

Lab Session 7

1. Singular value decomposition (SVD) of a nonsingular matrix A can be computed as follows.

1. Compute spectral decomposition $A^*A = V\text{diag}(\lambda_1, \dots, \lambda_n)V^*$ such that $\lambda_j \geq \lambda_{j+1}$.
2. Define $\Sigma := \text{diag}(\sqrt{\lambda_1}, \dots, \sqrt{\lambda_n})$.
3. Compute $U := AV\Sigma^{-1}$. Then $A = U\Sigma V^*$ is an SVD of A .

Write a MATLAB function, say, `[U, S, V] = mysvd1(A)` that implements the above method. Use built in MATLAB functions such as `eig` and `svd` in your function and write comments about input and output. Your function may look like this

```
function [U, S, V] = mysvd1(A)
% [U, S, V] = mysvd1(A) produces a diagonal matrix S of singular
% values of nonsingular matrix A and two unitary matrices U and V
% whose columns are the corresponding left and right singular
% vectors so that A * V = U * S.
```

2. Singular value decomposition (SVD) of an m -by- n matrix A can be computed as follows.

1. Compute spectral decomposition $A^*A = V\text{diag}(\sigma_1^2, \dots, \sigma_r^2, 0, \dots, 0)V^*$, where $r = \text{rank}(A)$ and $\sigma_j \geq \sigma_{j+1}$.
2. Set $V_1 := V(:, 1:r)$, the first r columns of V , and set $\Sigma_r := \text{diag}(\sigma_1, \dots, \sigma_r)$.
3. Compute spectral decomposition $AA^* = Z\text{diag}(\sigma_1^2, \dots, \sigma_r^2, 0, \dots, 0)Z^*$ and set $U_2 := Z(:, r+1:m)$, the last $m-r$ columns of Z .
4. Compute $U_1 := AV_1\Sigma_r^{-1}$.
5. Set $U := [U_1, U_2]$ and $\Sigma := \begin{bmatrix} \Sigma_r & 0 \\ 0 & 0 \end{bmatrix} \in \mathbb{C}^{m \times n}$. Then $A = U\Sigma V^*$ is an SVD of A .

Write a MATLAB function `[U, S, V] = mysvd2(A)` that implements the above method.

3. **Experiment:** Your task is to compare `mysvd1` and `mysvd2` with the MATLAB function `svd` by applying these functions on some test matrices. For this purpose, compute $\|V^*V - I\|_2$, $\|U^*U - I\|_2$ and $\|AV - US\|_2$ when U , S and V are obtained by your functions as well as by `svd(A)`. Plot these results and conclude which method is better and reliable. Plot the singular values and compare the smallest singular value obtained by all three methods. Consider the following test matrices.

1. Consider the magic squares `magic(n)` for $n = 4, 5, 6$.
2. Consider the Hilbert matrix `hilb(n)` for $n = 4, 5, \dots, 12$.
3. The frank matrix `F = gallery('frank', n)` for $n = 8, 10, 12$.
4. Random matrix with normalized columns and specified singular values generated by MATLAB command `gallery('randcolu', x)`. For more information type `help private/randcolu`. If σ_j 's are specified singular values and $\hat{\sigma}_j$'s are computed singular values then compute the relative errors $|\sigma_j - \hat{\sigma}_j|/\sigma_j$ for all three methods. Which method is better?

*** End ***