TEKNISKA HÖGSKOLAN I LINKÖPING
Matematiska institutionen
Beräkningsmatematik/Fredrik Berntsson

Exam TANA15 Numerical Linear Algebra, Y4, Mat4

Datum: 23:e Mars, 2021.

Hjälpmedel:

1. Föreläsningsanteckningar utskrivna från kurshemsidan utan egna anteckningar.
2. Räknedosa i fickformat, med nollställt minne och utan instruktionsbok.

Examinator: Fredrik Berntsson

Maximalt antal poäng: 25 poäng. För godkänt krävs 10 poäng.

Jourhavandelärare Fredrik Berntsson - (telefon 013 28 28 60)

Good luck!

(5p) **1:** Do the following:

a) Let $\|\cdot\|$ be a vector norm. Clearly state the definition of the matrix norm *induced* by the vector norm. Also show that for all matrix norms *induced* by a vector norm we have $\|I\| = 1$, where I is the identity.

b) Show that for all *induced* matrix norms the submultiplicative property, $\|AB\| \leq \|A\|\|B\|$, holds. Also show that $\|A^{-1}\|\|A\| \geq 1$.

c) Prove the inequality $\|x\|_\infty \leq \|x\|_2 \leq \sqrt{n}\|x\|_\infty$.

d) Let $(\cdot,\cdot)$ be a scalar product and $\|\cdot\|$ the corresponding vector norm. Show that if P is an orthogonal projection, with respect to $(\cdot,\cdot)$, then $\|Px\| \leq \|x\|$.

(3p) **2:** Consider the matrix

$$A = \begin{pmatrix} 2.3 & -0.2 & 0.3 \\ 0.7 & -5.3 & 0.5 \\ 1.1 & -0.4 & 1.7 \end{pmatrix}$$

with eigenvalues $\lambda_1 = 2.6095$, $\lambda_2 = 1.3466$, and $\lambda_3 = -5.2561$. We want to use power-iteration to compute an approximate eigenvalue of A. The rate of convergence is defined as,

$$\gamma_k = \frac{|\lambda^{(k+1)} - \lambda|}{|\lambda^{(k)} - \lambda|}$$

where λ is the exact eigenvalue. The asymptotic rate of convergence is $\gamma = \lim_{k\to\infty} \gamma_k$.

a) If we apply power iteration to the matrix A. To which eigenvalue will the iterations converge? Also give a good theoretical estimate of the asymptotic rate of convergence.

b) Let $s = 0.8$ and apply power iteration to the matrix $(A - sI)^{-1}$. To which eigenvalue of A will we have convergence now? Also estimate the asymptotic rate of convergence for this case.

c) Let $s = 4$ and apply power iteration to $A + sI$. To which eigenvalue of A will we have convergence now? Also estimate the asymptotic rate of convergence for this case.

(4p) **3:** Let A be an $m \times n$ matrix, where $m >> n$. Do the following

a) A Householder reflection can be written as

$$H = I - 2uu^T,$$

where $\|u\|_2 = 1$. Demonstrate how the product of HA can be computed as efficiently as possible and estimate the amount of arithmetic work needed.

b) Use the result from **a)** to estimate the total amount of arithmetic work required for computing both the R and the Q matrices in the *full* QR decomposition of the matrix A.

Hint Use $m >> n$ to simplify the expression for the amount of work required.

(4p) **4:** **a)** Suppose A is an $n \times n$ matrix and let (λ_1, x_1) be one eigenpair. Clearly demonstrate how an orthogonal matrix Q such that

$$Q^T A Q = \begin{pmatrix} \lambda_1 & w^T \\ 0 & B \end{pmatrix},$$

where B is an $(n-1) \times (n-1)$ matrix, can be found. Also clearly demonstrate that your proposed matrix solves the problem, i.e. $Q^T A Q$ has the correct structure. *It is required to present a detailed proof and not just make a reference to a lemma.*

b) The Hessenberg decomposition $A = QHQ^T$, where Q is orthogonal and H is a Hessenberg matrix, can be computed using a sequence of Householder reflections. Clearly show how Housedolder reflections can be used to reduce a matrix into Hessenberg form by a sequence of similarity transformations. It is enough to consider the 4×4 case.

(4p) **5:** Consider the system of equations

$$\begin{aligned} &(x_1 - 1)^2 + 3x_2 - 3 = 0, \\ &\cos(x_1) + (x_3 - 1)^2 - 1 = 0, \\ &x_1 + x_2^2 + (x_3 + 1)^2 - 2 = 0. \end{aligned}$$

a) Describe how the Newton method can be used for solving the above problem. What is the function $f(x)$ and the Jacobian $J_f(x)$?

b) If the Jacobian is difficult to compute we can use updating methods. Suppose B_k is the approximation of the Jacobian that is used at step k. In Broydens method we find the next approximation by a rank one update $B_{k+1} = B_k + uv^T$ so that the formula

$$B_{k+1} s^{(k)} = f(x^{(k+1)}) - f(x^{(k)}),$$

is satisfied. Clearly show how to find the update uv^T, with the smallest norm $\|uv^T\|_2$, that satisfies the above relation. *You need to present a proof that your suggested update has the required properties.*

(5p) **6:** Let A be symmetric and positive definite. Consider a projection method, for solving a linear system $Ax = b$, where at each step $\mathcal{K} = \mathcal{L} = \text{span}(r, Ar)$, and $r = b - Ax$ is the current residual. Do the following:

a) As basis for $\mathcal{K}$ we use r and a vector p obtained by orthogonalizing Ar against r with respect to the A-inner product. Derive a formula for computing p.

b) Write down the algorithm for performing the projection step using the subspace $\mathcal{K}$. What is the minimum number of multiplications by the A matrix in each step?

1: For **a)** the matrix norm is defined by

$$\|A\| = \max_{x \neq 0} \frac{\|Ax\|}{\|x\|}.$$

From the definition we obtain $\|I\| = \max_{x\neq 0} \|Ix\|/\|x\| = \max_{x\neq 0} \|x\|/\|x\| = 1$.

For **b)** we use the definition to obtain

$$\|AB\| = \max_{x\neq 0} \frac{\|ABx\|}{\|x\|} = \max_{x\neq 0} \frac{\|ABx\|}{\|Bx\|}\frac{\|Bx\|}{\|x\|} \leq \max_{x\neq 0} \frac{\|ABx\|}{\|Bx\|}\|B\| \leq \max_{y\neq 0} \frac{\|Ay\|}{\|y\|}\|B\| \leq \|A\|\|B\|.$$

Now we can use the submultiplicative property to show $1 = \|I\| = \|AA^{-1}\| \leq \|A\|\|A^{-1}\|$.

For **c)** we recall that $\|x\|_\infty = \max |x_i|$. Thus, if $|x_k|$ is the largest element of x,

$$\|x\|_\infty = |x_k| = \left(|x_k|^2\right)^{1/2} \leq \left(|x_1|^2 + \ldots + |x_n|^2\right)^{1/2} = \|x\|_2.$$

Also

$$\|x\|_2 = \left(|x_1|^2 + \ldots + |x_n|^2\right)^{1/2} \leq \left(|x_k|^2 + \ldots + |x_k|^2\right)^{1/2} = \left(n|x_k|^2\right)^{1/2} = \sqrt{n}\|x\|_\infty.$$

Finally, for **d)** we observe that $x = (I-P)x + Px = x_1 + x_2$, where x_1 is orthonal to x_2. Thus $\|x\|^2 = (x,x) = (x_1+x_2, x_1+x_2) = (x_1,x_1) + 2(x_1,x_2) + (x_2,x_2) = \|x_1\|^2 + 0 + \|x_2\|^2$. This is really the Phytagorean theorem. Thus $\|x\| \geq \|x_2\| = \|Px\|$.

2: For **a** we note that $|\lambda_3|$ is the largest eigenvalue and $|\lambda_1$ is the second largest. Thus $\gamma = |\lambda_1/\lambda_3| = 2.6095/5.2561 = 0.4965$.

For **b** we introduce $B = (A - sI)^{-1}$ and note that if λ is an eigenvalue of A then $\mu = 1/(\lambda - s)$ is an eigenvalue of B. This means that the eigenvalues of B are $\mu_1 = 0.5526$, $\mu_2 = 1.8295$ and $\mu_3 = -0.1651$. Thus $\gamma = |\mu_1/\mu_2| = 0.5526/1.8295 = 0.3020$. We have convergence to the eigenvalue μ_2, or to $\lambda_2 = 1.3466$.

For **c)** we similarily observe that if $B = A + sI$, where $s = 4$, then the eigenvalues of B are $\mu_1 = 6.6095$, $\mu_2 = 5.3466$ and $\mu_3 = -1.2561$. Thus we have convergence towards $\lambda_1 = 2.6095$ and the rate of convergence is $\gamma = |\mu_2/\mu_1| = 5.3466/6.6095 = 0.8089$.

3: For **a)** we need to compute

$$HA = (I - 2uu^T)A = A - 2u(u^TA).$$

First $y = u^TA$ is a matrix-vector multiply that requires $2mn$ floating point operations. Second we have to compute the outer product $B = uy^T$. This again requires mn multiplications (we ignore the 2 as that could be included in the y matrix using

n operations). Finally $A - B$ is computed using mn subtractions. The total is this $4mn$ floating point operations.

For **b)** we just need to recall that in step k of the Householder algorithm we need to apply a reflection H_k to the block $A(k:m, k:n)$, of size $(m-k+1)\times(n-k+1)$, and will get R after n steps. This means that the total amount of work is

$$\sum_{k=1}^{n} 4(m-k+1)(n-k+1) \approx 4m\sum_{k=1}^{n}(n-k+1) \approx 4mn(n/2).$$

where we used the asumption $m >> n$ to obtain $m-k+1 \approx m$. Otherwise we need to look up the sum in a table.

Similarily, to get the full Q we need to start with the identity matrix I, of size $m \times m$, and apply H_k to the block $Q(k:m, 1:m)$, which is of size $(m-k+1)\times m$. The work is thus

$$\sum_{k=1}^{n} 4(m-k+1)m \approx 4m^2 n,$$

where again $m >> n$ was used. To conclude $2mn^2$ operations needed for R and $4m^2n$ needed for the full Q.

4: For **a)** We have the eigenpair (λ_1, x_1). If we compute the full QR decomposition of $x_1 \in \mathbb{R}^{n\times 1}$ we obtain an orthogonal matrix suxch that $Q = (x_1, Q_2)$, where $Q_2^T x_1 = 0$. This is assuming that $\|x_1\|_2 = 1$. We find that

$$Q^T A Q = (x_1, Q_2)^T A (x_1, Q_2) = (x_1, Q_2)^T (Ax_1, AQ_2) = (x_1, Q_2)^T(\lambda_1 x_1, AQ_2) =$$

$$\begin{pmatrix} \lambda_1 x_1^T x_1 & x_1^T A Q_2 \\ \lambda_1 Q_2^T x_1 & Q_2^T A Q_2 \end{pmatrix} = \begin{pmatrix} \lambda_1 & w^T \\ 0 & B \end{pmatrix},$$

where we have the correct structure.

For **b)** we illustrate the algorithm as follows: First we use the same reflection H_1 applied from the left and from the right. The reflection is selected so the elements $A(3:4,1)$ are set to zero. We get

$$H_1 \begin{pmatrix} x & x & x & x \\ x & x & x & x \\ x & x & x & x \\ x & x & x & x \end{pmatrix} H_1^T = \begin{pmatrix} x & x & x & x \\ + & + & + & + \\ 0 & + & + & + \\ 0 & + & + & + \end{pmatrix} H_1^T = \begin{pmatrix} x & + & + & + \\ x & + & + & + \\ 0 & + & + & + \\ 0 & + & + & + \end{pmatrix}.$$

Second we find a reflection H_2 that zeroes out the element $A(4,2)$. We get

$$H_2 \begin{pmatrix} x & x & x & x \\ x & x & x & x \\ 0 & x & x & x \\ 0 & x & x & x \end{pmatrix} H_2^T = \begin{pmatrix} x & x & x & x \\ x & x & x & x \\ 0 & + & + & + \\ 0 & 0 & + & + \end{pmatrix} H_2^T = \begin{pmatrix} x & x & + & + \\ x & x & + & + \\ 0 & x & + & + \\ 0 & 0 & + & + \end{pmatrix},$$

which is Hessenberg.

5: For **a)** the function $f(x)$ is obtained by putting the equations in a vector, i.e.

$$f(x) = \begin{pmatrix} (x_1-1)^2 + 3x_2 - 3 \\ \cos(x_1) + (x_3-1)^2 - 1 \\ x_1 + x_2^2 + (x_3+1)^2 - 2 \end{pmatrix}.$$

The equation to solve is then $f(x) = 0$, where $x \in \mathbb{R}^3$. The Jacobian is obtained by computing derivatives. We see that

$$J_f(x) = (\partial f_i / \partial x_j) = \begin{pmatrix} 2(x_1-1) & 3 & 0 \\ -\sin(x_1) & 0 & 2(x_3-1) \\ 1 & 2x_2 & 2(x_3+1) \end{pmatrix}.$$

Given a starting guess $x^{(0)}$ the Newton iteration can be written

$$x^{(k+1)} = x^{(k)} + J_f(x^{(k)})^{-1} f(x^{(k)}).$$

For **b)** We note that the requirement on uv^T is satisfied if

$$(B_k + uv^T)s^{(k)} = f(x^{(k+1)}) - f(x^{(k)}) = y^{(k)}.$$

This is equivalent to

$$(v^T s^{(k)})u = f(x^{(k+1)}) - f(x^{(k)}) = y^{(k)} - B_k s^{(k)} = z^{(k)}$$

Thus u and $z^{(k)}$ has to be paralell. We can pick $u = z^{(k)}/\|s^{(k)}\|_2^2$ and then chose v so that $\|uv^T\|_2$ is minimized, while the restriction $v^T s^{(k)} = \|s^{(k)}\|_2^2$ holds. This leads to the choice $v = s^{(k)}$.

6: For **a)** we let $p = Ar - \alpha r$ and chose α so that p is A-orthogonal to r. This means that

$$0 = (p, r)_A = r^T A^T (Ar - \alpha r) = r^T A^T A r - \alpha r^T A r \Longrightarrow \alpha = \frac{\|Ar\|_2^2}{r^T A r}.$$

We also note that α is always well defined unless $r = 0$ but in that case we already have the exact solution to the linear system $Ax = b$.

For **b)** the algorithm for computing the next iterate $x^{(k+1)}$ from the current $x^{(k)}$ is as follows. The next iterate will be of the form $x^{(k+1)} = x^{(k)} + \beta_1 r_k + \beta_2 p_k$. We get

$$r_{k+1} = b - Ax^{(k+1)} = r_k - \beta_1 A r_k - \beta_2 A p_k.$$

We need to select β_1 and β_2 so that r_{k+1} is orthogonal (not A-orthogonal) to both r_k and p_k. We obtain,

$$0 = r_k^T (r_k - \beta_1 A r_k - \beta_2 A p_k) = \|r_k\|_2^2 - \beta_1 r_k^T A r_k - \beta_2 r_k^T A p_k = \|r_k\|_2^2 - \beta_1 r_k^T A r_k,$$

since r_k and p_k are A-orthogonal, or

$$\beta_1 = \frac{\|r_k\|_2^2}{r_k^T A r_k}.$$

The constant β_2 is computed by

$$0 = p_k^T(r_k - \beta_1 Ar_k - \beta_2 Ap_k) = p_k^T r_k - \beta_1 p_k^T Ar_k - \beta_2 p_k^T Ap_k = p_k^T r_k - \beta_2 p_k^T Ap_k,$$

or

$$\beta_2 = \frac{p_k^T r_k}{p_k^T Ap_k}.$$

Now we have everything needed to compute $x^{(k+1)}$.

The algorithm can be written in several ways. The only important things is to introduce intermediate results $z_k = Ar_k$ and $w_k = Ap_k$ since both these factors appear multiple times in the formulas. After $x^{(k+1)}$ is computed we avoid a multiplication by A by updating the residual using the formula

$$r_{k+1} = b - Ax^{(k+1)} = r_k - \beta_1 z_k - \beta_2 w_k.$$

This means that the algorithm requires two multiplications by A in each step.