MCS 471 Project Three: Gram-Schmidt Orthogonalization and QR

The goal of the project is to study the use of orthogonal matrices to solve linear systems.

In our experiments we will use MATLAB or Octave. MATLAB is available in all computer labs on campus, but is commercial and expensive. Octave is similar to MATLAB (for our work, Octave runs just as fine), and is free to download from `http://www.octave.org/`.

0. Solving Linear Systems in MATLAB or Octave

The general command to solve linear systems is done by the "backslash" operator: for a matrix `A` and right-hand-side vector `b`, we find the solution `x` as `x = A\b`. The residual vector is then `r = b - A*x`.

The LU decomposition is provided in the command **lu**. For a matrix `A`, `[l,u,p] = lu(A)` returns in `l`, `u`, and `p` the lower, upper, and permutation matrix used in the LU factorization with partial pivoting. To solve `A x = b`, the `[l,u,p] = lu(A)` is followed by the sequence `pb = p*b; y = l\pb; x = u\y;`.

The command **qr** returns the QR decomposition of a matrix. For a matrix `A`, `[q,r] = qr(A)` returns in `q` an orthogonal matrix (i.e.: `q'*q = eye(size(q))`), and in `r` an upper triangular matrix. Notice that `'` is the transpose operator. To solve `A x = b`, the `[q,r] = qr(A)` is followed by `qb = q'*b;` and `x = r\qb;`. The solution `x` minimizes the 2-norm of the residual vector, it is the least squares solution.

The command **cond** provides an estimate for the condition number of a matrix.

We can group commands into .m files, defining functions. The name of the function should match the file name. For this project, we need the function "gsqr" with its definition in the file "gsqr.m" (download "gsqr.m" from the class web site).

If the path is set correctly (do **help path** otherwise to see how to set the search path), then you can call this function just as a regular command.

1. Gram-Schmidt Orthogonalization

Given a real n-by-k matrix A (with $n \leq k$), denote the k columns of A as $[\mathbf{a}_1 \ \mathbf{a}_2 \ \cdots \ \mathbf{a}_k]$. To find an ortogonal n-by-k matrix $Q = [\mathbf{q}_1 \ \mathbf{q}_2 \ \cdots \ \mathbf{q}_k]$ with the same column span as A, we can apply the formulas

$$\mathbf{p}_j = \mathbf{a}_j - \sum_{i=1}^{j-1} \mathbf{q}_i(\mathbf{q}_i^T \mathbf{a}_j), \quad \mathbf{q}_j = \frac{\mathbf{p}_j}{||\mathbf{p}_j||_2}, \quad \text{for } j = 1, 2, \ldots, k. \tag{1}$$

The formulas (1) are known as the Gram-Schmidt orthogonalization method. We observe that $\mathbf{p}_j$ is obtained from $\mathbf{a}_j$, subtracting the orthogonal projections of $\mathbf{a}_j$ on the column space of $[\mathbf{a}_1 \ \mathbf{a}_2 \ \cdots \ \mathbf{a}_{j-1}]$. Collecting the coefficients $\mathbf{q}_i^T \mathbf{a}_j$ into r_{ij} leads to an upper triangular matrix R such that $A = QR$, the QR decomposition of A. To compute this QR decomposition, use `[q,r] = gsqr(A)`.

1.1 Cost of the Gram-Schmidt Orthogonalization.

The Gram-Schmidt orthogonalization method needs $O(n^3)$ operations, just like the LU decomposition. In the following assignment we will compute precise expressions for the cost of this method.

Assignment One. For this assignment, we consider square matrices, i.e.: $k = n$. Count the number of operations for $n = 2, 3, 4$ and 5, distinguishing between additions (subtractions) and multiplications (divisions). Use interpolation to find the cubic polynomials in n which respectively return the number of additions (subtractions) and multiplications (divisions).

1.2 Loss of Orthogonality

The Gram-Schmidt orthogonalization method as given by formulas (1) is numerically instable, as it leads to a loss of orthogonality. The accumulation of errors on larger matrices may lead to a relatively large inner product of the last two vectors, even on random matrices.

The modified Gram-Schmidt orthogonalization uses the formulas

$$\mathbf{p}_j = \mathbf{a}_j - \sum_{i=1}^{j-1} \mathbf{q}_i(\mathbf{q}_i^T \mathbf{p}_j), \quad \mathbf{q}_j = \frac{\mathbf{p}_j}{||\mathbf{p}_j||_2}, \quad \text{for } j = 1, 2, \ldots, k. \tag{2}$$

Assignment Two. Use the formulas (2) to modify the script `gsqr.m` into `mgsqr.m`, implementing the modified Gram-Schmidt orthogonalization method.

Generate random matrices of size 20, 40, 60, 80, and 100. Apply `gsqr` and `mgsqr` on each matrix and compute the inner product of the two last vectors in the orthogonal matrix Q, as returned by `gsqr` and `mgsqr`. List these inner products in a table, using scientific format for the numbers.

What do you observe about the growth of the numbers as the dimensions of the random matrices increase?

2. The Normal Equations ant the QR Decomposition

To solve an overdetermined system $Ax = b$, for A an n-by-k matrix, we could solve the normal equations $A^T Ax = A^T b$. While theoretically, the solution to the normal equations is the least squares solution, solving the normal equations is numerically not as stable as using the QR decomposition of the matrix A. The goal of the following assignment is to confirm the numerical instability of the normal equations approach experimentally.

Assignment Three. For $k = 10, 20, 30, 40, 50, 60, 70, 80, 90$ and $n = 100$, generate a random matrix n-by-k matrix A. Let the exact solution x be a vector of ones, i.e.: `x = ones(k,1)` and then `b = A*x`.

The \ operator will automatically return the least squares solution for an overdetermined linear system. Solve the 9 linear systems once with `A\b` and once with the normal equations approach, also using \ after computing `A'*A` and `A'*b`. Compare the norm of the residual vectors for the two methods on the 9 linear systems.

For which k is the difference between the norms of the residual vectors largest? Try to explain why this is so.

3. Ill-conditioned Linear Systems

The (i, j)-th element a_{ij} of the n-dimensional Hilbert matrix A is defined as $\frac{1}{i+j-1}$, for i and j ranging between 1 and n. The condition number of this matrix A worsens as n grows.

Assignment Four. For $n = 2, 3, \ldots$, generate the n-dimensional Hilbert matrix `A` and set up the linear system with `x = ones(n,1)` as the exact solution and corresponding right hand side vector `b = A*x`.

Use `y = A\b` to solve the linear system for increasing values of n. How far can you go with n and still have two decimal places correct in the answer `y`? Apply the formula

$$\frac{||x - \bar{x}||}{||x||} \leq \text{cond}(A)\frac{||A - \bar{A}||}{||A||} \tag{3}$$

to predict the error using the value of the condition number, estimated by the **cond** command. The machine precision is returned by the command **eps**.

Make a table with four columns. The first column contains the values for $n = 2, 3, \ldots$, followed by `cond(A)` in the second column. The third column contains the predicted error, obtained by application of (3). The observed error `norm(x-y,2)` is listed in the last column. As always, use scientific format to display the floating-point numbers.

Describe how well the predicted errors match the observed errors.

4. The deadline is Monday 31 October 2005 at 3PM

Bring *your* solution to the project to class. The *your* is emphasized to stress that your solution is the result of an *individual* effort. Collaborations are **not** permitted.

Your solution should contain the following:

1. Answers to the questions in the assignments. Please write complete grammatically correct sentences.
2. Tables summarizing the numerical experiments you have done. Use the **format short e** command when generating data, so your numbers in the tables will also be in scientific notation.
3. Sequences of commands used to generate the data for the experiments. It is good practice to store these commands in little .m files, e.g., in `assignment_one.m`, `assignment_two.m`, etc.
4. Output of your sessions with MATLAB or Octave as *an appendix*. Typing **diary** followed by the name of a file in a session creates a new file with the given name which will contain everything you see on the screen during the session. You may edit out mistakes from the output of diary.

The solution to the project is essentially a report on paper.

If you have questions or difficulties with the assignments, feel free to come to my office for help.