SF3580

Anna Broms & Fredrik Fruklund

2018/11/11

1 Task 2

1.1 (a)

Insert figure

1.2 (b)

Insert figure

1.3 (c)

Insert figure

The Rayleigh quotient only uses the symmetric part of A in

$$r(\mathbf{x}) = \mathbf{x}^H A \mathbf{x}$$

assuming $\mathbf{x}$ is normalized. The matrix A is no longer symmetric, i.e. $A \neq A^H$, but any square matrix can be decomposed into a symmetric part A_s and a nonsymmetric part A_{ns} by

$$A = \underbrace{\frac{1}{2}\left(A + A^H\right)}_{=A_s} + \underbrace{\frac{1}{2}\left(A - A^H\right)}_{=A_{ns}}.$$

Thus

$$r(\mathbf{x}) = \mathbf{x}^H A_s \mathbf{x} + \mathbf{x}^H A_{ns} \mathbf{x} = \mathbf{x}^H A_s \mathbf{x}$$

since

$$\mathbf{x}^H A_{ns} \mathbf{x} = \mathbf{x}^H A \mathbf{x} - \mathbf{x}^H A^H \mathbf{x} = 0.$$

For a nonsymmetric matrix all avaliable information is not used.

2 Task 3

The performance of different versions of Gram-Schmidt orthogonalizations is investigated when combined with the Arnoldi method. We consider the matrix A constructed by

```
Random.seed!(0); A=matrixdepot("wathen",nn,nn)
```
(1)

where we choose $nn = 500$ and use m number of iterations in the Arnoldi method.

Results for CPU time and orthogonality of the basis in Q is given in Table 1.

Table 1: Comparison for different types of Gram-Schmidt (GS) orthogonalisation in the Arnoldi method: SGS (single GS), MDS (modified GS), DGS (double GS), TGS (triple GS), where *time* is the measured CPU time and *orth* is the orthogonality of the basis in terms of $\|Q_m^T Q_m - I\|$.

m	SGS time	SGS orth	MGS time	MGS orth	DGS time	DGS orth	TGS time	TGS orth
5	188 ms	$4.82 \cdot 10^{-13}$	241 ms	$3.28 \cdot 10^{-15}$	246 ms	$3.81 \cdot 10^{-15}$	298 ms	$3.84 \cdot 10^{-15}$
10	496 ms	$2.15 \cdot 10^{-12}$	705 ms	$1.32 \cdot 10^{-14}$	665 ms	$4.19 \cdot 10^{-15}$	820 ms	$4.23 \cdot 10^{-15}$
20	1.10 s	$1.04 \cdot 10^{-11}$	2.03 s	$6.41 \cdot 10^{-14}$	1.682 s	$5.66 \cdot 10^{-15}$	2.16 s	$5.72 \cdot 10^{-15}$
50	4.05 s	$1.06 \cdot 10^{-10}$	9.95 s	$6.04 \cdot 10^{-13}$	6.69 s	$6.18 \cdot 10^{-15}$	9.49 s	$6.38 \cdot 10^{-15}$
100	12.0 s	$4.77 \cdot 10^{-10}$	37.8 s	$3.29 \cdot 10^{-12}$	21.7 s	$7.82 \cdot 10^{-15}$	31.2 s	$7.87 \cdot 10^{-15}$

It can be concluded that double Gram-Schmidt performs the best in terms of orthogonalization error, while single Gram-Schmidt is the fastest among the algorithms. Triple Gram-Schmidt performs almost exactly as well as double Gram-Schmidth in terms of error, which is not surprising. It was stated during the lecture that the best result possible to achieve using multiple Gram-Schmidt is indeed obtained for double Gram-Schmidt. Note however that the CPU time required for triple Gram-Schmidt of course is considerably larger than for double Gram-Schmidt. The orthogonalization errors for the modified Gram-Schmidt is smaller than for single Gram-Schmidt but the algorithm is in this example seen to be worse than double Gram-Schmidt both in terms of CPU time and orthogonalization error.

3 Task 4

We investigate a primitive version of the Arnoldi method. Let K_m be a matrix representing the Krylov subspace:

$$K_m = [b, Ab/\|Ab\|, \ldots, A^{m-1}b\|A^{m-1}b\|] \in \mathbb{R}^{n \times m}. \tag{2}$$

3.1 (a)

The equation

$$\mu K_m^T K_m w = K_m^T A K_m w \tag{3}$$

is stemming from the Galerking method applied to the bilinear form associated with the eigenvalue problem $a(u, v) = u^T Av - \mu u^T v$*, with* $f(v) = 0$*. Prov that* (3) *is identical to the approximation generated by Arnoldi's method for eigenvalue problems.*

The Arnoldi method computes an orthogonal basis of K_m such that after m iterations

$$AQ_m = Q_{m+1} H_{\underline{m}}, \tag{4}$$

where Q_m is an orthogonal matrix of size m and $H_{\underline{m}}$ is the corresponding Hessenberg matrix. The eigenvalues of A can then be approximated by the eigenvalues of

$$Q_m^T A Q_m. \tag{5}$$

We use the QR-factorisation of the matrix K_m, such that $K_m = Q_m R$. We first show that $R^T R = K_m^T K_m$. Using orthogonality, we have

$$I = Q_m^T Q_m. \tag{6}$$

Multiplying both sides of (6) with R^T and R respectively yields

$$\begin{aligned} R^T = R^T Q^T Q_m = K_m^T Q_m \Leftrightarrow \\ R^T R = K_m^T Q_m R = K_m^T K_m. \end{aligned} \tag{7}$$

Now, considering (3), we have

$$\begin{aligned} \mu K_m^T K_m w = K_m^T A K_m w \Leftrightarrow \\ \mu R^T R w = K_m^T A K_m w. \end{aligned} \tag{8}$$

Replacing K_m and K_m^T with the QR-factorisation of K_m yields

$$\mu R^T R w = R^T Q_m^T A Q_m R w. \tag{9}$$

Using that R is non-singular, we obtain

$$\mu w = Q_m^T A Q_m w. \tag{10}$$

Thus, the approximation computed from (3) is identical to the eigenvalue approximation obtained from (5).

4 Task 6

test

5 Task 8

test