Project 1

Jacky Lin

October 7, 2025

1 SVD BlackBox Comparison

```
Matrix for SVD test:
 [[1 2 3]
 [0 1 4]
 [5 6 0]]
My SVD
U:
 [[-0.33780332  0.51318841  0.78900353]
 [-0.19885965  0.78044262 -0.59275978]
 [-0.91996943 -0.35713718 -0.16158364]]
Sigma:
 [[8.27883923 0.         0.        ]
 [0.         4.84357297 0.        ]
 [0.         0.         0.02493818]]
VT:
 [[-0.59641821 -0.77236466 -0.2184906 ]
 [-0.26271876 -0.06937103  0.96237545]
 [-0.75846171  0.63137983 -0.16154054]]
Condition number: 331.9745145824363
Inverse:
 [[-24.  18.   5.]
 [ 20. -15.  -4.]
 [ -5.   4.   1.]]
Reconstruction
 [[1.00000000e+00 2.00000000e+00 3.00000000e+00]
 [4.64878775e-16 1.00000000e+00 4.00000000e+00]
 [5.00000000e+00 6.00000000e+00 2.97488237e-16]]
```

Figure 1: Project SVD Routine

```
NumPy Black-box SVD
U:
 [[-0.33780332 -0.51318841 -0.78900353]
 [-0.19885965 -0.78044262  0.59275978]
 [-0.91996943  0.35713718  0.16158364]]
Sigma:
 [[8.27883923 0.         0.        ]
 [0.         4.84357297 0.        ]
 [0.         0.         0.02493818]]
VT:
 [[-0.59641821 -0.77236466 -0.2184906 ]
 [ 0.26271876  0.06937103 -0.96237545]
 [ 0.75846171 -0.63137983  0.16154054]]
Reconstruction
 [[ 1.00000000e+00  2.00000000e+00  3.00000000e+00]
 [ 3.39735283e-16  1.00000000e+00  4.00000000e+00]
 [ 5.00000000e+00  6.00000000e+00 -6.19060897e-16]]
```

Figure 2: Blackbox SVD Routine

After running the black box Python routine on an invertible matrix and comparing it with my own, I notice some key differences. The magnitudes of all of the values in the respective matrices are identical; however, the sign of each value varies depending on the function called, as seen in Figure 1 and Figure 2. Some values have the same sign while others have opposite signs. Despite this, however, they all contain the same Σ matrices and reconstruct back to the same invertible A matrix. The reason for these differences is that the SVD black box has a different ordering than my routine. This is important because it shows that the signs of singular vectors in SVD are not unique. Consequently, differences in sign between routines do not affect the decomposition's validity.

2 Two Free-End Examination

In the case of two free ends for a spring–mass system, the behavior is fundamentally different from the two-fixed-end or single-fixed-end cases. Physically, a two-free-end system cannot exist and is therefore not solvable. But we can also mathematically represent why this is unsolvable.

2.1 System Definition

In a two-free-end system, the number of springs n is one less than the number of masses m. To construct the stiffness matrix, we first define the elongation matrix A, which relates spring elongations to nodal displacements. Since $n < m$, A is a rectangular $n \times m$ matrix that contains fewer equations than unknowns. The diagonal matrix C contains the spring constants and is invertible (positive definite). The overall stiffness matrix is then.

$$K = A^T CA$$

The general form of A is:

$$A = \begin{bmatrix} 1 & -1 & 0 & \cdots & 0 \\ 0 & 1 & -1 & \cdots & 0 \\ \vdots & & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 & -1 \end{bmatrix} \tag{1}$$

2.2 Nullspace of A and Rank

Each row of A contains $(1,-1)$ corresponding to the displacement difference across a spring. The nullspace of A is then just the vector of ones, resulting in the non-trivial solution.

$$\mathbf{u} = [1,1,\ldots,1]^T$$

Because of this non-trivial solution, the resulting K matrix will be rank-deficient.

$$A\mathbf{u} = 0$$

Substituting this expression into the stiffness matrix definition gives us the following.

$$K\mathbf{u} = A^T CA\mathbf{u} = A^T C(0) = 0$$

Here we can see that the Null Space of K also contains the nontrivial solution, which ultimately means that K is singular. Consequently, K is also not invertible, proving that the two-free-end system has no solution. However, I will also show it through the rank of K. Firstly, the full rank of a rectangular matrix is whichever of n and m is smallest, and in this situation, the rank of A would be n.

$$\text{rank}(A) = m-1 = n$$

2.3 Rank of K

Since K results in an $m \times m$ matrix, its full rank should be m. However, due to the nature of the system, K is rank deficient. Firstly, C only contains diagonal terms of the stiffness coefficients, meaning that multiplying A by C does not change the linear independence of A's columns. Multiplying A by C only leads to a linear combination of the columns of A, meaning that the rank of A stays the same. Multiplying a matrix by a full-rank, invertible square matrix does not change its own rank.

$$\text{Rank}(CA) = \text{Rank}(A)$$

Multiplying by A^T also does not increase the rank, as

$$\text{rank}(A^T A) \leq \min(\text{rank}(A), \text{rank}(A^T)) = \text{rank}(A)$$

Consequently, the rank($A^T A$) must be less than or equal to n or $m-1$. I can then infer the following expression showing the rank deficiency.

$$\text{rank}(A^T CA) = \text{rank}(A^T A) \leq \text{rank}(A) < m$$

2.4 Rank-Nullity

According to many sources online, the rank of a square matrix is equal to their number of columns subtracted by the dimension of their null space. In this situation, the dimension of the null space of A is 1 ($u = [1, 1, ...,]^T$). Substituting this idea into the stiffness matrix definition gives us the following.

$$K\mathbf{u} = A^T CA\mathbf{u} = A^T C(0) = 0$$

This means that the null space of k must also be equal to the null space of A, meaning that the rank of K is also m - Null(k).

$$\text{rank}(K) = m - \text{Null}(K) = m - 1 = n$$

and thus K is rank-deficient by one, so we cannot solve this system.

2.5 Consequences

Because K is singular, the system $Ku = f$ cannot be solved. Physically, this corresponds to a rigid-body translation in which all masses move together without stretching any springs. Mathematically, the two-free-end spring–mass system has no unique solution due to the rank deficiency.

$$\text{rank}(K) = m - 1 < m$$

Consequently, K does not have an inverse and the overall system cannot be solved.

2.6 Condition of the K matrix

```
Free Free

Singular values of K:
[[3.00000000e+00 0.00000000e+00 0.00000000e+00]
 [0.00000000e+00 1.00000000e+00 0.00000000e+00]
 [0.00000000e+00 0.00000000e+00 3.54767319e-08]]
Eigen values of K:
[9.00000000e+00 1.00000000e+00 1.25859851e-15]
K:
 [[ 1. -1.  0.]
 [-1.  2. -1.]
 [ 0. -1.  1.]]
Displacements:
 [[2.22044605e-16]
 [2.22044605e-16]
 [8.88178420e-16]]
Elongations:
 [[0.00000000e+00]
 [6.66133815e-16]]
Spring stresses:
 [[0.00000000e+00]
 [6.66133815e-16]]
Condition number of K: 84562467.83072342
```

Figure 3: Free-Free Results

For this project, I also ran a case of the free-free system using my code. The results are shown above in Figure 3. As we can see, the interesting thing is that the condition number of the K matrix is extremely high, further highlighting that the K matrix is ill conditioned and the system cannot be solved. Overall, this behavior corresponds to that of a rigid-body translation, where all masses move together without deforming the springs.