i, r_{i+1}\rangle_{H(x_{i+1})}}{\langle p_i, p_i\rangle_{H(x_{i+1})}}, \tag{4.3c}$$

$$\beta_i = -\frac{\langle p_i, r_{i+1}\rangle_M}{\langle p_i, p_i\rangle_M}, \tag{4.3d}$$

where $H(x_{i+1})$ is the Hessian of $\rho(x)$ at $x = x_{i+1}$, and M is some positive definite matrix. We have $\langle p_i, p_{i+1}\rangle_{H(x_{i+1})} = 0$ in the case of (4.3c) and $\langle p_i, p_{i+1}\rangle_M = 0$ in the case of (4.3d).

Takahashi [17] suggested another choice for β_i: it, together with α_{i+1}, minimizes the objective function at $x_{i+2} = x_{i+1} + \alpha_{i+1}(r_{i+1} + \beta_i p_i)$. This is equivalent to minimize, for the present case, the Rayleigh quotient in $\text{span}\{x_{i+1}, r_{i+1}, p_i\} = \text{span}\{x_{i+1}, r_{i+1}, x_i\}$. Hence it is known as being ***locally optimal*** [15, 8].

Since $p_0 = r_0$, x_1 is simply the steepest decent solution; so is x_{i+2} if $\beta_i = 0$. In the nonlinear optimization, it is suggested to reset β_i to 0 every n CG steps, i.e., $i = 0 \pmod n$, so-called ***restarting*** to "periodically refresh the algorithm, erasing old information that may not be beneficial" [13]. In the context of large scale eigenvalue computations as we have here, it is hoped that convergence would occur much sooner than i reaches n, for otherwise it is not practical.

An convergence analysis for the CG methods is given by Yang [18]. Feng and Owen [4] presented a very enlightening asymptotic convergence analysis. However no quantitative estimate on the convergence rate of the CG method is available yet.

Exercises

4.1. Establish the properties (4.4). Use the fact that $\dfrac{d}{d\alpha}\phi(x_i + \alpha p_i) = 0$ at $\alpha = \alpha_i$ to obtain (4.2a) and $p_i^T r_{i+1} = 0$, and the fact that $p_i^T H p_{i+1} = 0$ and $p_{i+1} = r_{i+1} + \beta_i p_i$ to obtain (4.3a).

4.2. Use the fact that $\dfrac{d}{d\alpha}\rho(x_i + \alpha p_i) = 0$ at $\alpha = \alpha_i$ to show that $p_i^T r_{i+1} = 0$.

4.3 Computing several eigenpairs

Now that we know how to use the CG method to compute one eigenpair of $A - \lambda B$. The idea to modify it to compute several eigenpairs is much the same as we did to the steepest decent method, i.e., through deflation to compute the left most eigenpairs one at a time, or establishing some kind of simultaneous approximations to the k left most eigenpairs at the same time.

Through deflation, we have

Algorithm 4.2 (Deflated Conjugate Gradient Method)**.** Suppose accurate approximations to the first $j-1$ eigenpairs of (λ_i, u_i) are known as $(\boldsymbol{\lambda}_i, \boldsymbol{u}_i)$ for $i = 1, 2, \ldots, j-1$ with $\|\boldsymbol{u}_i\|_B = 1$, and suppose that $\boldsymbol{u}_i$ for $i = 1, 2, \ldots, j-1$ are B-orthonormal. Given an initial approximation x_0 to u_j, and a relative tolerance `rtol`, the algorithm attempts to compute an approximation pair to (λ_j, u_j) with the prescribed `rtol`.

1 B-orthogonalize x_0 against $\boldsymbol{u}_i$, $i = 1, 2, \ldots, j-1$, and denote the orthogonalized vector still by x_0;
2 $x_0 = x_0/\|x_0\|_B$, $\rho_0 = x_0^T A x_0$, $r_0 = Ax_0 - \rho_0 B x_0$, $p_0 = r_0$;
3 for $i = 0, 1, \ldots,$ do
4 if $\|r_i\|/(\|Ax_i\|_2 + |\rho_i|\,\|Bx_i\|_2) \le$ `rtol`, **BREAK**;
5 B-orthogonalize p_i against $\boldsymbol{u}_i$, $i = 1, 2, \ldots, j-1$, and denote the orthogonalized vector still by p_i;
6 Compute a, b, c as in (2.5) with $x = x_i$ and $p = p_i$;
7 Compute $\alpha_i = t_{\text{opt}}$ by (2.9);
8 $\hat{x} = x_i + \alpha_i\, p_i$;
9 B-orthogonalize $\hat{x}$ against $\boldsymbol{u}_i$, $i = 1, 2, \ldots, j-1$, and denote the orthogonalized vector still by $\hat{x}$;
10 $x_{i+1} = \hat{x}/\|\hat{x}\|_B$, $\rho_{i+1} = x_{i+1}^T A x_{i+1}$, $r_{i+1} = Ax_{i+1} - \rho_{i+1} B x_{i+1}$;
11 $p_{i+1} = r_{i+1} + \beta_i p_i$, where β_i is commonly chosen by either (4.3b) or (4.3c);
12 end
13 Return (ρ_i, x_i) as an approximate eigenpair to (λ_j, u_j).

In Line 11, β_i could also be chosen to achieve local optimality, i.e., through minimizing the Rayleigh quotient in $\text{span}\{x_{i+1}, x_i, \hat{r}_{i+1}\}$, where $\hat{r}_{i+1}$ is the B-orthogonalized r_{i+1} against $\boldsymbol{u}_i$, $i = 1, 2, \ldots, j-1$.

Similar to SIRQIT-G, Longsine and McCormick [12] proposed SIRQIT-CG. The difference is in the selection of searching directions, much like that between the steepest decent method and the CG method. SIRQIT-CG is outlined as follows:

1. Given $X_0 \in \mathbb{R}^{n\times k}$ whose columns are B-orthonormal, i.e., $X_0^T B X_0 = I_k$, and span an approximate deflating subspace $\text{span}\{u_j,\ 1 \le j \le k\}$.

2. Compute eigendecomposition:

$$Q^T (X_0^T A X_0) Q = \Omega,\ Q^T Q = I_k,\ \Omega = \text{diag}(\omega_1, \ldots, \omega_k)$$

   with $\omega_1 \le \cdots \le \omega_k$.

3. Set $Z_0 = X_0 Q$, $R_0 = AZ_0 - BZ_0\Omega$, and $P_0 = R_0$.

4. For i=0,1,. . . ,

   (a) $\hat{X} = Z_i + P_i D_i$, where D_i is diagonal and

$$(D_i)_{(j,j)} = \operatorname*{argmin}_t \rho((Z_i)_{(:,j)} + t\,(P_i)_{(:,j)}),\ 1 \le j \le k. \tag{4.5}$$

(b) B-orthogonalize the columns of $\hat{X}$ to get X_{i+1}.

(c) Compute eigendecomposition:

$$Q^T(X_{i+1}^T A X_{i+1})Q = \Omega,\ Q^T Q = I_k,\ \Omega = \mathrm{diag}(\omega_1, \ldots, \omega_k).$$

with $\omega_1 \le \cdots \le \omega_k$.

(d) Set $Z_{i+1} = X_{i+1}Q$, $R_{i+1} = AZ_{i+1} - BZ_{i+1}\Omega$, and $P_{i+1} = R_{i+1} + P_i E_i$, where E_i is diagonal and

$$(E_i)_{(j,j)} = \frac{(R_{i+1})_{(:,j)}^T (R_{i+1})_{(:,j)}}{(R_i)_{(:,j)}^T (R_i)_{(:,j)}}, \quad 1 \le j \le k. \tag{4.6}$$

The roles of the diagonal entries of D_i and E_i are the same as those of α_i and β_i, respectively, in the CG method. Obviously (4.6) only lists one possible choice. Other choices for β_i can be suitable ones, too, for $(E_i)_{(j,j)}$.

This version of SIRQIT-CG is given in [12]. There are other conceivable variations. For example, we may replace Items 2 and 4(a,b,c) by projecting $A - \lambda B$ onto $\mathrm{span}\{X_i, P_i\}$, taking Q to be the eigenvector matrix corresponding to the k smallest eigenvalues of the projected problem, much like in SIRQIT-G2. We call this variation SIRQIT-CG2. We may also determine E_i by projecting $A - \lambda B$ onto $\mathrm{span}\{X_{i+1}, X_i, P_i\}$, much as the locally optimal way to determine β_i for the CG method, to obtain another variation. We call this SIRQIT-LOCG. Yet another possible modification is for Item 4(a,b): compute the columns of X_{i+1} one at a time sequentially with the present search direction being the B-orthogonalized $(P_i)_{(:,j)}$ against the already computed columns of X_{i+1}.

4.4 Pre-conditioned conjugate gradient method

The preconditioned version of the CG method can be similarly viewed as the application of the *vanilla* CG method after a linear transformation $\widetilde{x} = Lx$ done on the Rayleigh quotient $\rho(x)$, as we did in Subsection 3.3. Again set $\widetilde{x} = Lx$, and then (3.8) and (3.9) hold. Ignoring the (theoretically nonessential) scaling operation on $\hat{x}_{i+1}$, we conclude the main part of the CG method applied to minimize $\widetilde{\rho}(\widetilde{x})$ is

$$\begin{aligned} \widetilde{\alpha}_i &= \operatorname*{argmin}_{\widetilde{\alpha}} \widetilde{\rho}(\widetilde{x}_i + \widetilde{\alpha}\widetilde{p}_i), & \widetilde{x}_{i+1} &= \widetilde{x}_i + \widetilde{\alpha}_i \widetilde{p}_i, \\ \widetilde{r}_{i+1} &= L^{-T}AL^{-1}\widetilde{x}_{i+1} - \widetilde{\rho}(\widetilde{x}_{i+1})L^{-T}BV^{-1}\widetilde{x}_{i+1}, & \widetilde{p}_{i+1} &= \widetilde{r}_{i+1} + \widetilde{\beta}_i \widetilde{p}_i. \end{aligned}$$

Perform substitutions $\widetilde{x}_j = Lx_j$ and $\widetilde{r}_j = L^{-T}r_j$, and incorporate (3.8) and (3.9) to get

$$\begin{aligned} \widetilde{\alpha}_i &= \operatorname*{argmin}_{\widetilde{\alpha}} \rho(x_i + \widetilde{\alpha}L^{-1}\widetilde{p}_i), & x_{i+1} &= x_i + \widetilde{\alpha}_i L^{-1}\widetilde{p}_i, \\ r_{i+1} &= Ax_{i+1} - \rho(x_{i+1})Bx_{i+1}, & L^{-1}\widetilde{p}_{i+1} &= (L^TL)^{-1}r_{i+1} + \widetilde{\beta}_i L^{-1}\widetilde{p}_i. \end{aligned}$$

Finally rename $L^{-1}\widetilde{p}_j$ by p_j, and $(L^TL)^{-1}$ by K, the pre-conditioner, to arrive at the following pre-conditioned Conjugate Gradient Method.

1. Give an initial guess x_0 and a preconditioner K, compute $r_0 = Ax_0 - \rho(x_0)Bx_0$, and set $p_0 = Kr_0$;

2. For $i = 0, 1, \ldots,$ do

$$\begin{aligned} \alpha_i &= \operatorname*{argmin}_{\alpha} \rho(x_i + \alpha p_i), & \hat{x}_{i+1} &= x_i + \alpha_i p_i, & x_{i+1} &= \hat{x}_{i+1}/\|\hat{x}_{i+1}\|_B, \\ r_{i+1} &= Ax_{i+1} - \rho(x_{i+1})Bx_{i+1}, & p_{i+1} &= Kr_{i+1} + \beta_i p_i, \end{aligned}$$

where β_i is commonly chosen by one of

$$\beta_i = \frac{r_{i+1}^T K r_{i+1}}{r_i^T K r_i}, \tag{4.7a}$$

$$\beta_i = \frac{r_{i+1}^T K (r_{i+1} - r_i)}{r_i^T K r_i}. \tag{4.7b}$$

Other choices, as counterparts to (4.3c) and (4.3d), for β_i have also been used:

$$\beta_i = -\frac{\langle p_i, K r_{i+1} \rangle_{H(x_{i+1})}}{\langle p_i, p_i \rangle_{H(x_{i+1})}}, \tag{4.7c}$$

$$\beta_i = -\frac{\langle p_i, K r_{i+1} \rangle_M}{\langle p_i, p_i \rangle_M}, \tag{4.7d}$$

Comparing the CG method and its pre-conditioned version, we see the difference is the modification of the residual from r_i to Kr_i by the selected pre-conditioner K. In view of this, there are pre-conditioned versions of all algorithms previously discussed in this section — simply by multiplying all residual vectors by K. In particular SIRQIT-LOCG combined with a pre-conditioner become the so-called Locally Optimal Block Preconditioned Conjugate Gradient Method (LOBPCG) [9].

Our discussions on selecting a good pre-conditioner in Subsection 3.4 are often followed for the pre-conditioned CG method and its many variations. Numerical tests support this practice.

There are various heuristics on the convergence rates of the pre-conditioned CG method [1], but none is rigorously proved. Even less can be said about the theoretical analysis of block (or subspace) versions of the pre-conditioned CG method.

References

[1] L. Bergamaschi, G. Gambolati, and G. Pini, *Asymptotic convergence of conjugate gradient methods for the partial symmetric eigenproblem*, Numer. Linear Algebra Appl., 4 (1997), pp. 69–84.

[2] J. W. Demmel, *Applied Numerical Linear Algebra*, SIAM, Philadelphia, PA, 1997.

[3] D. K. Faddeev and V. N. Faddeeva, *Computational Methods of Linear Algebra*, Undergraduate Mathematics Books, W.H.Freeman & Co Ltd, San Francisco, 1963. Translated by R. C. Williams.

[4] Y. T. Feng and D. R. J. Owen, *Conjugate gradient methods for solving the smallest eigenpair of large symmetric eigenvalue problems*, Internat. J. Numer. Methods Eng., 39 (1996), pp. 2209–2229.

[5] R. Fletcher and C. M.Reeves, *Function minimization by conjugate gradients*, Comput. J., 7 (1964), pp. 149–154.

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

[7] M. R. Hestenes and E. Stiefel, *Methods of conjugate gradients for solving linear systems*, J. Res. Nat. Bur. Standards, 49 (1952), pp. 409–436.

[8] A. V. Knyazev, *A preconditioned conjugate gradient method for eigenvalue problems and its implementation in a subspace*, Internat Series Numer. Math., 96 (1991), pp. 143–154.

[9] A. V. Knyazev, *Toward the optimal preconditioned eigensolver: Locally optimal block preconditioned conjugate gradient method*, SIAM J. Sci. Comput., 23 (2001), pp. 517–541.

[10] A. V. Knyazev and K. Neymeyr, *A geometric theory for preconditioned inverse iteration III: A short and sharp convergence estimate for generalized eigenvalue problems*, Linear Algebra Appl., 358 (2003), pp. 95–114.

[11] A. V. Knyazev and A. L. Skorokhodov, *On exact estimates of the convergence rate of the steepest ascent method in the symmetric eigenvalue problem*, Linear Algebra Appl., 154-156 (1991), pp. 245–257.

[12] D. E. Longsine and S. F. McCormick, *Simultaneous Rayleigh-quotient minimization methods for $Ax = \lambda Bx$*, Linear Algebra Appl., 34 (1980), pp. 195–234.

[13] J. Nocedal and S. Wright, *Numerical Optimization*, Springer, 2nd ed., 2006.

[14] E. E. Ovtchinnikov, *Sharp convergence estimates for the preconditioned steepest descent method for Hermitian eigenvalue problems*, SIAM J. Numer. Anal., 43 (2006), pp. 2668–2689.

[15] B. T. Polyak, *Introduction to optimization*, Optimization Software, New York, 1987.

[16] B. Samokish, *The steepest descent method for an eigenvalue problem with semi-bounded operators*, Izv. Vyssh. Uchebn. Zaved. Mat., 5 (1958), pp. 105–114. in Russian.

[17] I. Takahashi, *A note on the conjugate gradient method*, Inform. Process. Japan, 5 (1965), pp. 45–49.

[18] H. Yang, *Conjugate gradient methods for the rayleigh quotient minimization of generalized eigenvalue problems*, Computing, 51 (1993), pp. 79–94.