Week #12: Linear Algebra III

Goals:

- Describe the properties of the matrix determinant.
- Explain the concept of an orthogonal and orthonormal basis.
- Diagonalize a matrix and use the result to interpret the behaviour of a linear system.

The Determinant

The determinant of a square matrix (having the same number of rows as columns) is a useful way to find out whether a matrix forms a basis for the space it is in. Recall that a set of column vectors of a matrix forms a basis if the rank of the matrix is equal to the dimension of the matrix, and the null set of the matrix is the zero vector, $\overrightarrow{0}$. The determinant of the matrix allows us to check whether a matrix is a basis very quickly using the following fact:

If the determinant of a matrix is not equal to zero, then that matrix forms a basis. If the matrix determinant is equal to zero, then that matrix does not form a basis and is called **singular**.

Finding the determinant by hand can require a lot of work, especially for very large matrices, but we have MATLAB to help us. To find the determinant of a matrix **A** in MATLAB we use the command:

```
det(A)
```

If MATLAB returns a zero (or something very close to it), then that matrix does not form a basis.

Practice finding the determinant of the following matrices

$$\mathbf{A} = \begin{pmatrix} 2 & 3 & 7 \\ 1 & -9 & 0 \\ 3 & -6 & 7 \end{pmatrix} \quad \mathbf{B} = \begin{pmatrix} -6 & 1.1 & 8 & 3.2 \\ -4.5 & 10 & 6.6 & 8.9 \\ 0.5 & 5.8 & -7.1 & 2.3 \\ 0.8 & -9 & 4.2 & 18 \end{pmatrix}$$

$$\mathbf{C} = \begin{pmatrix} 2.4 & 9.3 & 11.7 \\ 6.1 & -2.9 & 3.2 \\ 10.1 & 5.4 & 4.7 \end{pmatrix} \quad \mathbf{D} = \begin{pmatrix} 8.6 & 7.2 & 8.8 & -2.6 \\ 2.5 & -0.9 & 12.8 & 3.3 \\ -2.7 & -3.8 & 4.7 & -8.9 \\ 1 & 0 & 0 & -4 \end{pmatrix}$$

Try and think about how the concepts of rank, null spaces, dimension and the determinant are related.

$\det(A) = 0$, $\det(B) = 8753.8$, $\det(C) = 687.8520$, $\det(D) = 1002.2$

Hopefully, you come to the conclusion that the rank of a matrix can only equal its dimension if the determinant is not equal to zero. If a matrix has a nullspace that consists of anything other than the zero vector, it has a determinant of 0. While checking the rank and dimension will give you the same answer as checking the determinant when looking for whether a matrix forms a basis or not, it is (slightly) faster to simply check one thing, the determinant, than two things, rank and dimension.

The physical interpretation of the determinant requires some geometry and intuition beyond the scope of this course. Suffice it to say that the determinant is a quick way to check whether or not a matrix forms a basis in the space.

Orthogonal Basis

Before we get into why we should learn about the uses of an orthogonal basis, we should recall the definition of **orthogonal** first. A set of vectors is orthogonal if the inner products of all of the vectors in that set are 0. We are going to ignore the case when all of the vectors are the zero vector, and assume that the vectors in our set are nonzero.

This leads into a really nice fact: If a set of vectors are all mutually orthogonal, then all of those vectors are linearly independent. If you have enough mutually orthogonal vectors to span the space (meaning the rank of the matrix is equal to its dimension), then those vectors form a basis.

Why all the fuss about a basis being orthogonal? If we already have a basis formed by the matrix $\mathbf{A}$, we know that the equation

$$\mathbf{A}\overrightarrow{x} = \overrightarrow{b}$$

has a unique solution. Recall that this means that for any vector $\overrightarrow{b}$, we can find the unique vector $\overrightarrow{x}$ that will make the equation above true. However, finding $\overrightarrow{x}$ can be very computationally intensive with any given basis.

If we use something called the Gram-Schmidt process (which MATLAB already has built in), we can convert any basis into an **orthonormal** basis. An orthonormal basis consists of a set of orthogonal position vectors, each of length 1. This process gives us a new basis that has all of the same properties as the basis with which we started. The standard normal basis is an orthonormal basis.

How does this help anything? If we change the current basis we are working with into an orthonormal basis, the calculations required to find a solution to that equation become a good deal quicker and easier for a computer. In all of the examples we have done, the matrices are relatively small and so computation time isn't really a factor. However, if we have a matrix with hundreds but possibly tens of thousands of entries (not an unreasonable or unusual situation in some engineering contexts), then computation time will matter a great deal.

Consider the following matrix:

$$\mathbf{A} = \begin{pmatrix} 2 & 3 & 5 \\ -1 & 6 & 4 \\ 9 & 1 & 1 \end{pmatrix}$$

the first thing we do is check the determinant, using the MATLAB command

```
det(A)
```

which returns a value of -160. We know then that $\mathbf{A}$ is a basis. However, let's assume this basis is difficult to work with. If we want an easier basis to work with such that this new basis has all of the same properties of $\mathbf{A}$, we enter the command

```
>> Q = orth(A)
```

which returns

```
Q =
-0.5300 -0.3219 -0.7845
-0.4044 -0.7172 0.5675
-0.7454 0.6180 0.2500
```

Q is a matrix that has all of the same properties of **A**, but is much nicer to deal with from a computational perspective. Even if a matrix is singular, it is possible to derive an orthonormal matrix from it, although it will have a smaller rank than the original matrix.

Consider the matrix

$$\mathbf{C} = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 5 & 7 & 9 \end{pmatrix}$$

There are a variety of ways to check that $\mathbf{C}$ is not a basis. Checking MATLAB for the value of the determinant gives $3.3307e^{-15}$, which is the computational equivalent of zero. However, we can still use the `orth(C)` command on this matrix. If we use

```
>> Q = orth(C)
Q =
    -0.2354  0.7818
    -0.5594 -0.5948
    -0.7948  0.1870
```

is returned.

How can we interpret this response? MATLAB is telling us that it was able to generate two orthogonal vectors from **C**, meaning that **C** only forms a basis in 2 dimensions, which is the same information given when we find the rank of **C**. The nice thing about the orthonormal version of **C** is that we now know the vectors that form that basis. Remember, depending on the context you are working in, it may be impossible to get a matrix that forms a basis. If that is the case, you can still find the orthonormal basis and work with that instead.

Find the orthonormal basis of the following matrices:

$$\mathbf{A} = \begin{pmatrix} 2 & 3 & 7 \\ 1 & -9 & 0 \\ 0 & 5 & -1 \end{pmatrix} \quad \mathbf{B} = \begin{pmatrix} -6 & 1.1 & 8 & 3.2 \\ -4.5 & 10 & 6.6 & 8.9 \\ 0.5 & 5.8 & -7.1 & 2.3 \end{pmatrix}$$

$$\mathbf{C} = \begin{pmatrix} 2.4 & 9.3 \\ 6.1 & -2.9 \\ 10.1 & 5.4 \end{pmatrix} \quad \mathbf{D} = \begin{pmatrix} 8.6 & 7.2 & 8.8 & -2.6 \\ 2.5 & -0.9 & 12.8 & 3.3 \\ -2.7 & -3.8 & 4.7 & -8.9 \\ 1 & 0 & 0 & -4 \end{pmatrix}$$

$$\mathbf{A} = \begin{pmatrix} 0.4145 & -0.9053 & -0.0927 \\ -0.8042 & -0.3167 & -0.5030 \\ 0.4260 & 0.2831 & -0.8593 \end{pmatrix}$$

$$\mathbf{B} = \begin{pmatrix} -0.5069 & 0.4733 & 0.7204 \\ -0.8619 & -0.2699 & -0.4292 \\ -0.0087 & -0.8385 & 0.5448 \end{pmatrix}$$

$$\mathbf{C} = \begin{pmatrix} -0.5602 & 0.6471 \\ -0.2014 & -0.7119 \\ -0.8035 & -0.2727 \end{pmatrix}$$

$$\mathbf{D} = \begin{pmatrix} -0.7351 & 0.1751 & -0.6481 & 0.0949 \\ -0.6628 & 0.0007 & 0.7241 & -0.1905 \\ -0.1420 & -0.9357 & -0.0448 & 0.3200 \\ -0.0120 & 0.3064 & 0.2316 & 0.9232 \end{pmatrix}$$

Diagonalization

We saw earlier that changing a matrix into its orthonormal basis allowed for easier computation, and we could even find the orthonormal basis of a matrix that is not a basis. If the matrix we are working with is nonsingular, meaning it has a determinant not equal to zero, then there is a process that produces an even simpler matrix than the orthonormal basis. This process is called diagonalization.

If we have a square matrix $\mathbf{A}$, and the determinant of $\mathbf{A}$ is not equal to zero, then we can change $\mathbf{A}$ into the diagonalized matrix $\mathbf{D}$. $\mathbf{D}$ has all of the same properties of $\mathbf{A}$, but is simpler to work with and often has most entries (numbers in the matrix) as zero. The diagonalization of a matrix is often used in a variety of engineering contexts.

Let us see what a diagonalized matrix looks like:
Consider the matrix

$$\mathbf{A} = \begin{pmatrix} 8.6 & 7.2 & 8.8 & 2.6 \\ 2.5 & 0.9 & 12.8 & 3.3 \\ 2.7 & 3.8 & 4.7 & 8.9 \\ 1 & 1 & 2 & 4 \end{pmatrix}$$

In order to have MATLAB find the diagonalized matrix, we use the command

```
D = eig(A, 'matrix')
```

This will produce the matrix

$$\mathbf{D} = \begin{pmatrix} 17.6656 & 0 & 0 & 0 \\ 0 & -4.1211 & 0 & 0 \\ 0 & 0 & 3.0918 & 0 \\ 0 & 0 & 0 & 1.5638 \end{pmatrix}$$

A diagonalized matrix has a value of zero in every position except the diagonal, which has the matrix's eigenvalues. How do we interpret these diagonal values? Learning about eigenvalues will help with that interpretation.

Eigenvalues

The word **eigenvalue** of a matrix comes from German and it can be interpreted as meaning the "proper" or "characteristic" value. Only square matrices have eigenvalues, and those values can be real or complex. The mathematical definition of eigenvalue is as follows: Let the matrix $\mathbf{A}$ be square, the vector $\overrightarrow{x}$ be non-zero (meaning not all of its components are zero), and let λ be an eigenvalue of $\mathbf{A}$. Then

$$\mathbf{A}\overrightarrow{x} = \lambda\overrightarrow{x}$$

What is a useful interpretation of these values?

Recall that when a matrix is multiplied by a vector, the resulting vector can be moved, rotated, shrunk or enlarged. Eigenvectors are a way of describing what a matrix does to a vector during that multiplication process. For example, consider the matrix

$$\mathbf{M} = \begin{pmatrix} 1 & 1 \\ 2 & -2 \end{pmatrix}$$

and the vector $\overrightarrow{x} = \begin{bmatrix} 3 \\ 5 \end{bmatrix}$.

Use MATLAB to check that $\mathbf{M}\overrightarrow{x} = \begin{bmatrix} 8 \\ -4 \end{bmatrix}$. If we want to know the eigenvalues of $\mathbf{M}$, we can use the

```
eig(M)
```

command.

Using that command will return

```
ans =
    1.5616
   −2.5616
```

How to interpret these values? Recall that matrices operate on vectors, and vectors exist in more than one dimension. The first eigenvalue returned by MATLAB is how the matrix acts on the vector in the first dimension (usually what we refer to as the x-axis). In the case of our example, we can see it stretches the vector in the first dimension by a factor of 1.5616, keeping the same direction as the vector was originally pointed in along the first dimension. The matrix also stretches the vector in the second dimension by a factor of 2.5616, and then flips it across the second dimension (due to the negative). That is how we get $\mathbf{M}\overrightarrow{x} = \begin{bmatrix} 8 \\ -4 \end{bmatrix}$ as our answer. **Eigenvalues describe what a matrix does to a vector.** It can be very difficult to interpret them though.

Complex Eigenvalues

There is no reason that the eigenvalues of a matrix should always be real numbers. In fact, this is often not the case. Consider the matrix

$$\mathbf{C} = \begin{pmatrix} 3 & -9 \\ 4 & -3 \end{pmatrix}$$

If we use MATLAB to find the eigenvalues we get

```
ans =
     0.0000 + 5.1962i
     0.0000 - 5.1962i
```

Complex eigenvalues are a good deal more difficult to interpret, but a general rule to follow is that a matrix with complex eigenvalues rotates a vector as well as stretching or shrinking it.

A **symmetric** matrix is a matrix that has the following form:

$$\begin{pmatrix} d_1 & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & d_2 & a_{23} & \cdots & a_{2n} \\ \vdots & \ddots & & & \vdots \\ a_{n1} & a_{n2} & & \cdots & d_n \end{pmatrix}$$

That has the property that $a_{ij} = a_{ji}$. For example

$$\begin{pmatrix} 1 & 2 & 3 \\ 2 & 4 & 7 \\ 3 & 7 & 2 \end{pmatrix}$$

is a symmetric matrix. Whenever a symmetric matrix is diagonalized, it always has the form

$$\begin{pmatrix} d_1 & 0 & 0 & \cdots & 0 \\ 0 & d_2 & 0 & \cdots & 0 \\ \vdots & \ddots & & & \vdots \\ 0 & 0 & & \cdots & d_n \end{pmatrix}$$

and every value of d along the diagonal is a real number.

Diagonalize the following symmetric matrices.

$$\mathbf{A} = \begin{pmatrix} 1 & 4 & -1 & 2 \\ 4 & 8 & 3 & -3 \\ -1 & 3 & 7 & 5 \\ 2 & -3 & 5 & 2 \end{pmatrix}$$

$$\mathbf{B} = \begin{pmatrix} -2 & 9 & -1 & -2 \\ 9 & -3 & 3 & 1 \\ -1 & 3 & -4 & 5 \\ -2 & 1 & 5 & -5 \end{pmatrix}$$

$$\mathbf{D}_A = \begin{pmatrix} -4.9872 & 0 & 0 & 0 \\ 0 & 2.3045 & 0 & 0 \\ 0 & 0 & 9.3853 & 0 \\ 0 & 0 & 0 & 11.3243 \end{pmatrix}$$

$$\mathbf{D}_B = \begin{pmatrix} -12.5216 & 0 & 0 & 0 \\ 0 & -9.6117 & 0 & 0 \\ 0 & 0 & 1.4447 & 0 \\ 0 & 0 & 0 & 6.6886 \end{pmatrix}$$

Why is diagonalization important? Interestingly enough, these concepts are used in the analysis of combined stress on surfaces, like tunnels and bridges. A combined stress can't be represented by a single vector, since the pressure on something like a bridge or tunnel comes from multiple directions. So we create a 3 dimensional matrix (covering the three directions, which is a combination of vectors, to describe that stress. This matrix has some really useful properties, like the fact that it is symmetric. The reason for the symmetry of this matrix has do with laws of conservation. This means that we can take the matrix representing stress on an object and diagonalize it. The diagonalized matrix is a lot easier to analyze and implement in a numerical modelling problem.

Consider the following matrix of stresses on an underground pipe at a certain point along the pipe.

$$\mathbf{S} = \begin{pmatrix} 131.22 & -44.7 & 114.1 \\ -44.7 & 77.43 & 391.32 \\ 114.1 & 391.32 & 200.65 \end{pmatrix}$$

If we diagonalize $\mathbf{S}$, we get

$$\mathbf{D} = \begin{pmatrix} -285.4322 & 0 & 0 \\ 0 & 151.1862 & 0 \\ 0 & 0 & 543.5460 \end{pmatrix}$$

The values along the diagonal of the matrix $\mathbf{D}$ are the eigenvalues of the matrix $\mathbf{S}$, which represent the combined stresses in each dimension. So along the first dimension at the point at which we care about, the pipe is experiencing a combined stress of -285.4322, which also happens to be the first eigenvalue of $\mathbf{S}$.

Using MATLAB to solve systems of ordinary differential equations

Differential equations can come in the form of a system of linked equations. For example, consider the Van der Pol equation:

$$y'' - \mu(1 - y^2)y' + y = 0,\ y(0) = 2,\ y'(0) = 0$$

This is a second-order differential equation, but it can be solved using some interesting algebra and with the help of MATLAB. The first step in solving a second-order differential equation like this is to add extra variables and create a system. We first let $y = y_1$, and then let $y' = y_1' = y_2 \to y'' = y_2'$. This is usually the first step in converting any second-order differential equation into a system of first order differential equations. Now we have:

$$y_1' = y_2 \text{ and } y_2' = \mu(1 - y_1^2)y_2 - y_1$$

See the accompanying video for details, and how to plot this system in MATLAB.

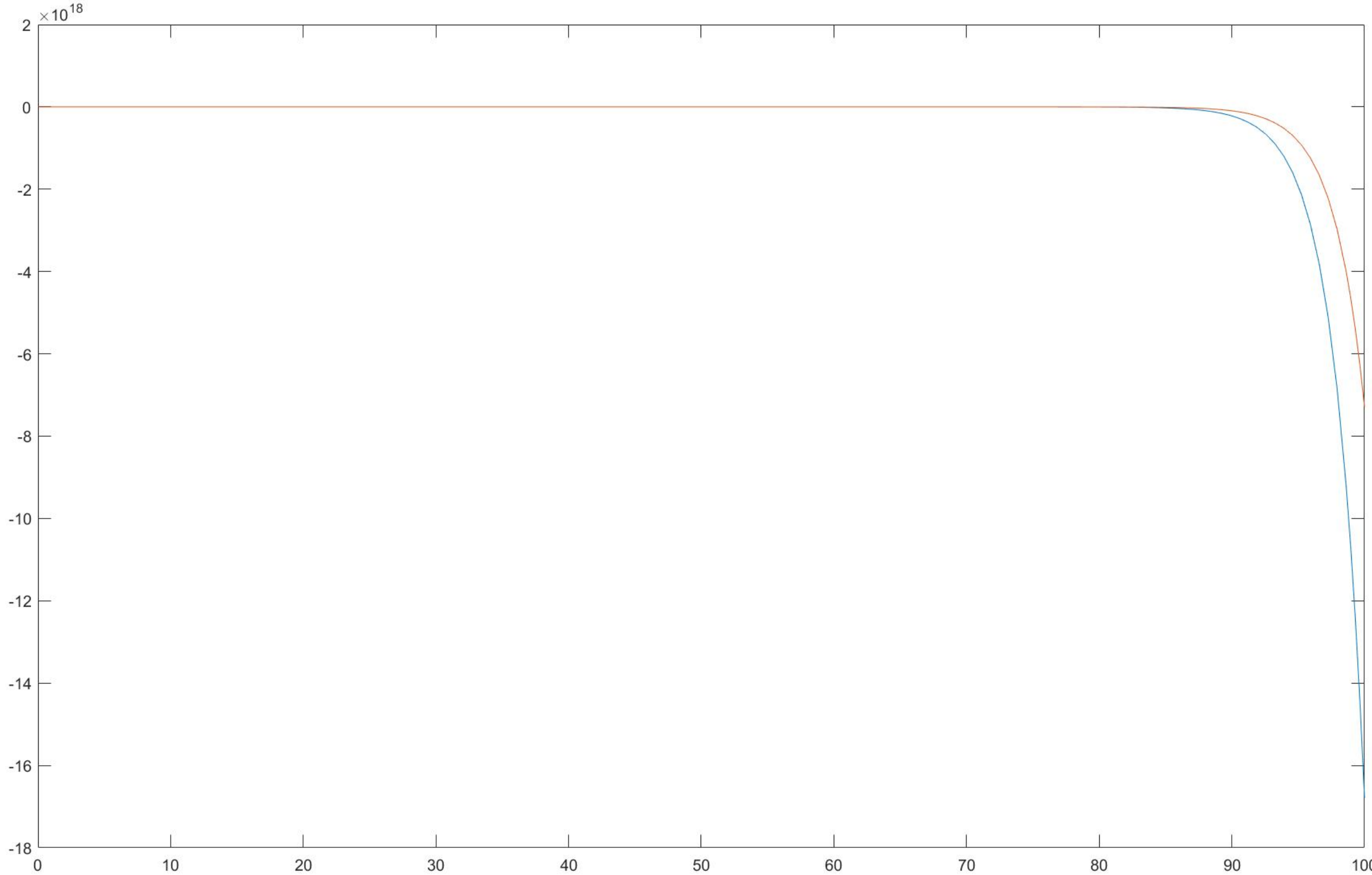
Convert these second order ODEs into systems of first order ODEs and use MATLAB to plot the solution.

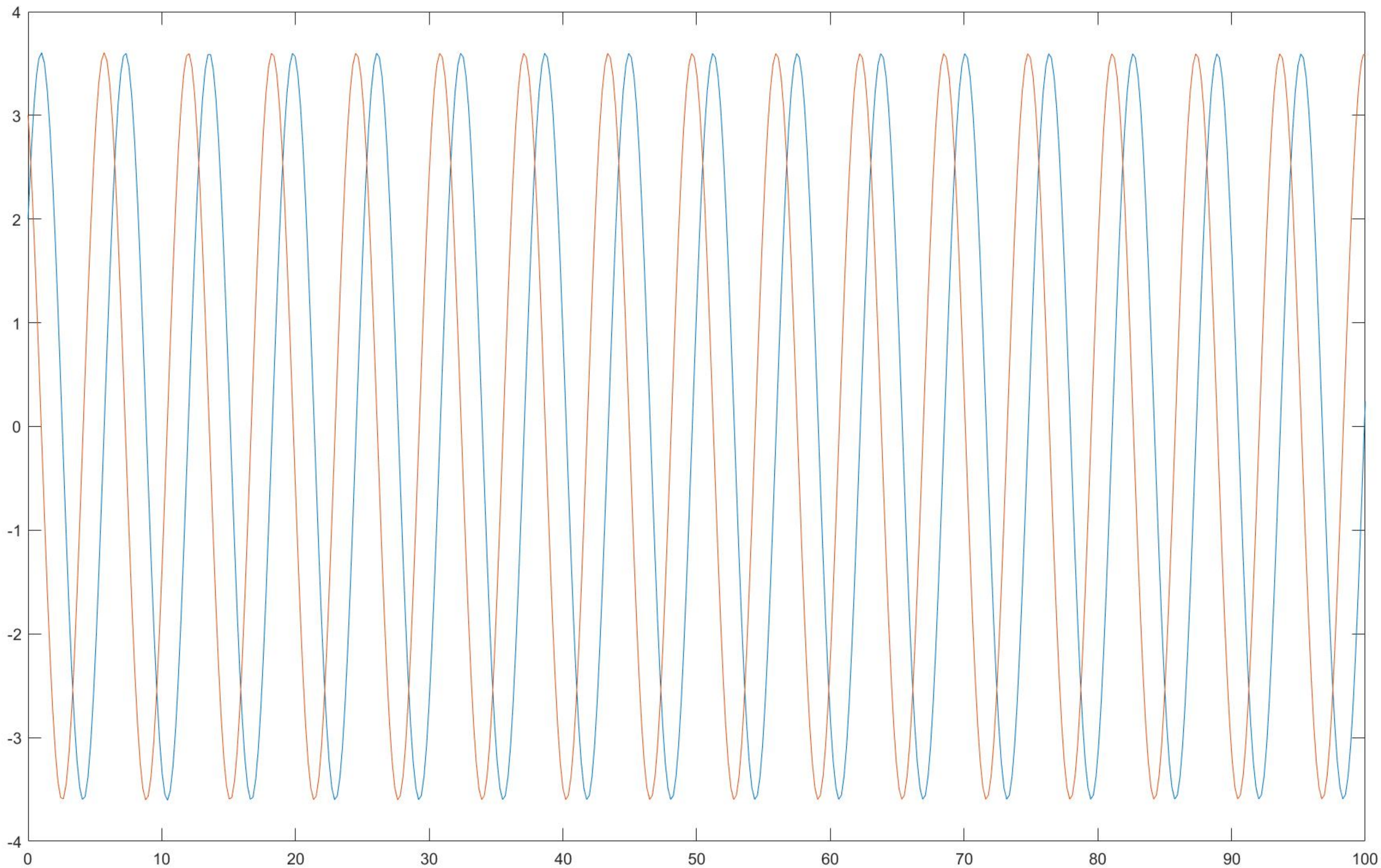
$$y'' - yy' + 2 = 0,\ y(0) = 1,\ y'(0) = 1$$

$$y'' + 11y' + 24y = 0,\ y(0) = 0,\ y'(0) = -7$$

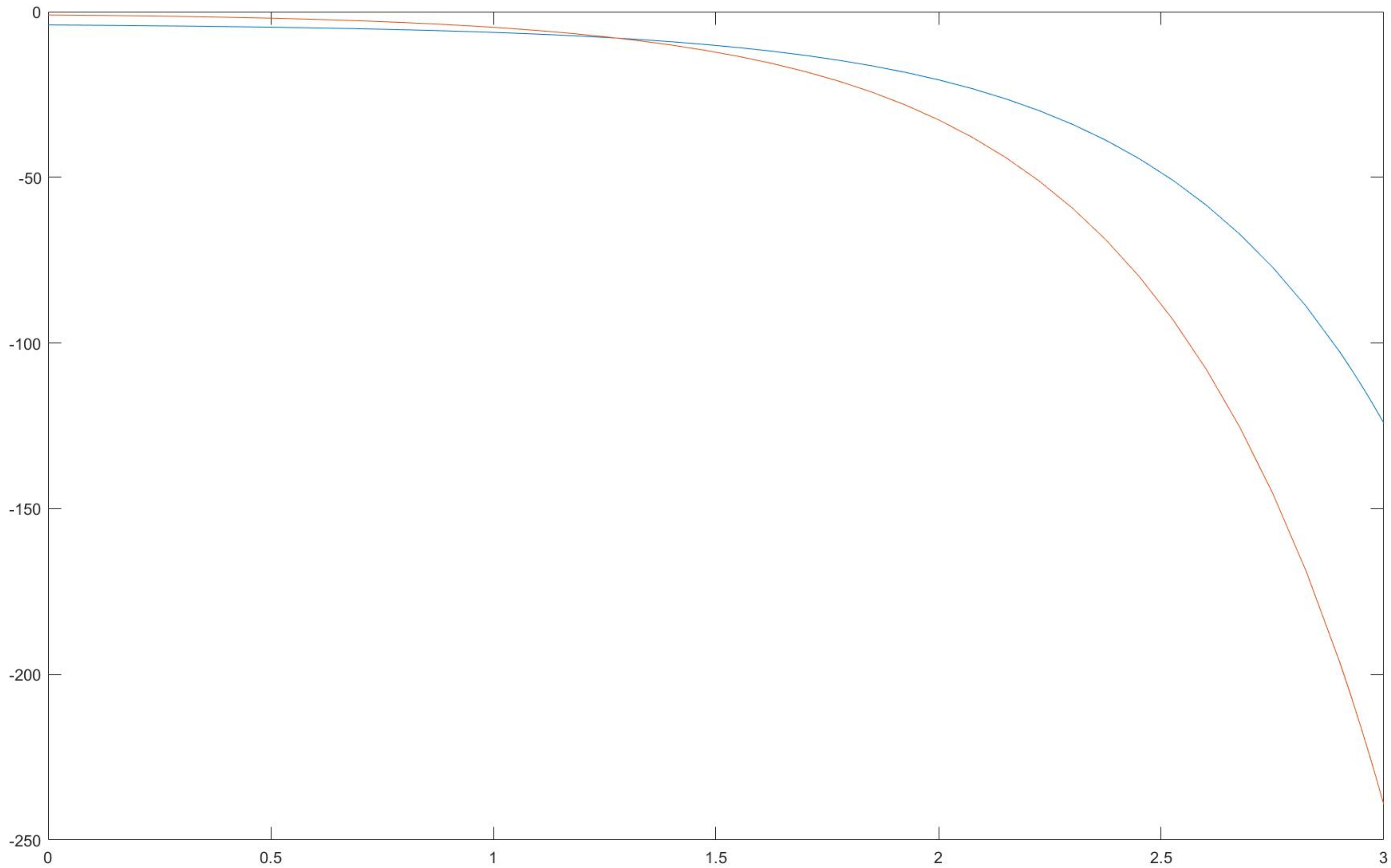
$$y'' - 8y' + 17y = 0,\ y(0) = -4,\ y'(0) = -1$$

System 1: $y_1' = y_2$ and $y_2' = y_1 y_2 - 2$
System 2: $y_1' = y_2$ and $y_2' = -11y_2 - 24y_1$
System 3: $y_1' = y_2$ and $y_2' = 8y_2 - 17y_1$





4
3
2
1
0
-1
-2
-3
-4
0
10
20
30
40
50
60
70
80
90
100


0
-50
-100
-150
-200
-250
0
0.5
1
1.5
2
2.5
3