Problem 1 (10 points)

Consider the following matrix

\[
\begin{bmatrix}
A & uu^T \\
uu^T & -A^T
\end{bmatrix}
\]

\[A \in \mathbb{R}^{n \times n}, u \in \mathbb{R}^n, v \in \mathbb{R}^n.\]

How would you efficiently compute \(B = C^2\)? Answer the question by completing the following MATLAB function:

```matlab
function B = HamSqr(A,u,v)
    % A is an n-by-n matrix and u and v are column n-vectors.
    % B is the square of the matrix C = [ A  u*u' ; v*v'  -A']
    y = A*u; z = A'*v;
    C11 = A*A + (u'*v)*u*v';
    C12 = (A*u') - u*(A')
    C21 = v(A') - (A'*v)*v;
    C22 = v*v' + A2';
    C = [C11 y*u'-u*y'; v*z'-z*v' C11']
end
```

Solution

First, work out the product...

\[
C = \begin{bmatrix}
C_{11} & C_{12} \\
C_{21} & C_{22}
\end{bmatrix} = \begin{bmatrix}
A & uu^T \\
uu^T & -A^T
\end{bmatrix} \begin{bmatrix}
A & uu^T \\
uu^T & -A^T
\end{bmatrix} = \begin{bmatrix}
A^2 + uu^T & Au^T - uu^T A^T \\
Au^T - uu^T A^T & uu^T + A^2
\end{bmatrix}
\]

Then make some observations and put parentheses in the right places....

\[
C_{11} = A^2 + (u^T v)uv^T = C_{22}^T
\]

\[
C_{12} = (Au)u^T - u*(Au)^T
\]

\[
C_{21} = v(A^T v)^T - (A^Tv)v^T
\]

Note that \((A^Tv)v^T\) is \(O(n^2)\) while \(A*(vv^T)\) is \(O(n^3)\). So...

5 points for the diagonal blocks, 5 points for the corner blocks. -4 points for correct but very inefficient, e.g., things like

\[
T1 = u*u'; T2 = v*v'; C11 = A*A + T1*T2; C12 = A*T1 - T1*A'
\]
Problem 2 (10 points)

Consider the following linear system
\[
\begin{bmatrix}
A^T & B \\
0 & A
\end{bmatrix}
\begin{bmatrix}
y \\
z
\end{bmatrix} =
\begin{bmatrix}
c \\
d
\end{bmatrix}
\]  
\(A \in \mathbb{R}^{n \times n}, y, z, c, d \in \mathbb{R}^n.\)

(a) How would you efficiently solve for \(y\) and \(z\) using Gaussian elimination with partial pivoting? Answer the question by completing the following MATLAB function:

```matlab
function [y,z] = Solve(P,L,U,c,d)
% P is an n-by-n permutation matrix
% L is an n-by-n unit lower triangular matrix
% U is an n-by-n nonsingular upper triangular matrix
% B is an n-by-n matrix
% c and d are column n-vectors
% y and z are column n-vectors with the property that
% \([A', B ; 0 A]*[ y;z] = [c;d]\)
% where P*A = L*U.
You may use the "$\backslash$" operator to solve triangular systems.

Solution

We have two equations, \(Az = d\), and \(A^T y + Bz = c\). So solve for \(z\) using \(PA = LU\), i.e., \(Lv = Pd, Uz = v\). Then solve for \(y\) via \(A^T y = c - Bz\). To do this, note that \(A^T y = A^T P^T Py = U^T L^T w = c - Bz\) where \(w = Py\). We obtain

\[
\begin{align*}
z &= U \backslash (L \backslash (P*d)); \\
y &= P^\prime * (L^\prime \backslash (U^\prime \backslash (c - B*z))); \\
\end{align*}
\]

Note that parentheses are important or an extra \(O(n^3)\) computation can arise.

Basically 4 points for \(y\) and 4 points for \(z\). Things like -1 and -2 for getting \(L\) and \(U\) mixed up or forgetting a transpose.

(b) What can you say about the accuracy of the computed \(z\) ?

Solution

Since the system for \(z\) is standard we can invoke the standard result for computed solutions that are obtained using Gaussian Elimination with pivoting...

\[
\frac{\| \hat{z} - z \|}{\| z \|} \approx u_k(A)
\]

Two points for this part. 1 for unit roundoff and 1 for condition. Can sometimes get a point for other relevant comments.
Problem 3 (10 points)

Suppose $A \in \mathbb{R}^{m \times n}$ has SVD $U^TAV = \Sigma$ where $U \in \mathbb{R}^{m \times m}$ and $V \in \mathbb{R}^{n \times n}$ are orthogonal and $\Sigma = \text{diag}(\sigma_1, \ldots, \sigma_n)$ has the property that
\[
\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > \sigma_{r+1} = \cdots = \sigma_n = 0
\]
Assume that $r < n$ so that $A$ is rank deficient.

(a) Using the SVD, specify a basis for the null space of $A$.

Solution

Comparing columns in $U^TAV = \Sigma$ we conclude that
\[
U^TAV(:,i) = \Sigma(:,i) = 0, \quad i = r + 1:n.
\]
Thus, columns $r + 1$ through $n$ of the $V$ matrix span the null space of $A$.

Three points for this. 1 point for "playing around" with $V$.

(b) Using the SVD, specify a minimizer of $\|Ax - b\|_2$ that is orthogonal to the null space of $A$.

Solution

\[
\|Ax - b\|_2^2 = \|USV^Tx - c\|_2^2 = \|\Sigma y - c\|_2^2 = \sum_{i=1}^r (\sigma_i y_i - c_i)^2 + \sum_{i=r+1}^n c_i^2
\]
where $c = U^Tb$ and $y = V^Tx$. It follows that the best way to choose $y$ is
\[
y_i = \begin{cases} 
    c_i/\sigma_i & i = 1:r \\
    \text{anything} & i = r + 1:n
\end{cases}
\]
Since
\[
x = Vy = y_1V(:,1) + \cdots + y_nV(:,n)
\]
we simply set $y_i = 0, i = r + 1:n$ and this gives a minimizer that is orthogonal to
\[
\text{span}\{V(:,r+1),\ldots,V(:,n)\} = \text{null}(A).
\]
4 points for this.
(c) Using the SVD, how would you compute a minimizer of \( \|Ax - b\|_2 \) that is closest (in the 2-norm sense) to a given vector \( z \in \mathbb{R}^n \)? Specify your answer by completing the following function

```matlab
function x = SpecialLS(U,Sigma,V,r,b,z)
% U (m-by-m) and V (n-by-n) are orthogonal
% Sigma is an m-by-n diagonal matrix with
% Sigma(1,1) >= ... >= Sigma(r,r) > Sigma(r+1,r+1)=...=Sigma(n,n) = 0
% b is a column m-vector and z is a column n-vector.
% Assume that r is an integer that satisfies 1 <= r < n.
% x is the minimizer of norm(A*x - b,2) that minimizes norm(x - z,2)
% where A = U*Sigma*V'.

Solution

Let’s go back to part(b) and think about how we might choose \( y_{r+1}, \ldots, y_n \) so as to get as close as possible to \( z \). Since

\[
\| x - z \| = \| V^T x - V^T z \| = \| y - w \|
\]

we see that we should set \( y_i = w_i = V(:,i)^T z \) for \( i = r + 1 : n \).

```matlab
n = length(z);
x = zeros(n,1);
for i=1:n
    if i<=r
        % this builds up the part (b) solution
        x = x + ((U(:,i)'*b)/Sigma(i,i))*V(:,i);
    else
        x = x + (V(:,i)'*z)*V(:,i);
    end
end
```

3 points for this