550.291 Linear Algebra and Differential Equations
Spring 2008
Project 2: Orthogonal Matrices
(Extra Credit)

Open a workspace in MATLAB by selecting the program from the Start Menu. When you are finished with this project, print the commands and results shown in the command window by selecting File/print. Note: The output can be quite long is you have been working in MATLAB for some time. You can save the command window display into a text file, edit it, and print it later.

Part 1: Gram-Schmidt Orthogonalization.

(a) We are going to create a new M-file. In the MATLAB command window, type `edit`.

(b) Type the following commands into the MATLAB editor window:

```
function [W, U] = gram_schmidt(V)
% Input matrix V; Output matrices W and U
% The columns of matrix V form a basis for vector space V
% The columns of matrix W are an orthogonal basis for vector space V
% The columns of matrix U are an orthonormal basis for vector space V

% initialization of variables
m = size(V,1);
n = size(V,2);
W = zeros(m,n);
U = W;

for j = 1:n
   W(:,j) = V(:,j);
   if j > 1
       for k = 1:j-1
          pk = ((V(:,j)'*W(:,k))/norm(W(:,k))^2)*W(:,k);
          W(:,j) = W(:,j) - pk;
       end % for k = 1:j-1
   end % if j > 1
   U(:,j) = W(:,j)/norm(W(:,j),2);
end % for j = 1:n

% end function gram_schmidt
```

(c) Save your work.

(d) Provide a line-by-line interpretation of the code *gram_schmidt.m*. (This should be done on a separate piece of paper.)

(e) Return to the MATLAB command window. We would like to test the code using the following vectors: $\mathbf{v}_1 = (1, 3, -1, 2)$, $\mathbf{v}_2 = (0, -4, 5, 1)$, $\mathbf{v}_3 = (-7, 2, 1, 0)$. Create the matrix $\mathbf{V}$ using these vectors.

(f) Use *gram_schmidt.m* to find an orthonormal basis constructed from $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ by typing `[W, U] = gram_schmidt(V)`.

(g) We would like to modify our code so that we can enter any matrix $\mathbf{V}$ but return a basis. To do this, we must make sure that the columns of $\mathbf{V}$ form a basis.

Modify your code as follows

- Replace the line "The columns of matrix V form a basis for vector space V" with "The columns of matrix V span vector space V"
- Before the initialization of variables, insert the following

```
T = V;
[R, b] = rref(T);
V = T(:, b)
if size(T,2) ~= size(V,2)
   fprintf(1,'\n The given vectors are not linearly independent.\n');
   V
end
```

(h) Save your work.

(i) Provide a line-by-line interpretation of the modifications of the code. (This should be done on a separate piece of paper.)

(j) Return to the MATLAB command window. Test the code using the following sets of vectors: $\{(1, 0, 0,), (1, 1, 0,), (1, 1, 1), (1, 2, 3)\}$ and $\{(1, 1, 1), (1, 2, 3,), (8, 4, 8), (6, 1, 4)\}$. Explain why the code gives the results that it does.

Part 2: Symmetric, Skew-Symmetric, and Orthogonal Matrices.

If $\mathbf{A} = \mathbf{A}^T$, we say that $\mathbf{A}$ is *symmetric*. If $\mathbf{A} = -\mathbf{A}^T$, we say that $\mathbf{A}$ is *skew-symmetric*. If $\mathbf{A}^{-1} = \mathbf{A}^T$ we say that $\mathbf{A}$ is *orthogonal*. For example, the matrices

$$\mathbf{R} = \begin{bmatrix} 1 & 3 & -2 \\ 3 & 0 & 5 \\ -2 & 5 & 4 \end{bmatrix} \quad \text{and} \quad \mathbf{S} = \begin{bmatrix} 0 & 2 & 1 \\ -2 & 0 & -4 \\ -1 & 4 & 0 \end{bmatrix} \quad \mathbf{T} = \begin{bmatrix} 2/3 & 1/3 & 2/3 \\ -2/3 & 2/3 & 1/3 \\ 1/3 & 2/3 & -2/3 \end{bmatrix}$$

are symmetric, skew-symmetric, and orthogonal respectively.

(a) Using MATLAB, show that $\mathbf{R} - \mathbf{R}^T = 0$, $\mathbf{S} - (-\mathbf{S}^T) = 0$, and $\mathbf{T}^{-1} - \mathbf{T}^T = 0$.

(b) The following code tests whether a given matrix is symmetric, skew-symmetric, and/or orthogonal.

```
function transfun(A)


m = size(A,1);
n = size(A,2);


if n~=m
   fprintf(1,'\nError!  The matrix is not square!\n');
else
   if A == A'
      fprintf(1,'\nThe matrix is symmetric!\n');
   else
      fprintf(1,'\nThe matrix is not symmetric!\n');
   end
   if A == -A'
      fprintf(1,'\nThe matrix is skew-symmetric!\n');
   else
      fprintf(1,'\nThe matrix is not skew-symmetric!\n');
   end
   if inv(A) == A'
      fprintf(1,'\nThe matrix is orthogonal\n');
   else
      fprintf(1,'\nThe matrix is not orthogonal!\n');
   end
end
```

Test the code on the following matrices:

$$\mathbf{A} = \begin{bmatrix} 3 & 1 & 5 \\ 1 & 0 & -7 \\ 5 & -7 & 9 \end{bmatrix} \quad \text{and} \quad \mathbf{B} = \begin{bmatrix} 0 & 9 & -12 \\ -9 & 0 & 20 \\ 12 & -20 & 0 \end{bmatrix} \quad \mathbf{C} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Even if a square matrix is neither symmetric nor skew-symmetric, it can be decomposed as the sum of a symmetric matrix and a skew-symmetric matrix. Specifically $\mathbf{A} = \mathbf{A}_1 + \mathbf{A}_2$ where $\mathbf{A}_1 = 0.5(\mathbf{A} + \mathbf{A}^T)$ and $\mathbf{A}_2 = 0.5(\mathbf{A} - \mathbf{A}^T)$.

(c) Use the MATLAB program editor to create an m-file to decompose any square matrix as the sum of a symmetric matrix and a skew-symmetric matrix. The code should print $\mathbf{A}_1$ and $\mathbf{A}_2$ or return an error message if $\mathbf{A}$ is not square. Include your code when you turn in the project.

(d) Test your code on the following matrices:

$$U = \begin{bmatrix} 3 & 12 & 0 \\ 1 & -5 & 1 \\ 9 & 1 & 8 \end{bmatrix} \quad \text{and} \quad V = \begin{bmatrix} 1 & 0 & 1 \\ 2 & -1 & 0 \\ 4 & 0 & 6 \end{bmatrix}$$