Exercise session I: Krylov subspaces, Lanczos algorithms

Numerical Linear Algebra

13 October 2017

This exercise session is divided in two parts: at beginning there is a theoretical background with some question for better understanding the subject. In the second part you will work with MATLAB/Octave, on computational exercises.

1 Theoretical Background

The first part (Section 1.1) is devoted to have some insights on Krylov subspaces and the questions are indicated in the following way: ***questions?***. The second part (Section 1.2, devoted to iterative algorithms for eigenvalue problem) is preparatory for the computational exercises (Section 2).

1.1 Krylov subspaces

Suppose we have a vector b and a blackbox $x \longmapsto Ax$. ***Think an example where*** A ***may not be represented explicitly as a matrix but may be available only as a subroutine for computing*** Ax**.** Let $y_1 := b$, and compute a sequence of matrix vector products $y_i := Ay_{i-1}$. ***Do we get new information with every additional*** y_k***?***

Let us first consider the case where we perform n matrix vector products, where n is the size of the matrix A. If K is the matrix $[y_1, y_2, \ldots, y_n]$, we have

$$AK = [Ay_1, \ldots, Ay_{n-1}, Ay_n] = [y_2, \ldots, y_n, A^n y_1].$$

Suppose that K is invertible (***When is*** K ***singular?***), then compute a vector $c = -K^{-1}A^n y_1$. Now

$$AK = K[e_2, e_3, \ldots, e_n, -c] = KC,$$

where

$$K^{-1}AK = C = \begin{bmatrix} 0 & 0 & \cdots & 0 & -c_1 \\ 1 & 0 & \cdots & 0 & -c_2 \\ 0 & 1 & \cdots & 0 & -c_3 \\ \vdots & \vdots & \ddots & 0 & \vdots \\ 0 & 0 & \cdots & 1 & -c_n \end{bmatrix}$$

This nice procedure has the following problems:

1. inverting of K is not better than inverting of A, ***Why?***
2. K is (almost always) ill-conditioned, ***Why?***

Replace K with an orthogonal Q such that the first columns of K and of Q spans the same space. Orthogonal matrices are well-conditioned and easily invertible. Later we will see that we only care about the first columns of Q. Let us use a QR decomposition of K and write $K = QR$. Then

$$\begin{aligned} K^{-1}AK &= (R^{-1}Q^T)A(QR) = C, \quad \text{and} \\ Q^TAQ &= RCR^{-1} = H. \end{aligned} \tag{1}$$

Since R and R^{-1} are upper triangular and C is upper Hessenberg, it is easy to show that H is also upper Hessenberg. (In other words, we have transformed A to upper Hessenberg form using orthogonal transformations).

Let us show now how to compute the columns $q_1, \ldots, q_n$ of Q step-by-step. Since $Q^TAQ = H$, we have $AQ = QH$. This gives

$$Aq_j = \sum_{i=1}^{j+1} q_i h_{ij}. \tag{2}$$

Recall that vectors q_i are orthonormal and multiply each part of (2) by q_m^T. This gives

$$q_m^T A q_j = \sum_{i=1}^{j+1} h_{ij} q_m^T q_i = h_{mj} \quad \text{for} \quad 1 \le m \le j.$$

We also have

$$h_{j+1,j} q_{j+1} = Aq_j - \Sigma_{i=1}^{j} h_{ij} q_i.$$

This yields the following algorithm:

Algorithm 1: Arnoldi algorithm for (partial) reduction to Hessenberg form

```
begin
    q_1 = b/||b||_2;
    /* k is the number of columns of Q and H to compute.          */
    for j = 1 : k do
        z = A q_j;
        for i = 1 : j do
            h_ij = q_i^T z;
            z = z - h_ij q_i;
        end
        h_{j+1,j} = ||z||_2;
        if h_{j+1,j} = 0 then
            exit
        end
        q_{j+1} = z/h_{j+1,j};
    end
end
```

Vectors q_i are called *Arnoldi vectors*. The internal loop is a *modified Gram-Schmidt algorithm*. ***When does*** $z = 0$ ***occur?***

Sometimes the inner loop has to be used twice to get orthogonal vectors. ***Change the algorithm to do two reothogonalizations.***

1.1.1 Special case: A symmetric

Let A be symmetric. Then H is also symmetric. Since H is also Hessenberg, H is tridiagonal. Let us denote in this case H by T,

$$T = T_n = \begin{bmatrix} \alpha_1 & \beta_1 & & \\ \beta_1 & \ddots & \ddots & \\ & \ddots & \ddots & \beta_{n-1} \\ & & \beta_{n-1} & \alpha_n \end{bmatrix}.$$

The equality $AQ = QT$ gives that

$$Aq_j = \beta_{j-1}q_{j-1} + \alpha_j q_j + \beta_j q_{j+1}.$$

Multiply this equation by q_j^T and recall the orthogonality of Q. This gives an updated version of Arnoldi algorithm, which is called *Lanczos algorithm.*

Algorithm 2: Lanczos algorithm for (partial) reduction to tridiagonal form

```
begin
    q_1 = b/‖b‖_2; β_0 = 0; q_0 = 0;
    /* k stays for the number of columns in Q and H to be determined.  */
    for j = 1 : k do
        z = Aq_j;
        α_j = q_j^T z;
        z = z − α_j q_j − β_{j−1} q_{j−1};
        β_j = ‖z‖_2;
        if β_j = 0 then
            exit;
        end
        q_{j+1} = z/β_j;
    end
end
```

We call a span of vectors $b, Ab, A^2b, \ldots, A^{k-1}b$ a *Krylov subspace* $\mathcal{K}_k(A, b)$.

If the Arnoldi (Lanczos) algorithm does not stop early ($z = 0$), then columns of Q_k form the basis in $\mathcal{K}_k(A, b)$.

What did we learn about A ***after*** k ***step of Algorithm 1? and after*** k ***step of Algorithm 2?*** *(Hint. Divide the matrix Q in the first k and the last $u = n - k$ columns, and analyze Q^TAQ.* **What are the first leading** k **columns of** K **and** Q **of equation** (1)**?***)*

1.2 Iterative algorithms for Eigenvalue problems

1.2.1 Rayleigh-Ritz procedure

Consider the spectral problem (with symmetric or Hermitian) $Az = \lambda z$. Let us choose some space $\mathcal{L}$ and let us look for (approximations to) eigenvectors in this space. Denote an approximate eigenvector by $y \in \mathcal{L}$ and its corresponding eigenvalue by θ. We choose y and θ in such a way that the residual

$$r(\theta, y) = Ay - \theta y \tag{3}$$

is orthogonal to $\mathcal{L}$. Let $q_1, \ldots, q_k$ be some orthonormal basis of $\mathcal{L}$ and let $Q_k = [q_1, \ldots, q_k]$. Then we can write $y = Q_k s$. The condition $r(\theta, y) \perp \mathcal{L}$ gives that

$$(Q_k^T A Q_k)s = \theta s \quad \textbf{\textit{Prove this}}. \tag{4}$$

Thus the approximations θ_i are the eigenvalues of a reduced spectral problem (4), and their corresponding eigenvectors could be computed using the formula

$$y_i = Q_k s_i. \tag{5}$$

Values θ_i and vectors y_i are called *Ritz values* and *Ritz vectors*, correspondingly, for subspace $\mathcal{L}$.

Let us take $\mathcal{K}_k$ as $\mathcal{L}$ for some Hermitian matrix A and let $q_1, \ldots, q_k$ be its orthonormal basis formed by Lanczos vectors. ***Prove that the following equality holds***:

$$AQ_k = Q_k T_k + R_k, \tag{6}$$

where $R_k = r_k e_k^T$, and $Q_k^T r_k = 0$. ***Can you find an explicit formula for*** r_k ***in terms of already computed quantities?*** Multiplying (6) on the left by Q_k^T gives $Q_k^T A Q_k = T_k$. Thus the eigenvalues of T_k are the Ritz values for A and could be considered as approximations of eigenvalues of A.

Theorem 1 (Convergence - ([8], [6], [7], [5] and others).) *Suppose that all the eigenvalues of A are simple (i.e. different) and ordered decreasingly: $\lambda_1 > \lambda_2 > \cdots > \lambda_n$. Let us order Ritz values in the same way: $\theta_1 \geq \theta_2 \geq \cdots \geq \theta_k$. Suppose that an index of the starting vector b is equal to n. Let us define the relative apartness of eigenvalues as*

$$\gamma_j = \frac{\lambda_j - \lambda_{j+1}}{\lambda_{j+1} - \lambda_n}.$$

When $j \in \{1, \ldots, k\}$ is fixed and k is increasing, $\theta_j^{(k)}$ and $y_j^{(k)}$ converge to λ_j and z_j, where $Az_j = \lambda z_j$. The speed of convergence is almost geometric with the denominator $1 - 2\sqrt{\gamma_j}$ for eigenvectors and $(1 - 2\sqrt{\gamma_j})^2$ for eigenvalues.

When k is growing we can find better and better approximations to eigenvalues and eigenvectors of A in k-th Krylov subspace. Thus Lanczos algorithm is suitable for solution of an incomplete eigenvalue problem. (Do several steps of Lanczos, then compute the eigenvalues of T_k by means of any algorithm, then compute eigenvectors by means of (5).) The quality of approximations could be estimated on-line by computing norms of corresponding residuals.

Let (θ, s) be the eigenpair for T_k: $T_k s = \theta s$. Multiplying (6) on the right by s, we get

$$AQ_k s = Q_k T_k s + R_k s = \theta Q_k s + s_{(k)} r_{k+1}, \tag{7}$$

where $s_{(k)}$ denotes the last component of k-vector s. Thus the residual of Ritz vector $y = Q_k s$ is equal to $r(\theta, y) = s_{(k)} r_{k+1}$ (***Prove it***), and its length is equal to

$$\|r(\theta, y)\|_2 = \beta_k |s_{(k)}|, \tag{8}$$

Prove it using the information derived by Algorithm 2.

Thus for computing only the length of the residual there is no need to build Ritz vector. This is very important since otherwise we would have to store all Lanczos vectors q_i even in the case of searching only for the eigenvalues.

1.2.2 Lanczos Algorithm in Exact Arithmetic

In finite arithmetics Lanczos and Arnoldi algorithm can behave differently. Vectors q_i very often loose orthogonality. Full or partial reorthogonalization could be applied.

Algorithm 3: Lanczos algorithm for computing eigenvalues and eigenvectors of $A = A^T$ with(**I**) or without(**II**) full reorthogonalization

begin
  $q_1 = b/\|b\|_2$; $\beta_0 = 0$; $q_0 = 0$;
  `/* k stays for the number of columns in` Q `and` H `to be determined */`
  for $j = 1 : k$ **do**
    $\alpha_j = q_j^T z$;
    I: $z = z - \Sigma_{i=1}^{j-1}(z^T q_i)q_i$;$z = z - \Sigma_{i=1}^{j-1}(z^T q_i)q_i$;
    II:$z = z - \alpha_j q_j - \beta_{j-1}q_{j-1}$;
    $\beta_j = \|z\|_2$;
    if $\beta_j = 0$ **then**
      exit
    end
    $q_{j+1} = z/\beta_j$;
    Compute eigenvalues and eigenvectors of T_j and corresponding residuals;
  end
end

The version with full reorthogonalization behaves very similar to an infinite-precision implementation.

More deep insight you can get from Jim Demmel's book [1].

2 Computational exercises

(Very plain) implementation of these methods you can download as **Lanczos-software** from the course Toledo page. These implementations also plot some pictures corresponding to the errors of the algorithm. You can estimate different types of errors for the eigenvalues:

1. *Global (relative) errors* $|\lambda_i(T_k) - \lambda_i(A)|/|\lambda_i(A)|$. We expect that this global error would tend to machine precision.
2. *Local (relative) errors* $\min_j |\lambda_i(T_k) - \lambda_j(A)|/|\lambda_j(A)|$. This measures the distance between Ritz value and the closest eigenvalue of A.
3. *Error bounds* using (8) $\|r(\theta_i, y_i)\|_2/|\lambda_i(A)|$.

Exercise 1. Start with the matrix provided by Jim Demmel `LanczosDemo.m` and with the procedure using full re-orthogonalization `LanczosFullReorthog.m`. Answer the following questions for the biggest and the smallest eigenvalues separately.

- Which eigenvalues converge faster?
- Is error estimation by (8) good?
- Look at the convergence of the second smallest eigenvalue. Try to find in the code some explanation for this behavior.
- Check carefully (probably magnify the picture) the convergence to λ_{998} and λ_{997}.

Exercise 2. Answer the same questions for Lanczos procedure with no orthogonalizatio `LanczosNoReorthog.m`. Plot also the smallest singular value of Q_m. (It helps you to discover the loss of orthogonality.) Try to find extra 'ghost' converged eigenvalues.

Exercise 3. Construct your own very small (symmetric or diagonal) matrix. Try to discover the influence of a starting vector b on the convergence. Select

- a random vector as b,
- an eigenvector as b,
- a special vector that has zero component in the direction of some eigenvector as b.

Exercise 4. (Life is harder.) Consider a matrix where the eigenvalues are very much concentrated at the extremes. For example, you can take as the eigenvalue vector e:

```
e1 = (0.3/N: 2/N : 1).^4;
e = sort([-1+e1 1-e1]);
```

for some N.

Exercise 5. Repeat the steps above for Strakos matrices. These are the matrices with eigenvalues of the form

$$\lambda_i = \lambda_1 + \frac{i-1}{n-1}(\lambda_n - \lambda_1)\rho^{n-i}, \quad i = 2, \ldots, n, \quad \rho \in (0, 1),$$

where n, λ_1 and $\kappa = \lambda_n/\lambda_1$ are fixed. Try $n = 24$, $\lambda_1 = 0.1$, $\kappa = 1000$ and $\rho = 0.4, 0.6, 0.8, 1.0$.

The last two exercises deals with how Lanczos algorithms locate the Ritz values w.r.t. the eigenvalues of the original matrix. For further details on this topic, refer to [4, 3] and Lecture 34 in [2].

References

[1] J. W. Demmel, *Applied Numerical Linear Algebra*, SIAM, Philadelphia, Pennsylvania, USA, 1997.

[2] L. N. Trefethen and D. Bau, *Numerical Linear Algebra*, SIAM, Philadelphia, Pennsylvania, USA, 1997.

[3] A. B. J. Kuijlaars, *Convergence analysis of Krylov subspace iterations with methods from potential theory*, SIAM Review, 48 (2006), pp. 3–40.

[4] A. B. J. Kuijlaars, *Which eigenvalues are found by the Lanczos method?*, SIAM J. Matrix Anal. Appl., 22 (2000), pp. 306–421.

[5] V. Druskin and L. Knizhnerman, *Krylov subspace approximation of eigenpairs and matrix functions in exact and computer arithmetic*, Numerical Linear Algebra with Applications, 2 (1995), pp. 205–217.

[6] C. C. Paige, *Error analysis of the Lanczos algorithm for tridiagonalizing a symmetric matrix*, Journal of the Institute of Mathematics and its Applications, 18 (1976), pp. 341–349.

[7] C. C. Paige, B. N. Parlett, and H. A. van der Vorst, *Approximate solutions and eigenvalue bounds from Krylov subspaces*, Numerical Linear Algebra with Applications, 2 (1995), pp. 115–133.

[8] Y. Saad, *On the rates of convergence of the Lanczos and the block-Lanczos methods*, SIAM Journal on Numerical Analysis, 17 (1980), pp. 687–706.