September 13, 2017

Matlab Assignments – Lecture 3, Fall 2016

In these assignments, we investigate the stability of several variants of the Gram-Schmidt orthonormalisation process. These assignments extend the ones of Lecture 1.

Assignment 3.1. Let $\mathbf{a}_1, \dots, \mathbf{a}_k$ be n-vectors. We use the Gram–Schmidt process to construct an orthonormal sequence $\mathbf{q}_1, \dots, \mathbf{q}_k$ such that

$$\text{span}(\mathbf{A}_j) = \text{span}(\mathbf{Q}_j) \quad \text{for all } j = 1, 2, \dots, k. \tag{3.1}$$

Here, $\mathbf{A}_j$ is the $n \times j$ matrix with columns $\mathbf{a}_1, \dots, \mathbf{a}_j$. Similarly $\mathbf{Q}_j \equiv [\,\mathbf{q}_1, \dots, \mathbf{q}_j\,]$. With span($\mathbf{A}_j$) we refer to the space spanned by the columns of the matrix $\mathbf{A}_j$.

There are several variants of Gram–Schmidt that are mathematically equivalent. In this assignment we try to get some insight in their stability properties. All variants follow an recursive construction.

Gram–Schmidt

```
classical
for $j = 1, \dots, k$ do
    $\mathbf{v} = \mathbf{a}_j$
    $r_j = \mathbf{Q}_{j-1}^*\mathbf{v}$
    $\mathbf{v} \leftarrow \mathbf{v} - \mathbf{Q}_{j-1}\, r_j$
    $r_{jj} = \|\mathbf{v}\|_2$
    $\mathbf{q}_j = \mathbf{v}/r_{jj}$
```

```
repeated
for $j = 1, \dots, k$ do
    $\mathbf{v} = \mathbf{a}_j$ , $r_j = 0$
    repeat twice
        $h = \mathbf{Q}_{j-1}^*\mathbf{v}$
        $\mathbf{v} \leftarrow \mathbf{v} - \mathbf{Q}_{j-1}\, h$
        $r_j \leftarrow r_j + h$
    $r_{jj} = \|\mathbf{v}\|_2$
    $\mathbf{q}_j = \mathbf{v}/r_{jj}$
```

```
modified
for $j = 1, \dots, k$ do
    $\mathbf{v} = \mathbf{a}_j$
    for $i = 1, \dots, j-1$ do
        $r_{ij} = \mathbf{q}_i^*\mathbf{v}$
        $\mathbf{v} \leftarrow \mathbf{v} - \mathbf{q}_i r_{ij}$
    $r_{jj} = \|\mathbf{v}\|_2$
    $\mathbf{q}_j = \mathbf{v}/r_{jj}$
```

Here, we take the conventions that $\mathbf{Q}_0 = [\,]$ (in MATLAB, take for $\mathbf{Q}_0$ an $n \times 0$ matrix: `Q=zeros(n,0);`), $\mathbf{Q}_0^*\mathbf{a}_1 = [\,]$, etc., the loop $i = 1 : 0$ is empty and is to be skipped. r_j is a $(j-1)$-vector with ith entry equal to r_{ij} (of modified Gram–Schmidt).

Assemble the vectors r_j and scalars r_{jj} in a matrix R_k: the ij-entry of R_k is
the ith coordinate r_{ij} of r_j if $i < j$,
0 if $i > j$, and
r_{jj} if $i = j$:
r_j is the jth column of R_k appended with r_{jj} and zeros.

(a) Show that R_k is $k \times k$ upper triangular and

$$\mathbf{A}_k = \mathbf{Q}_k\, R_k$$

for the three variants: the variants are mathematically equivalent.

The equivalence is demonstrated under the assumption that no rounding errors are involved.

(b) Write MATLAB function subroutines for each of these variants
(input $\mathbf{A}$, output $\mathbf{Q}$ and R; `function [Q,R]=clasGS(A);`
`function [Q,R]=repeatedGS(A);`
`function [Q,R]=modGS(A);`).[1]

(c) To assess the orthogonality (or loss of orthogonality) plot the $\log_{10}$ of the

$$\nu_j \equiv \|I_j - \mathbf{Q}_j^*\mathbf{Q}_j\|_2$$

as function of j. Here I_j is the $j \times j$ identity matrix. Note that in exact arithmetic $\nu_j = 0$.

(d) Compare the results for, for instance, the matrices (larger n, $n = 200$?) from Assignment 1.2. Try also the square $n \times n$ matrix $\mathbf{A}$ ($n = 200$) obtained by replacing all diagonal entries of a random $n \times n$ matrix by α (say, $\alpha = 0.1$). Relate the loss of orthogonality to the condition number $\mathcal{C}_2(\mathbf{A}_j)$ (`cond(A,2);`) of $\mathbf{A}_j$.

(e) In a situation where repeating the orthogonalisation is helpful, investigate (experimentally) the effect of increasing the number of repetitions in repeated Gram–Schmidt.

(f) For repeated Gram–Schmidt, the following repetition criterion is popular. If the tangents between $\mathbf{v}_j = \mathbf{v}$ and $\text{span}(\mathbf{Q}_{j-1})$ is small, smaller than κ (say less than $\kappa = 0.5$) then repeat. The tangents can be computed as $\|h\|_2/\|\mathbf{v}\|_2$. Why? Include this repetition criterion (**DGKS** [Daniel-Gragg-Kaufman-Stewart] **criterion**) in your code and investigate its effect (on the orthogonality, and on the computational costs).

Note. A robust MATLAB routine should check whether divison by r_{jj} is allowed, i.e., whether $\|\mathbf{v}_j\|_2 \neq 0$: do not expand $\mathbf{Q}_{j-1}$ if $\|\mathbf{v}_j\|_2 = 0$. Unfortunately, even if in exact arithmetic $\|\mathbf{v}_j\|_2 = 0$, in rounded arithmetic the computed value will be non-zero but can be as big as $\mathbf{u}\, n\, \sqrt{k}\, \|\mathbf{a}_j\|_2$. Here, $\mathbf{u}$ is the relative machine precision (`0.5*eps` in MATLAB). Therefore, since $\|\mathbf{a}_j\|_2^2 = \|r_j\|_2^2 + \|\mathbf{v}_j\|_2^2$, do not expand if $\|\mathbf{v}_j\|_2 < \mathbf{u}\, n\, \sqrt{k}\, \|r_j\|_2$. Here, we neglected terms of order $\mathcal{O}(\mathbf{u}^2)$ and we replaced $\|\mathbf{a}_j\|_2$ by $\|r_j\|_2$ since the computation of the norm of $\mathbf{a}_j$ would require an additional inner product with an n-vector, while the computation of $\|r_j\|_2$ involves a j-vector only.

Assignment 3.2. Now, suppose $\mathbf{A}$ is a square $n \times n$ matrix. The **Arnoldi algorithm** is obtained from the Gram–Schmidt process by replacing the first line $\mathbf{v}_j = \mathbf{a}_j$ by $\mathbf{v}_j = \mathbf{A}\mathbf{v}_{j-1}$ except for $j = 1$, where $\mathbf{v}_1 = \mathbf{a}_1$. This leads to the **Arnoldi factorisation**

$$\mathbf{A}\mathbf{Q}_k = \mathbf{Q}_{k+1}\underline{H}_k,$$

where $\underline{H}_k$ is the $(k+1) \times k$ right block of R_{k+1}, the upper-triangular matrix as produced by Gram–Schmidt after $k + 1$ steps: $\underline{H}_k$ is upper Hessenberg.

[1] Some hints for MATLAB programming: MATLAB skips empty loops;
if `A` is an $n \times k$ matrix ($n \times k$ array), then
```
v=A(:,j);
```
sets `v` equal to the jth column of `A`, while
```
A(:,j)=w;
```
sets the jth column of `A` equal to the vector `w`, with an error message of the sizes do not match;
```
v=v-Q*r;
```
replaces the 'old' vector `v` by its updated version;
```
r=Q'*v;
```
computes $\mathbf{Q}^*\mathbf{v}$: MATLAB's prime takes the complex conjugate transpose;
if `Q` is an $n \times (j-1)$ array, then
```
Q=[Q,v/r(j,j)];
```
expands the array `Q` with the column vector `v/r(j,j)` to an $n \times j$ array.

(a) Write a MATLAB function subroutines to perform the Arnoldi process for the Gram–Schmidt variants and investigate the stability behaviour ((loss of) orthogonality). As an example take the test matrix `'lehmer'` (as in Assignment 1.2).