Numerical Analysis – Lecture 6[1]

Method 2.7 (Inverse iteration). This method is highly useful in practice. It is similar to the power method 2.2, except that, instead of $\boldsymbol{x}^{(k+1)}$ being a multiple of $A\boldsymbol{x}^{(k)}$, we make the choice

$$(A-\hat{\lambda}I)\boldsymbol{x}^{(k+1)} = \text{ scalar multiple of } \boldsymbol{x}^{(k)}, \qquad k=0,1,\ldots, \tag{2.1}$$

where $\hat{\lambda}$ is a scalar that may depend on k. Therefore the calculation of $\boldsymbol{x}^{(k+1)}$ from $\boldsymbol{x}^{(k)}$ requires the solution of an $n \times n$ system of linear equations whose matrix is $A-\hat{\lambda}I$. Further, if $\hat{\lambda}$ is a constant and if $A-\hat{\lambda}I$ is nonsingular, we deduce from (2.1) that $\boldsymbol{x}^{(k+1)}$ is a multiple of $(A-\hat{\lambda}I)^{-k-1}\boldsymbol{x}^{(0)}$.

We again let $\boldsymbol{x}^{(0)} = \sum_{j=1}^{n} \theta_j \boldsymbol{w}_j$, as in the proof of Theorem 2.3, assuming that $\boldsymbol{w}_j$, $j=1,2,\ldots,n$, are linearly independent eigenvectors of A that satisfy $A\boldsymbol{w}_j = \lambda_j \boldsymbol{w}_j$. Therefore we note that the eigenvalue equation implies $(A-\hat{\lambda}I)\boldsymbol{w}_j = (\lambda_j - \hat{\lambda})\boldsymbol{w}_j$, which in turn implies $(A-\hat{\lambda}I)^{-1}\boldsymbol{w}_j = (\lambda_j-\hat{\lambda})^{-1}\boldsymbol{w}_j$. It follows that $\boldsymbol{x}^{(k+1)}$ is a multiple of

$$(A-\hat{\lambda}I)^{-k-1}\boldsymbol{x}^{(0)} = \sum_{j=1}^{n} \theta_j (A-\hat{\lambda}I)^{-k-1}\boldsymbol{w}_j = \sum_{j=1}^{n} \theta_j (\lambda_j-\hat{\lambda})^{-k-1}\boldsymbol{w}_j.$$

Thus, if the lth number in the set $\{|\lambda_j-\hat{\lambda}| \,:\, j=1,2,\ldots,n\}$ is smaller than the rest and if θ_l is nonzero, then $\boldsymbol{x}^{(k+1)}$ tends to be a multiple of $\boldsymbol{w}_l$ as $k\to\infty$. We see that the speed of convergence can be excellent if $\hat{\lambda}$ is very close to λ_l. Further, it can be made even faster by adjusting $\hat{\lambda}$ during the calculation. Typical details are given in the following implementation.

Algorithm 2.8 (Typical implementation of inverse iteration)

0. Set $\hat{\lambda}$ to an estimate of an eigenvalue of A. Either prescribe $\boldsymbol{x}^{(0)} \neq \boldsymbol{0}$ or let it be chosen automatically in **3**. Let $0 < \varepsilon \ll 1$ and set $k=0$.

1. Calculate (with pivoting if necessary) the LU factorization of $A-\hat{\lambda}I$.

2. Stop if U is singular because then $\hat{\lambda}$ is an eigenvalue of A, while its eigenvector is any vector in the null space of U: it can be found easily, U being upper triangular.

3. If $k=0$ and unless $\boldsymbol{x}^{(0)}$ has been prescribed, define $\boldsymbol{x}^{(1)}$ by $U\boldsymbol{x}^{(1)} = \boldsymbol{e}_i$, where $\boldsymbol{e}_i$ is the ith coordinate vector, and where i is defined by the property that $|U_{i,i}|$ is the smallest modulus of a diagonal element of U. Further, we set $\boldsymbol{x}^{(0)} = L\boldsymbol{e}_i$, in order to satisfy $(A-\hat{\lambda}I)\boldsymbol{x}^{(1)} = \boldsymbol{x}^{(0)}$.
For $k \geq 1$ $\boldsymbol{x}^{(k+1)}$ is calculated by solving $(A-\hat{\lambda}I)\boldsymbol{x}^{(k+1)} = LU\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)}$, which is straightforward using the LU factorization from **1.**

4. Set η to the number that minimizes $f(\eta) = \|\boldsymbol{x}^{(k)} - \eta\boldsymbol{x}^{(k+1)}\|$.

5. Stop if $f(\eta) \leq \varepsilon\|\boldsymbol{x}^{(k+1)}\|$. Since $f(\eta) = \|(A-\hat{\lambda}I)^{-1}[A-(\eta+\hat{\lambda})I]\boldsymbol{x}^{(k)}\|$, we let $\hat{\lambda}+\eta$ be the calculated eigenvalue of A and $\boldsymbol{x}^{(k+1)}/\|\boldsymbol{x}^{(k+1)}\|$ be its eigenvector.

6. Otherwise, replace $\boldsymbol{x}^{(k+1)}$ by $\boldsymbol{x}^{(k+1)}/\|\boldsymbol{x}^{(k+1)}\|$, increase k by one, and either return to **3** without changing $\hat{\lambda}$ or to **1** after replacing $\hat{\lambda}$ by $\hat{\lambda}+\eta$.

Remark 2.9 (Further on inverse iteration). Algorithm 2.8 is very efficient if A is an *upper Hessenberg matrix:* every element of A under the first subdiagonal is zero (i.e. $A_{i,j}=0$, $j \leq i-2$). In this case the LU factorization in **1** requires just $\mathcal{O}(n^2)$ or $\mathcal{O}(n)$ when A is nonsymmetric or symmetric, respectively. Thus the replacement of $\hat{\lambda}$ by $\hat{\lambda}+\eta$ in **6** need not be expensive, so fast convergence can often be achieved easily.

[1] Please email all corrections and suggestions to these notes to `A.Iserles@damtp.cam.ac.uk`. All handouts are available on the WWW at the URL `http://www.damtp.cam.ac.uk/user/na/PartII/Handouts.html`.

There are standard ways of giving A this convenient form which will be considered later. This and our next topic, *deflation,* depend on the following basic result.

Theorem 2.10 *Let A and S be $n \times n$ matrices, S being nonsingular. Then $\boldsymbol{w}$ is an eigenvector of A with eigenvalue λ if and only if $S\boldsymbol{w}$ is an eigenvector of SAS^{-1} with the same eigenvalue.*

Proof

$$A\boldsymbol{w} = \lambda \boldsymbol{w} \qquad \Leftrightarrow \qquad AS^{-1}(S\boldsymbol{w}) = \lambda \boldsymbol{w} \qquad \Leftrightarrow \qquad (SAS^{-1})(S\boldsymbol{w}) = \lambda(S\boldsymbol{w}).$$

□

Definition 2.11 (Deflation). Suppose that we have found one solution of the eigenvector equation $A\boldsymbol{w} = \lambda\boldsymbol{w}$, where A is again $n \times n$. Then *deflation* is the task of constructing an $(n-1)\times(n-1)$ matrix, B say, whose eigenvalues are the other eigenvalues of A. Specifically, we apply a similarity transformation S to A such that the first column of SAS^{-1} is λ times the first coordinate vector, because it follows from the characteristic equation for eigenvalues and from Theorem 2.10 that we can let B be the bottom right $(n-1)\times(n-1)$ submatrix of SAS^{-1}.

We write the condition on S as $(SAS^{-1})\boldsymbol{e}_1 = \lambda\boldsymbol{e}_1$. Therefore the last part of the proof of Theorem 2.10 shows that it is sufficient if S has the property $S\boldsymbol{w} = c\boldsymbol{e}_1$, where c is any nonzero scalar.

Technique 2.12 (Algorithm for deflation for symmetric A). Suppose that A is symmetric and $\boldsymbol{w} \in \mathbb{R}^n \setminus \{\boldsymbol{0}\}$, $\lambda \in \mathbb{R}$ are given so that $A\boldsymbol{w} = \lambda\boldsymbol{w}$. We seek a nonsingular matrix S such that $S\boldsymbol{w} = c\boldsymbol{e}_1$ and such that SAS^{-1} is also symmetric. The last condition holds if S is orthogonal, since then $S^{-1} = S^\top$. It is suitable to pick a *Householder reflection,* which means that S has the form $I - 2\boldsymbol{u}\boldsymbol{u}^\top/\|\boldsymbol{u}\|^2$, where $\boldsymbol{u} \in \mathbb{R}^n \setminus \{\boldsymbol{0}\}$. Specifically, we recall from the ***D3 Numerical Analysis*** course that Householder reflections are orthogonal and that, for any nonzero $\boldsymbol{w} \in \mathbb{R}^n$, a real vector $\boldsymbol{u}$ can be found with the property

$$\left(I - 2\frac{\boldsymbol{u}\boldsymbol{u}^\top}{\|\boldsymbol{u}\|^2}\right)\boldsymbol{w} = \pm\|\boldsymbol{w}\|\boldsymbol{e}_1. \tag{2.2}$$

Thus, we let $u_i = w_i$ for $i = 2, 3, \ldots, n$ and choose u_1 so that $2\boldsymbol{u}^\top\boldsymbol{w} = \|\boldsymbol{u}\|^2$. Since

$$\left(I - 2\frac{\boldsymbol{u}\boldsymbol{u}^\top}{\|\boldsymbol{u}\|^2}\right)\boldsymbol{w} = \boldsymbol{w} - 2\frac{\boldsymbol{u}^\top\boldsymbol{w}}{\|\boldsymbol{u}\|^2}\boldsymbol{u} = \boldsymbol{w} - \boldsymbol{u},$$

it follows that (2.2) holds for components $i = 2, 3, \ldots, n$. Finally, we compute u_1 so that $2\boldsymbol{u}^\top\boldsymbol{w} = \|\boldsymbol{u}\|^2$ is true. Since for our $\boldsymbol{u}$ it is true that $\boldsymbol{u}^\top\boldsymbol{w} = \|\boldsymbol{w}\|^2 + (u_1w_1 - w_1^2)$ and $\|\boldsymbol{u}\|^2 = \|\boldsymbol{w}\|^2 + u_1^2 - w_1^2$, we deduce that $u_1^2 - 2u_1w_1 + w_1^2 = \|\boldsymbol{w}\|^2$, hence $u_1 = w_1 \pm \|\boldsymbol{w}\|$ and (2.2) is obeyed also for $i = 1$.

Since the bottom $n-1$ components of $\boldsymbol{u}$ and $\boldsymbol{w}$ coincide, the calculation of $\boldsymbol{u}$ requires only $\mathcal{O}(n)$ computer operations. Further, the calculation of SAS^{-1} can be done in only $\mathcal{O}(n^2)$ operations, taking advantage of the form $S = I - 2\boldsymbol{u}\boldsymbol{u}^\top/\|\boldsymbol{u}\|^2$, even if all the elements of A are nonzero. After deflation, we may find an eigenvector, $\hat{\boldsymbol{w}}$ say, of SAS^{-1}. Then the corresponding eigenvector of A, given in Theorem 2.10, is $S^{-1}\hat{\boldsymbol{w}} = S\hat{\boldsymbol{w}}$, because Householder matrices, like all symmetric orthogonal matrices, are *involutions:* $S^2 = I$.

Technique 2.13 (Algorithm for deflation when A is nonsymmetric). There is a faster algorithm for deflation in the nonsymmetric case (of course, it applies also to symmetric matrices). Let w_i, $i = 1, 2, \ldots, n$, be the components of the eigenvector $\boldsymbol{w}$ and assume $w_1 \neq 0$ (which can be achieved by reordering the variables if necessary). Further, we let S be the $n \times n$ matrix whose elements agree with those of the $n \times n$ unit matrix, except that the off-diagonal part of the first column of S has the elements $S_{i1} = -w_i/w_1$, $i = 2, 3, \ldots, n$. Then S is nonsingular and has the property $S\boldsymbol{w} = w_1\boldsymbol{e}_1$, so it is suitable for our purpose. Moreover, the elements of S^{-1} also agree with those of the $n \times n$ unit matrix, except that $(S^{-1})_{i,1} = +w_i/w_1$, $i = 2, 3, \ldots, n$. These forms of S and S^{-1} allow the calculation of SAS^{-1}, and hence B, to be done in only $\mathcal{O}(n^2)$ operations. Further, because the last $n-1$ columns of SAS^{-1} and of SA are the same, B is just the bottom right $(n-1)\times(n-1)$ submatrix of SA. Therefore, for every integer $1 \le i \le n-1$ we form the ith row of B in the following way: subtract w_i/w_1 times the first row of A from the $(i+1)$th row of A, and ignore the first component of the resultant row vector.