3 Bases

In Section 1 we defined vectors as abstract algebraic entities that can be added and scaled in a certain way. We next defined vector spaces as collections of vectors that are closed under vector operations. Now that you know what the fundamental objects of linear algebra are, on the abstract level, it is time to make them more tangible and amenable to machine computations.

In order to perform vector computations in `Matlab`, or any other computer language for that matter, one has to be able to represent vectors in a way that computers can understand. The following code snippet shows how vector algebra works in `Matlab`.

```
x = [1;2;3];
y = [4;5;6];
a = -2;
s = x + y;  % vector addition
z = a*x;    % scalar multiplication
```

Run this code and examine the output: notice that we work in $\mathbb{R}^3$ whose elements we represent as *column* vectors. We could also work with *row* vectors, as we shall do every now and then. At the moment, we need not distinguish between column vectors and row vectors. Yet, there is a subtle difference which will become important in the future.

`Matlab` supports complex numbers, so working in $\mathbb{C}^n$ is just as simple. The following snippet computes a linear combination of two vectors in $\mathbb{C}^2$:

```
x = [1+2*i;3+4*i];
y = [5+6*i;7+8*i];
a = -2;
b = 2*i;
s = a*x + b*y;  % linear combination in C^2
```

All of this is well and good. Yet, what if our vector space consists of, say, polynomials in one variable? How do we use `Matlab` to manipulate vectors that are not columns of real or complex numbers?

The answer is profoundly simple but requires some new notions that are introduced below. As always, we encourage the reader to think about the question and formulate an opinion before proceeding to the answer.

3.1 Linear independence

We start with the fundamental idea of linear independence which is a characteristic of finite, possibly empty, collections of vectors. In the following definition, which applies to all vector spaces indiscriminately, by a 'nontrivial' linear combination we mean the one in which at least one scalar coefficient is nonzero.

Definition 3 (Linear independence)**.** *A finite set of vectors $\{x_i\}_{i=1}^n$ is said to be linearly independent if all nontrivial linear combinations of these vectors are nonzero. In other words, the vanishing of a linear combination*

$$\sum_{i=1}^{n} \alpha_i \, x_i = 0$$

implies that all scalar coefficients are zero: $\alpha_i = 0$ for each $i = 1, \ldots, n$.

As a simple example, consider the following pair of vectors in $\mathbb{R}^3$:

$$x_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad x_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

These two vectors, which you may have recognized as $\mathbf{i}$ and $\mathbf{j}$, are linearly independent. Indeed, suppose that their linear combination vanishes:

$$\alpha_1 \, x_1 + \alpha_2 \, x_2 = \alpha_1 \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + \alpha_2 \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

Then, necessarily, both α_1 and α_2 must be zero, so the combination is trivial.

As a counterexample, consider these three vectors in $\mathbb{R}^2$:

$$x_1 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \quad x_2 = \begin{bmatrix} 3 \\ 4 \end{bmatrix}, \quad x_3 = \begin{bmatrix} 5 \\ 6 \end{bmatrix}.$$

You can easily check that $x_1 - 2\,x_2 + x_3 = 0$. This is a nontrivial zero combination, so the vectors are linearly dependent.

If a collection of vectors is linearly dependent then some vectors in that collection can be expressed as linear combinations of other vectors from the same collection, e.g., in the last example $x_3 = -x_1 + 2\,x_2$. If vectors represent some sort of data, then linear dependence can be loosely thought of as redundancy in the data set.

Finally, at the start of this section we suggested that Definition 3 may be applied to *empty* collections of vectors. It is safe to say that we will not run into situations where we will have to allow for the possibility of a vector collection being completely devoid of vectors. Nevertheless, as a linear algebra curiosity, we will state that empty collections of vectors are considered linearly independent by "vacuous implication": if there are no vectors, it is impossible to form a nontrivial vanishing linear combination, or any linear combination at all.

3.2 Span

Let V be a vector space defined over some scalar field F and let $\{x_i\}_{i=1}^n$ be a finite collection of vectors in V.

Definition 4 (Span)**.** *The span of $\{x_i\}_{i=1}^n$ is the set of all linear combinations:*

$$\operatorname{span}\{x_i\}_{i=1}^n = \left\{ \sum_{i=1}^n \alpha_i \, x_i \mid \alpha_i \in F \right\}.$$

The key thing to notice is that the span of a collection of vectors in V is a much larger collection of vectors in V which, by construction, is closed under vector addition and scalar multiplication. The output of span is thus a vector space residing in V—a vector *subspace* of V.

The following are three examples in $\mathbb{R}^2$ which provide a good segue to the next section. Let

$$x_1 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \quad x_2 = \begin{bmatrix} 3 \\ 4 \end{bmatrix}, \quad x_3 = \begin{bmatrix} 5 \\ 6 \end{bmatrix},$$

as before. Then

$$S_1 = \operatorname{span}\{x_1\} = \{\alpha \, x_1 \mid \alpha \in \mathbb{R}\} = \left\{ \begin{bmatrix} \alpha \\ 2\,\alpha \end{bmatrix} \mid a \in \mathbb{R} \right\}.$$

In words, S_1 is the set of all scalar multiples of x_1; it is an example of a one-dimensional subspace of $\mathbb{R}^2$. Any one-dimensional vector (sub)space consists of scalar multiples of the same nonzero vector.

As the next example, consider

$$\begin{aligned} S_2 &= \operatorname{span}\{x_1, x_2\} = \{\alpha_1 \, x_1 + \alpha_2 \, x_2 \mid \alpha_1, \alpha_2 \in \mathbb{R}\} \\ &= \left\{ \begin{bmatrix} \alpha_1 + 3\,\alpha_2 \\ 2\,\alpha_1 + 4\,\alpha_2 \end{bmatrix} \mid \alpha_1, \alpha_2 \in \mathbb{R} \right\}. \end{aligned}$$

How many vectors from $\mathbb{R}^2$ does S_2 contain? To answer this question, let

$$b = \begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix}$$

be an *arbitrary* vector in $\mathbb{R}^2$; we emphasize the word 'arbitrary' because it is part of the argument. The following linear system

$$\begin{aligned} \alpha_1 + 3\,\alpha_2 &= \beta_1 \\ 2\,\alpha_1 + 4\,\alpha_2 &= \beta_2 \end{aligned}$$

has the unique solution:

$$\alpha_1 = -2\,\beta_1 + \frac{3}{2}\,\beta_2, \quad \alpha_2 = \beta_1 - \frac{1}{2}\,\beta_2.$$

Consequently, $b = \alpha_1\,x_1 + \alpha_2\,x_2$ with the above choice of α_1 and α_2: hence $b \in S_2$. Since b was picked *arbitrarily* in $\mathbb{R}^2$, we conclude that S_2 contains *all* vectors in $\mathbb{R}^2$ and therefore *is* $\mathbb{R}^2$. Incidentally, this shows why $\mathbb{R}^2$ is two-dimensional: it has a *basis* $\{x_1, x_2\}$ which has two vectors.

Finally, consider

$$S_3 = \text{span}\{x_1, x_2, x_3\} = \{\alpha_1\,x_1 + \alpha_2\,x_2 + \alpha_3\,x_3 \mid \alpha_1, \alpha_2, \alpha_3 \in \mathbb{R}\}$$

Since α_3 can be zero, S_3 contains $S_2 = \mathbb{R}^2$. On the other hand, S_3 is contained in $\mathbb{R}^2$. We conclude that $S_3 = \mathbb{R}^2$. So, the addition of a new vector—x_3—to the collection $\{x_1, x_2\}$ does not change the span. This is an instance of a general phenomenon whereby span remains unaffected if one adds vectors that depend linearly on the vectors already present.

3.3 Bases and dimension

Having defined linear independence and spans, we are ready to define bases.

Definition 5 (Basis)**.** *A finite collection of vectors $\{x_i\}_{i=1}^n$ is a basis of a vector space V if:*

1. *The vectors $\{x_i\}_{i=1}^n$ are linearly independent.*
2. $\text{span}\{x_i\}_{i=1}^n = V$.

Notice the use of the indefinite article in the opening sentence of Definition 5: we defined *a* basis. Except for the trivial vector space, whose basis is its only vector—namely, the zero vector—any vector space has infinitely many different bases. As you will soon find out, this is actually a good thing! You will also soon learn that some bases are better than others. However, whatever basis one chooses in a given vector space, it will always have the same number of vectors (exercise). This motivates the next definition.

Definition 6 (Dimension)**.** *The dimension of a vector space is the number of vectors in any of its bases.*

We have already given an example of a basis of $\mathbb{R}^2$ in the previous section. Here is another, possibly familiar basis of $\mathbb{R}^2$, called the *standard basis*:

$$e_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad e_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Showing that $\{e_1, e_2\}$ are linearly independent and span $\mathbb{R}^2$ is left as an easy exercise. We note that henceforth we will follow the usual practice of using the symbols e_i to denote the standard basis vectors in $\mathbb{R}^n$. So, unless noted otherwise, e_i is a column vector whose i-th component is 1 and whose other components are all zero. *Apropos* of Definition 6, since the standard basis $\{e_i\}_{i=1}^n$ of $\mathbb{R}^n$ has n vectors, $\mathbb{R}^n$ is n-dimensional, as its symbol clearly suggests. Keep in mind, however, that not every space that carries n as a sub- or superscript is necessarily n-dimensional. For instance, we may use the symbol P_2 to denote the space of quadratic polynomials in one variable:

$$P_2 = \{a\,x^2 + b\,x + c \mid a, b, c \in \mathbb{R}\}.$$

Here the subscript "2" refers to the *degree* of the polynomials. Meanwhile, the dimension of P_2 is readily seen to be three. The next section explains that and gives further examples of bases.

3.4 Vector components

With all of the technical definitions in place, we are finally ready for the main point of this handout. Recall that our overarching concern is with *representation* of vectors in a manner suitable for machine computations. If a problem takes place in some strange vector space where vectors are "weird", how can we use `Matlab` to perform vector computations? The answer, as we have already intimated, is profoundly simple: all we need to do is pick a basis.

As a concrete example, suppose that we want to work with quadratic polynomials—the space we denoted P_2 in the previous section. Any $p \in P_2$ can be written uniquely in the form:

$$p = a\,x^2 + b\,x + c.$$

This suggests a basis of monomials $\{x^2, x, 1\}$; it will become clear in a moment why we wrote the monomials in descending order. In the meantime, let $p_1, p_2 \in P_2$ be any two quadratics and let us say that we want to compute their sum s. Write:

$$p_1 = a_1\,x^2 + b_1\,x + c_1, \quad p_2 = a_2\,x^2 + b_2\,x + c_2.$$

These are expansions of p_1 and p_2 *with respect to the basis* $\{x^2, x, 1\}$. On one hand,

$$p_1 + p_2 = (a_1 + a_2)\,x^2 + (b_1 + b_2)\,x + (c_1 + c_2).$$

On the other hand, to compute s we merely added the like coefficients in the basis expansions. If we agree to *represent* polynomials by listing their coefficients in descending order, we can add p_1 and p_2 in `Matlab` like so:

```
% we assume that a1,b1,c1,a2,b2,c2 have been defined
p1 = [a1 b1 c1];
p2 = [a2 b2 c2];
s = p1 + p2;
```

The result, clearly, is `[a1+a2 b1+b2 c1+c2]` which we *interpret* as the coefficients of the quadratic:

$$s = (a_1 + a_2)\,x^2 + (b_1 + b_2)\,x + (c_1 + c_2).$$

General linear combinations of quadratics, or polynomials of any order, can be computed in a similar manner. In fact, this is exactly how one works with polynomials in `Matlab`. Just remember that `Matlab` assumes that the coefficients of a polynomial are listed in *descending* order and that it infers the degree of the polynomial from the number of coefficients. Thus `[2 1]` represents $2\,x + 1$ while `[2 1 0]` is $2\,x^2 + x$. Also, it is advisable to input the coefficients of a polynomial into `Matlab` as a *row* vector. You certainly can input polynomials as column vectors. However, `Matlab`'s polynomial routines, such as `polyder`, output polynomials as row vectors and sometimes throw errors when the input is not a row.

Quite generally, let V be an n-dimensional vector space and let $\{x_i\}_{i=1}^n$ be a basis of V, *any* basis. Any vector $v \in V$ can be written in a unique way as a linear combination of the basis vectors:

$$v = \sum_{i=1}^{n} \alpha_i\,x_i.$$

The proof that basis expansions are unique is left as an exercise. The scalar coefficients α_i in the basis expansion are called *components* of v *with respect to the basis* $\{x_i\}_{i=1}^n$. Notice that components depend on the choice of the basis! Change the basis and the components change. However, once the basis has been chosen and fixed, the vector components become fixed as well. As the example of quadratics shows, all vector operations can be performed *componentwise*. Therefore, if vectors are represented as ordered lists of components, which is to say, as columns (or rows) of numbers, we can perform all computations in `Matlab` as if the vectors are in $\mathbb{R}^n$ (or $\mathbb{C}^n$)! In abstract language,

> All n-dimensional real vector spaces are isomorphic to $\mathbb{R}^n$. Similarly, all n-dimensional complex spaces are isomorphic to $\mathbb{C}^n$.

As much as I would like to formally define *isomorphisms*, I will resist the temptation. The term, when translated from Latin, means 'made in the same way.' All finite dimensional vector spaces are made in the same way and can be identified, through choices of bases, with n-tuples of scalars. Since the scalars in Math 145 will be usually real or complex numbers, most of our computations will end up taking place either in $\mathbb{R}^n$ or $\mathbb{C}^n$.

3.5 Matrices

In linear algebra there are relatively few questions that pertain solely to vector spaces. It should now be clear why: all finite dimensional vector spaces are isomorphic! If the scalars are real or complex numbers, there are, really, only two kinds of spaces: $\mathbb{R}^n$ and $\mathbb{C}^n$. And soon you will know all there is to know about them! Most of the interesting (and often difficult!) problems in linear algebra arise in connection with linear transformations (if you forgot what that means, look it up in Section 2). Given a linear transformation $T : V \mapsto W$ one might be interested in any or all of the following questions:

1. Compute $T(v)$ for a given $v \in V$. E.g., rotate a set of geometric objects in the plane or in space by some angle.

2. Given $w \in W$, determine if there is a $v \in V$ such that $T(v) = w$. This is the so-called range characterization problem. A simple example is determining wether there is a rotation which brings two points into coincidence.

3. Given w in the range of T, determine if there is a unique $v \in V$ such that $T(v) = w$. This is, as you may have already guessed, the uniqueness problem.

4. If there are several solutions of $T(v) = w$, find the one satisfying some additional constraints. The constraints usually come from physics or some other context. E.g., the solution vector may need to have nonnegative components, if it represents annual production of some commodity. Or it may need to have the smallest length which could be synonymous with energy. And so on.

5. If the linear equation $T(v) = w$ does not have a solution, due to noise in the data w, say, find the best candidate for the solution. Most linear problems in applied mathematics are of this nature.

If T is a linear operator on V, that is, if T maps V into itself, we may apply it repeatedly. This leads to a host of interesting questions specific to linear operators on vector spaces.

1. How does one efficiently compute powers $T^n(v)$?

2. What is $\lim_{n\to\infty} T^n(v)$?

The first step in answering these and similar questions is representing T in `Matlab`-friendly form. Once again, we need to use bases!

Let us consider the completely general situation of a linear transformation T mapping an n-dimensional vector space V into an m-dimensional vector space W. Since there are two vector spaces involved, we need to choose two bases, one in each space: say, $\{x_i\}_{i=1}^n$ in V and $\{y_j\}_{j=1}^m$ in W. Pick an arbitrary vector $v \in V$. Since we have selected a basis in V, we can expand that vectors with respect to the basis: $v = \sum_{i=1}^n \alpha_i\, x_i$. We now apply T to v. By linearity,

$$T(v) = T\left(\sum_{i=1}^{n} \alpha_i\, x_i\right) = \sum_{i=1}^{n} \alpha_i\, T(x_i).$$

Evidently, in order to continue the computation, we need to know what T does to the *basis vectors* of its domain space. Well, it maps them into some vectors in W which can be represented as linear combinations of the basis vectors of W—the ones we denoted y_j:

$$T(x_i) = \sum_{j=1}^{m} c_{ij}\, y_j, \quad i = 1, \ldots, n. \tag{1}$$

Notice how working with two bases forces us to use two indices in Equation (1). We can now express $T(v)$ as a linear combination of basis vectors in W:

$$T(v) = \sum_{i=1}^{n} \alpha_i \sum_{j=1}^{m} c_{ij}\, y_j = w. \tag{2}$$

Observe that all of the information about T is now encoded in a rectangular array of numbers c_{ij}; Equation (1) is the definition of these numbers.

So, how do we make `Matlab` compute $w = T(v)$? Firstly, we represent vectors in V and W as columns of their (scalar) components:

$$v = \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{bmatrix}, \quad w = \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_m \end{bmatrix}.$$

Next we use Equation (2) to relate the *components* of v and w. It is evident that the components of the former are linear combinations of the components of the latter. For instance, β_1 is the scalar coefficient in front of y_1 (by definition) in Equation (2):

$$\beta_1 = \sum_{i=1}^{n} c_{i1}\, \alpha_i.$$

More generally,

$$\beta_j = \sum_{i=1}^{n} c_{ij}\, \alpha_i, \quad j = 1, \ldots, m.$$

At this point it seems natural to introduce the *matrix of* T as a rectangular array

$$C = \begin{bmatrix} c_{11} & \cdots & c_{n1} \\ \vdots & \ddots & \vdots \\ c_{1m} & \cdots & c_{nm} \end{bmatrix},$$

and define matrix-vector multiplication using the *row-by-column rule*:

$$\begin{bmatrix} c_{11} & \cdots & c_{n1} \\ \vdots & \ddots & \vdots \\ c_{1m} & \cdots & c_{nm} \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{bmatrix} = \begin{bmatrix} c_{11}\, \alpha_1 + \cdots + c_{n1}\, \alpha_n \\ \vdots \\ c_{1m}\, \alpha_1 + \cdots + c_{nm}\, \alpha_n \end{bmatrix}$$

Indeed, with this definition of matrix-vector multiplication, $T(v) = C\, v$. And this is exactly how matrix-vector multiplication is defined in `Matlab`.

Let us reinforce this very important discussion by revisiting an old example from Section 2. There we considered differentiation of quadratics which is, clearly, a linear operation. Let P_2 denote quadratics, as before, and let P_1 be the first order polynomials which are the derivatives of quadratics. In order to represent the derivative $D : P_2 \mapsto P_1$ in matrix form, we first need to choose bases. The simplest thing

to do is to use the bases of monomials: $\{x^2, x, 1\}$ for P_2 and $\{x, 1\}$ for P_1, respectively. Next we compute the numbers c_{ij} by applying D to basis elements of P_2 and expanding the resulting vectors in P_1 with respect to the basis of P_1:

$$\begin{aligned} D(x^2) &= 2\,x = 2 \cdot x + 0 \cdot 1, \\ D(x) &= \ \ 1 = 0 \cdot x + 1 \cdot 1, \\ D(1) &= \ \ 0 = 0 \cdot x + 0 \cdot 1. \end{aligned}$$

The matrix of D is therefore:

$$D = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}.$$

To see that D is indeed represented by the 2-by-3 matrix above, let

$$p = a\,x^2 + b\,x + c = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

be an arbitrary vector in P_2. Then

$$D(p) = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 2\,a \\ b \end{bmatrix} = 2\,a\,x + b,$$

as required. The computation can be easily carried out in `Matlab`:

```
% we assume that a,b,c have been defined
p = [a;b;c];
D = [2 0 0; 0 1 0];
dp = D*p;  % compute the derivative using matrix multiplication
```

We conclude with a few closing remarks.

1. In the future, we will usually identify linear transformations with matrices that represent them. Keep in mind, however, that a linear transformation is not completely synonymous with a matrix.

2. In general, the matrix representation of a linear transformation $T : V \mapsto W$ depends on two separate choices of bases. So, one and the same transformation can be represented with different matrices. Infinitely many different matrices, as a matter of fact. Echoing our earlier comment about vector components, choice is good! Much of our discussion will soon revolve around construction of bases which simplify linear transformations.

3. If $T : V \mapsto V$ maps a vector space into itself, we can find its matrix representation by choosing a single basis. It is, however, still possible and often advantageous to represent T in matrix form using two different bases in one and the same space V.

4. You may have noticed that matrix representation of a linear operation involves a transposition. The entries of the matrix are read from n linear combinations of m vectors. Yet these entries are transposed into an array with m rows and n columns. Watch out for that! Keep in mind that the number of columns must match the dimension of the domain space while the number of rows much match the dimension of the range space.

5. The row-by-column rule of matrix-vector multiplication explains why vectors are best represented as *columns*. Traditionally, the matrix in a matrix-vector product is placed on the *left*. Therefore the vector must be a column whose length must match the length of the rows of the matrix.

6. Although we defined only matrix-*vector* multiplication, the row-by-column rule readily extends to matrix-matrix multiplication. The latter, called simply *matrix multiplication*, is the *default* multiplication in `Matlab`. For example, multiplying a 3-by-2 matrix with a 2-by-3 matrix

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \end{bmatrix}$$

   produces a 3-by-3 matrix

$$\begin{bmatrix} a_{11}\,b_{11} + a_{12}\,b_{21} & a_{11}\,b_{12} + a_{12}\,b_{22} & a_{11}\,b_{13} + a_{12}\,b_{23} \\ a_{21}\,b_{11} + a_{22}\,b_{21} & a_{21}\,b_{12} + a_{22}\,b_{22} & a_{21}\,b_{13} + a_{22}\,b_{23} \\ a_{31}\,b_{11} + a_{32}\,b_{21} & a_{31}\,b_{12} + a_{32}\,b_{22} & a_{31}\,b_{13} + a_{32}\,b_{23} \end{bmatrix}$$

   In fact, `Matlab` regards column vectors as n-by-1 matrices. Likewise, row vectors are 1-by-m matrices while numbers are 1-by-1 matrices. So, matrix-vector multiplication in `Matlab` is really just matrix multiplication.

7. Only *conformable* matrices can be multiplied. As an illustration of non-conformism, consider the following piece of code:

```
x = [1;2;3];
y = [4;5;6];
z = x*y;
```

Beginners assume that since `Matlab` adds arrays componentwise, it multiplies them componentwise as well. Therefore the often anticipated results is the column vector `[4;10;18]`. Instead `Matlab` attempts matrix multiplication and throws the error:

```
??? Error using ==> mtimes
Inner matrix dimensions must agree.
```

Indeed, the length `y` is three but the rows of `x` have length one. If you need to multiply to arrays of the same dimension componentwise, precede the multiplication operator with a dot. As in the following code snippet:

```
x = [1;2;3];
y = [4;5;6];
z = x.*y

ans =

  4
 10
 18
```

On this note, do not pepper your code with dots in an effort to avoid potential `mtimes` errors. Sometimes you need the dot and sometimes you do not. Always put thought into your programming.

Homework

1. In this problem P_2 stands for the space of quadratics with real coefficients defined on the interval $-1 \leq x \leq 1$; $\mathcal{B} = \{1, x, \frac{3}{2}x^2 - \frac{1}{2}\}$; and $p = x^2 + x + 1$. The three vectors in $\mathcal{B}$ are the first three *Legendre polynomials.*

   (a) Provide a convincing explanation showing that $\mathcal{B}$ is a basis of P_2.

   (b) Represent p as a column vector using the basis $\mathcal{B}$

2. Let
$$x_1 = \begin{bmatrix} -98 \\ 72 \\ -176 \end{bmatrix}, \quad x_2 = \begin{bmatrix} -239 \\ 162 \\ 115 \end{bmatrix}, \quad x_3 = \begin{bmatrix} 186 \\ -114 \\ -573 \end{bmatrix}.$$
Do these vector form a basis of $\mathbb{R}^3$? Explain your answer clearly and provide supporting computations.

3. Let V be the real vector space of harmonics of frequency ω:
$$V = \{a\,\cos(\omega\,t) + b\,\sin(\omega\,t) \mid a, b, \in \mathbb{R}\}.$$
For $v \in V$, define: $T(v) = \frac{d^2v}{dt^2} + 3\,\frac{dv}{dt} + 2\,v$.
   (a) Explain why T is a linear.
   (b) Find a matrix representation of T using any bases.
   (c) Suppose T is applied twice in a row. The result is a linear transformation denoted T^2. Use the previous part to find its matrix.
   (d) Use `Matlab` to compute the matrix of T^n for $n = 10$; set $\omega = 2$.
   (e) Most choices of bases produce matrices that are *dense* meaning that they have very few zeros, if any. Try to find a basis or bases of V which lead to a matrix of T with at least one zero element.

4. Consider the problem of solving $T(v) = w$ where T is the linear transformation from the preceding problem. Find an explicit formula for v or explain why it does not exist.

5. We will say that a linear operator $T : V \mapsto V$ is *singular* if there exists a nonzero vector in V whose image under T is zero. Give one example of a singular and one example of a nonsingular linear operator acting on the space of quadratics P_2.

6. In `Matlab` the length of a vector can be computed using the `norm` command. As in the following snippet:

```
x = [1;2];
norm(x)


ans =
```

```
2.236067977499790
```

We will use the absolute value symbol $|x|$ to denote norms. Let

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

Pick a nonzero vector $x \in \mathbb{R}^2$ and compute the ratio $|A^{n+1}\,x|/|A^n\,x|$ for $n = 1, \ldots, 50$. Plot the resulting sequence of values. What is your observation? Does the result change if x is changed?

7. Find an example of a nonzero three-by-three matrix A with the following properties:

   (a) $A^2 = 0$

   (b) $A^2 = A$