Chapter III

Orthogonality

III.1. Projections

Prerequisites and Learning Goals

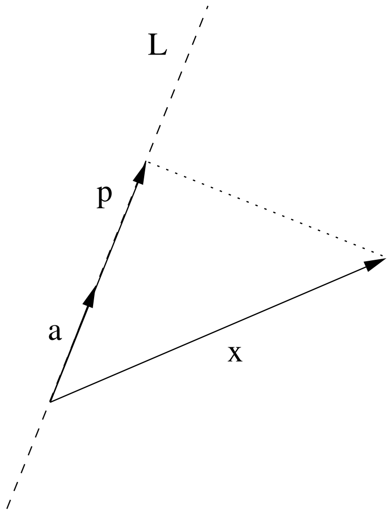
After completing this section, you should be able to

- Write down the definition of an orthogonal projection matrix and determine when a matrix is an orthogonal projection.
- Identify the range of a projection matrix as the subspace onto which it projects and use properties of the projection matrix to derive the orthogonality relation between the range and nullspace of the matrix.
- Given the projection matrix onto a subspace S, compute the projection matrix onto $S^{\perp}$; describe the relations between the nullspace and range of the two matrices.
- Use orthogonal projection matrices to decompose a vector into components parallel to and perpendicular to a given subspace.
- Explain how the problem of finding the projection of a vector $\mathbf{b}$ onto a certain subspace spanned by the columns of a matrix A translates into finding a vector $\mathbf{x}$ that minimizes the length $\|A\mathbf{x} - \mathbf{b}\|$; show that $\mathbf{x}$ always exists and satisfies the least squares equation; discuss how the results of the minimization problem can vary depending on the type of norm used; discuss the sensitivity of a least squares fit to outliers.
- Compute the orthogonal projection matrix whose range is the span of a given collection of vectors.
- Perform least squares calculations to find polynomial fits to a given set of points, or in other applications where overdetermined systems arise. You should be able to perform all necessary computations and plot your results using MATLAB/Octave.
- Interpret the output of the MATLAB/Octave \ command when applied to systems that have no solutions.

III.1.1. Warm up: projections onto lines and planes in $\mathbb{R}^3$

Let $\mathbf{a}$ be a vector in three dimensional space $\mathbb{R}^3$ and let $L = \text{span}(\mathbf{a}) = \{s\mathbf{a} : s \in \mathbb{R}\}$ be the line through $\mathbf{a}$. The line L can also be identified as the range $R(\mathbf{a})$, where $\mathbf{a}$ is considered to be a 3×1 matrix.

The *projection* of a vector $\mathbf{x}$ onto L is defined to be the vector in L that is closest to $\mathbf{x}$. In the diagram below, $\mathbf{p}$ is the projection of $\mathbf{x}$ onto L.



If $s\mathbf{a}$ is a point on the line L then the distance from $s\mathbf{a}$ to $\mathbf{x}$ is $\|s\mathbf{a}-\mathbf{x}\|$. To compute the projection we must find s that minimizes $\|s\mathbf{a}-\mathbf{x}\|$. This is the same as minmizing the square $\|s\mathbf{a}-\mathbf{x}\|^2$. Now

$$\begin{aligned}\|s\mathbf{a}-\mathbf{x}\|^2 &= (s\mathbf{a}-\mathbf{x})\cdot(s\mathbf{a}-\mathbf{x})\\ &= s^2\|\mathbf{a}\|^2 - 2s\mathbf{a}\cdot\mathbf{x} + \|\mathbf{x}\|^2.\end{aligned}$$

To minimize this quantity, we can use elementary calculus: differentiate with respect to s, set the derivative equal to zero and solve for s. This yields

$$s = \frac{\mathbf{a}\cdot\mathbf{x}}{\|\mathbf{a}\|^2} = \frac{1}{\|\mathbf{a}\|^2}\mathbf{a}^T\mathbf{x}.$$

and therefore the projection is given by

$$\mathbf{p} = \frac{1}{\|\mathbf{a}\|^2}(\mathbf{a}^T\mathbf{x})\mathbf{a}$$

It is useful to rewrite this formula. The product $(\mathbf{a}^T\mathbf{x})\mathbf{a}$ of the scalar $(\mathbf{a}^T\mathbf{x})$ times the vector $\mathbf{a}$ can also be written as a matrix product $\mathbf{a}\mathbf{a}^T\mathbf{x}$, if we consider $\mathbf{x}$ and $\mathbf{a}$ to be 3×1 matrices. Thus

$$\mathbf{p} = \frac{1}{\|\mathbf{a}\|^2}\mathbf{a}\mathbf{a}^T\mathbf{x}.$$

This formula says that the projection of $\mathbf{x}$ onto the line throught $\mathbf{a}$ can be obtained by multiplying $\mathbf{x}$ by the 3×3 matrix P given by

$$P = \frac{1}{\|\mathbf{a}\|^2}\mathbf{a}\mathbf{a}^T.$$

We now make some observations about the matrix P.

To begin, we observe that the matrix P satisfies the equation $P^2 = P$. To see why this must be true, notice that $P^2\mathbf{x} = P(P\mathbf{x})$ is the vector in L closest to $P\mathbf{x}$. But $P\mathbf{x}$ is already in L so the closest vector to it in L is $P\mathbf{x}$ itself. Thus $P^2\mathbf{x} = P\mathbf{x}$, and since this is true for every $\mathbf{x}$ it must be true that $P^2 = P$. We can also verify the equation $P^2 = P$ directly by the calculation

$$P^2 = \frac{1}{\|\mathbf{a}\|^4}(\mathbf{a}\mathbf{a}^T)(\mathbf{a}\mathbf{a}^T) = \frac{1}{\|\mathbf{a}\|^4}\mathbf{a}(\mathbf{a}^T\mathbf{a})\mathbf{a}^T = \frac{\|\mathbf{a}\|^2}{\|\mathbf{a}\|^4}\mathbf{a}\mathbf{a}^T = \frac{1}{\|\mathbf{a}\|^2}\mathbf{a}\mathbf{a}^T = P$$

(here we used that matrix multiplication is associative and $\mathbf{a}^T\mathbf{a} = \|\mathbf{a}\|^2$).

Another fact about P is that it is equal to its transpose, that is, $P^T = P$. This can also be verified directly by the calculation

$$P^T = \frac{1}{\|\mathbf{a}\|^2}(\mathbf{a}\mathbf{a}^T)^T = \frac{1}{\|\mathbf{a}\|^2}(\mathbf{a}^T)^T\mathbf{a}^T = \frac{1}{\|\mathbf{a}\|^2}\mathbf{a}\mathbf{a}^T = P.$$

(here we use that $(AB)^T = B^TA^T$ and $(A^T)^T = A$).

Clearly, the range of P is

$$R(P) = L.$$

The equation $P^T = P$ lets us determine the null space too. Using one of the orthogonality relations for the four fundamental subspaces of P we find that

$$N(P) = R(P^T)^\perp = R(P)^\perp = L^\perp.$$

Example: Compute the matrix P that projects onto the line L through $\mathbf{a} = \begin{bmatrix} 1 \\ 2 \\ -1 \end{bmatrix}$. Verify that $P^2 = P$ and $P^T = P$. What vector in L is closest to $\mathbf{x} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$?

Let's use MATLAB/Octave to do this calculation.

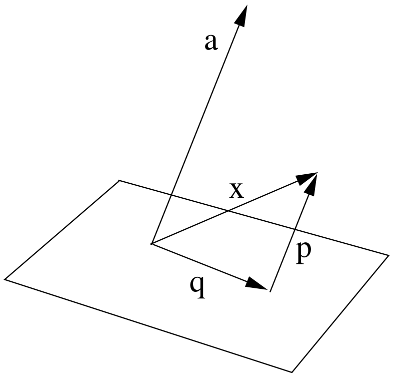
```
>x = [1 1 1]';
>a = [1 2 -1]';
>P = (a'*a)^(-1)*a*a'

P =

   0.16667   0.33333  -0.16667
   0.33333   0.66667  -0.33333
  -0.16667  -0.33333   0.16667

>P*P
```

```
ans =

   0.16667   0.33333  -0.16667
   0.33333   0.66667  -0.33333
  -0.16667  -0.33333   0.16667
```

This verifies the equation $P^2 = P$. The fact that $P^T = P$ can be seen by inspection. The vector in L closest to $\mathbf{x}$ is given by

```
>P*x

ans =

   0.33333
   0.66667
  -0.33333
```

Now we consider plane $L^\perp$ orthogonal to L. Given a vector $\mathbf{x}$, how can we find the projection of $\mathbf{x}$ onto $L^\perp$, that is, the the vector $\mathbf{q}$ in $L^\perp$ closest to $\mathbf{x}$? Looking at the picture,



we can guess that $\mathbf{q} = \mathbf{x} - \mathbf{p}$, where $\mathbf{p} = P\mathbf{x}$ is the projection of $\mathbf{x}$ onto L. This would say that $\mathbf{q} = \mathbf{x} - P\mathbf{x} = (I - P)\mathbf{x}$, where I denotes the identity matrix. In other words, $Q = I - P$ is the matrix that projects on $L^\perp$. We will see below that this guess is correct.

Example: Compute the vector $\mathbf{q}$ in the plane orthogonal to $\mathbf{a} = \begin{bmatrix} 1 \\ 2 \\ -1 \end{bmatrix}$ that is closest to $\mathbf{x} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$.

As in the previous example, let's use MATLAB/Octave. Assume that `a`, `x` and `P` have been defined in the previous example. The 3×3 identity matrix is computed using the command `eye(3)`. If we compute

```
>Q=eye(3)-P
Q =

   0.83333  -0.33333   0.16667
  -0.33333   0.33333   0.33333
   0.16667   0.33333   0.83333
```

then the vector we are seeking is

```
> Q*x
ans =

   0.66667
   0.33333
   1.33333
```

III.1.2. Orthogonal projection matrices

A matrix P is called an *orthogonal projection matrix* if

- $P^2 = P$
- $P^T = P$.

The matrix $\frac{1}{\|\mathbf{a}\|^2}\mathbf{a}\mathbf{a}^T$ defined in the last section is an example of an orthogonal projection matrix. This matrix projects onto its range, which is one dimensional and equal to the span of $\mathbf{a}$. We will see below that every orthogonal projection matrix projects onto its range, but the range can have any dimension.

So, let P be an orthogonal projection matrix, and let $Q = I - P$. Then

1. Q is also an orthogonal projection matrix.
2. $P+Q = I$ and $PQ = QP = \mathbf{0}$. (A consequence of this is that any vector in $R(P)$ is orthogonal to any vector in $R(Q)$ since $(P\mathbf{x}) \cdot (Q\mathbf{y}) = \mathbf{x} \cdot (P^TQ\mathbf{y}) = \mathbf{x} \cdot (PQ\mathbf{y}) = 0$.)
3. P projects onto its range $R(P)$. (In other words, $P\mathbf{x}$ is the closest vector in $R(P)$ to $\mathbf{x}$.)
4. Q projects onto $N(P) = R(P)^{\perp}$.

Let's verify these statements in order:

1. This follows from $Q^2 = (I-P)(I-P) = I - 2P + P^2 = I - 2P + P = I - P = Q$ and $(I-P)^T = I^T - P^T = I - P$

2. The identity $P + Q = I$ follows immediately from the definition of Q. The second identity follows from $PQ = P(I - P) = P - P^2 = P - P = \mathbf{0}$. The identity $QP = \mathbf{0}$ has a similar proof.

3. We want to find the closest vector in $R(P)$ (i.e., of the form $P\mathbf{y}$ for some $\mathbf{y}$) to $\mathbf{x}$. To do this we must find the vector $\mathbf{y}$ that minimizes $\|P\mathbf{y} - \mathbf{x}\|^2$. We have

$$\begin{aligned} \|P\mathbf{y} - \mathbf{x}\|^2 &= \|P(\mathbf{y} - \mathbf{x}) - Q\mathbf{x}\|^2 \quad \text{(using } \mathbf{x} = P\mathbf{x} + Q\mathbf{x}) \\ &= (P(\mathbf{y} - \mathbf{x}) - Q\mathbf{x}) \cdot (P(\mathbf{y} - \mathbf{x}) - Q\mathbf{x}) \\ &= \|P(\mathbf{y} - \mathbf{x})\|^2 + \|Q\mathbf{x}\|^2 \quad \text{(the cross terms vanish by 2.)} \end{aligned}$$

   This is obviously minimized when $\mathbf{y} = \mathbf{x}$. Thus $P\mathbf{x}$ is the closest vector in $R(P)$ to $\mathbf{x}$.

4. Since Q is an orthogonal projection (by 1.) we know it projects onto $R(Q)$ (by 3.). Since we know that $R(P)^\perp = N(P)$ (from the basic subspace relation $N(P) = R(P^T)^\perp$ and the fact that $P^T = P$) it remains to show that $R(Q) = N(P)$. First, note that $\mathbf{x} \in R(Q) \Leftrightarrow Q\mathbf{x} = \mathbf{x}$. (The implication $\Leftarrow$ is obvious, while the implication $\Rightarrow$ can be seen as follows. Suppose $\mathbf{x} \in R(Q)$. This means $\mathbf{x} = Q\mathbf{y}$ for some $\mathbf{y}$. Then $Q\mathbf{x} = Q^2\mathbf{y} = Q\mathbf{y} = \mathbf{x}$.) Now we can complete the argument: $\mathbf{x} \in R(Q) \Leftrightarrow Q\mathbf{x} = \mathbf{x} \Leftrightarrow (I - P)\mathbf{x} = \mathbf{x} \Leftrightarrow P\mathbf{x} = \mathbf{0} \Leftrightarrow \mathbf{x} \in N(P)$.

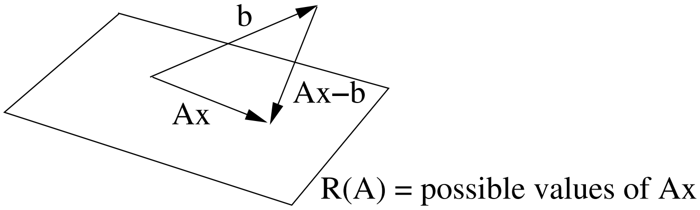
This section has been rather theoretical. We have shown that an orthogonal projection matrix projects onto its range. But suppose that a subspace is presented as $\text{span}(\mathbf{a}_1, \ldots, \mathbf{a}_k)$ (or equivalently $R([\mathbf{a}_1|\cdots|\mathbf{a}_k])$ for a given collection of vectors $\mathbf{a}_1, \ldots, \mathbf{a}_k$. How can we compute the projection matrix P whose range is this given subspace, so that P projects onto it? We will answer this question in the next section.

III.1.3. Least squares and the projection onto $R(A)$

We now consider linear equations

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

That do not have a solution. This is the same as saying that $\mathbf{b} \notin R(A)$ What vector $\mathbf{x}$ is closest to being a solution?



We want to determine $\mathbf{x}$ so that $A\mathbf{x}$ is as close as possible to $\mathbf{b}$. In other words, we want to minimize $\|A\mathbf{x} - \mathbf{b}\|$. This will happen when $A\mathbf{x}$ is the projection of $\mathbf{b}$ onto $R(A)$, that is, $A\mathbf{x} = P\mathbf{b}$, where P is the projection matrix. In this case $Q\mathbf{b} = (I - P)\mathbf{b}$ is orthogonal to $R(A)$. But

$(I-P)\mathbf{b} = \mathbf{b} - A\mathbf{x}$. Therefore (and this is also clear from the picture), we see that $A\mathbf{x} - \mathbf{b}$ is orthogonal to $R(A)$. But the vectors orthogonal to $R(A)$ are exactly the vectors in $N(A^T)$. Thus the vector we are looking for will satisfy $A^T(A\mathbf{x} - \mathbf{b}) = \mathbf{0}$ or the equation

$$A^T A\mathbf{x} = A^T\mathbf{b}$$

This is the least squares equation, and a solution to this equation is called a least squares solution.

(Aside: We can also use Calculus to derive the least squares equation. We want to minimize $\|A\mathbf{x} - \mathbf{b}\|^2$. Computing the gradient and setting it to zero results in the same equations.)

It turns out that the least squares equation always has a solution. Another way of saying this is $R(A^T) = R(A^T A)$. Instead of checking this, we can verify that the orthogonal complements $N(A)$ and $N(A^T A)$ are the same. But this is something we showed before, when we considered the incidence matrix D for a graph.

If $\mathbf{x}$ solves the least squares equation, the vector $A\mathbf{x}$ is the projection of $\mathbf{b}$ onto the range $R(A)$, since $A\mathbf{x}$ is the closest vector to $\mathbf{x}$ in the range of A. In the case where $A^T A$ is invertible (this happens when $N(A) = N(A^T A) = \{\mathbf{0}\}$), we can obtain a formula for the projection. Starting with the least squares equation we multiply by $(A^T A)^{-1}$ to obtain

$$\mathbf{x} = (A^T A)^{-1} A^T\mathbf{b}$$

so that

$$A\mathbf{x} = A(A^T A)^{-1} A^T\mathbf{b}.$$

Thus the projection matrix is given by

$$P = A(A^T A)^{-1} A^T$$

Notice that the formula for the projection onto a line through $\mathbf{a}$ is a special case of this, since then $A^T A = \|\mathbf{a}\|^2$.

It is worthwhile pointing out that if we say that the solution of the least squares equation gives the "best" approximation to a solution, what we really mean is that it minimizes the distance, or equivalently, its square

$$\|A\mathbf{x} - \mathbf{b}\|^2 = \sum((A\mathbf{x})_i - \mathbf{b}_i)^2.$$

There are other ways of measuring how far $A\mathbf{x}$ is from $\mathbf{b}$, for example the so-called L^1 norm

$$\|A\mathbf{x} - \mathbf{b}\|_1 = \sum |(A\mathbf{x})_i - \mathbf{b}_i|$$

Minimizing the L^1 norm will result in a different "best" solution, that may be preferable under some circumstances. However, it is much more difficult to find!

III.1.4. Polynomial fit

Suppose we have some data points $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ and we want to fit a polynomial $p(x) = a_1 x^{m-1} + a_2 x^{m-2} + \cdots + a_{m-1}x + a_m$ through them. This is like the Lagrange interpolation problem we considered before, except that now we assume that $n > m$. This means that in general there will no such polynomial. However we can look for the least squares solution.

To begin, let's write down the equations that express the desired equalities $p(x_i) = y_i$ for $i = 1, \ldots m$. These can be written in matrix form

$$\begin{bmatrix} x_1^{m-1} & x_1^{m-2} & \cdots & x_1 & 1 \\ x_2^{m-1} & x_2^{m-2} & \cdots & x_2 & 1 \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ x_n^{m-1} & x_n^{m-2} & \cdots & x_n & 1 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ \vdots \\ \vdots \\ y_n \end{bmatrix}$$

or $A\mathbf{a} = \mathbf{y}$. where A is a submatrix of the Vandermonde matrix. To find the least squares approximation we solve $A^T A\mathbf{a} = A^T\mathbf{y}$. In a homework problem, you are asked to do this using MATLAB/Octave.

In case where the polynomial has degree one this is a straight line fit, and the equation we want to solve are

$$\begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \\ x_n & 1 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

These equations will not have a solution (unless the points really do happen to lie on the same line.) To find the least squares solution, we compute

$$\begin{bmatrix} x_1 & x_2 & \cdots & x_n \\ 1 & 1 & \cdots & 1 \end{bmatrix} \begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \\ x_n & 1 \end{bmatrix} = \begin{bmatrix} \sum x_i^2 & \sum x_i \\ \sum x_i & n \end{bmatrix}$$

and

$$\begin{bmatrix} x_1 & x_2 & \cdots & x_n \\ 1 & 1 & \cdots & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} \sum x_i y_i \\ \sum y_i \end{bmatrix}$$

This results in the familiar equations

$$\begin{bmatrix} \sum x_i^2 & \sum x_i \\ \sum x_i & n \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} \sum x_i y_i \\ \sum y_i \end{bmatrix}$$

which are easily solved.

III.1.5. Football rankings

We can try to use least squares to rank football teams. To start with, suppose we have three teams. We pretend each team has a value v_1, v_2 and v_3 such that when two teams play, the difference in scores is the difference in values. So, if the season's games had the following results

1 vs. 2	30	40
1 vs. 2	20	40
2 vs. 3	10	0
3 vs. 1	5	0
3 vs. 2	5	5

then the v_i's would satisfy the equations

$$\begin{aligned}
v_2 - v_1 &= 10\\
v_2 - v_1 &= 20\\
v_2 - v_3 &= 10\\
v_3 - v_1 &= 5\\
v_2 - v_3 &= 0
\end{aligned}$$

Of course, there is no solution to these equations. Nevertheless we can find the least squares solution. The matrix form of the equations is

$$D\mathbf{v} = \mathbf{b}$$

with

$$D = \begin{bmatrix} -1 & 1 & 0\\ -1 & 1 & 0\\ 0 & 1 & -1\\ -1 & 0 & 1\\ 0 & 1 & -1 \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} 10\\ 20\\ 10\\ 5\\ 0 \end{bmatrix}$$

The least squares equation is

$$D^T D\mathbf{v} = D^T\mathbf{b}$$

or

$$\begin{bmatrix} 3 & -2 & -1\\ -2 & 4 & -2\\ -1 & -2 & 3 \end{bmatrix}\mathbf{v} = \begin{bmatrix} -35\\ 40\\ -5 \end{bmatrix}$$

Before going on, notice that D is an incidence matrix. What is the graph? (Answer: the nodes are the teams and they are joined by an edge with the arrow pointing from the losing team to the winning team. This graph may have more than one edge joining to nodes, if two teams play more than once. This is sometimes called a multi-graph.). We saw that in this situation $N(D)$ is not empty, but contains vectors whose entries are all the same. The situation is the same as for resistances, it is only differences in v_i's that have a meaning.

We can solve this equation in MATLAB/Octave. The straightforward way is to compute

```
>L = [3 -2 -1;-2 4 -2;-1 -2 3];
>b = [-35; 40; -5];
>rref([L b])

ans =

   1.00000   0.00000  -1.00000  -7.50000
   0.00000   1.00000  -1.00000   6.25000
   0.00000   0.00000   0.00000   0.00000
```

As expected, the solution is not unique. The general solution, depending on the parameter s is

$$\mathbf{v} = s\begin{bmatrix}1\\1\\1\end{bmatrix} + \begin{bmatrix}-7.5\\6.25\\0\end{bmatrix}$$

We can choose s so that the v_i for one of the teams is zero. This is like grounding a node in a circuit. So, by choosing $s = 7.5$, $s = -6.25$ and $s = 0$ we obtain the solutions $\begin{bmatrix}0\\13.75\\7.5\end{bmatrix}$, $\begin{bmatrix}-13.75\\0\\-6.25\end{bmatrix}$ or $\begin{bmatrix}-7.5\\6.25\\0\end{bmatrix}$.

Actually, it is easier to compute a solution with one of the v_i's equal to zero directly. If $\mathbf{v} = \begin{bmatrix}0\\v_2\\v_3\end{bmatrix}$ then $\mathbf{v}_2 = \begin{bmatrix}v_2\\v_3\end{bmatrix}$ satisfies the equation $L_2\mathbf{v}_2 = \mathbf{b}_2$ where the matrix L_2 is the bottom right 2×2 block of L and $\mathbf{b}_2$ are the last two entries of $\mathbf{b}$.

```
>L2 = L(2:3,2:3);
>b2 = b(2:3);
>L2\b2

ans =
   13.7500
    7.5000
```

We can try this on real data. The football scores for the 2007 CFL season can be found at `http://www.cfl.ca/index.php?module=sked&func=view&year=2007`. The differences in scores for the first 20 games are in `cfl.m`. The order of the teams is BC, Calgary, Edmonton, Hamilton, Montreal, Saskatchewan, Toronto, Winnipeg. Repeating the computation above for this data we find the ranking to be (running the file `cfl.m`)

```
v =

    0.00000
  -12.85980
  -17.71983
  -22.01884
  -11.37097
   -1.21812
    0.87588
  -20.36966
```

Not very impressive, if you consider that the second-lowest ranked team (Winnipeg) ended up in the Grey Cup game!

III.2. Complex vector spaces and inner product

Prerequisites and Learning Goals

From your work in previous courses, you should be able to

- Perform arithmetic with complex numbers.
- Write down the definition of and compute the complex conjugate, modulus and argument of a complex number.

After completing this section, you should be able to

- Define and perform basic matrix calculations with complex vectors and complex matrices.
- Define and compute the complex inner product and the norm of complex vectors, state basic properties of the complex inner product.
- Define and compute the matrix adjoint for a complex matrix; explain its relation to the complex inner product; compare its properties to the properties of the transpose of a real matrix.
- Define an orthonormal basis for $\mathbb{C}^n$ and determine whether a set of complex vectors is an orthonormal basis; determine the coefficients in the expansion of a complex vector in an orthonormal basis.
- Write down the definition of a unitary matrix and list its properties; recognize when a matrix is unitary.
- Define and compute the inner product and the norm for complex- (or real-) valued functions that are defined on a given interval; define what it means for two functions to be orthonormal, and verify it in specific examples.
- Define the complex exponential function, compute its value at given points and perform basic computations (addition, differentiation, integration) involving complex exponential functions.
- Explain what are the elements of the vector space $L^2([a, b])$ for an interval $[a, b]$.
- Use complex numbers in MATLAB/Octave computations, specifically `real(z)`, `imag(z)`, `conj(z)`, `abs(z)`, `exp(z)` and `A’` for complex matrices.

III.2.1. Why use complex numbers?

So far the numbers (or scalars) we have been using have been real numbers. Now we will to start using complex numbers as well. Here are two reasons why.

1. Solving polynomial equations (finding roots, factoring): If we use complex numbers, then every polynomial

$$p(z) = a_1 z^{n-1} + a_2 z^{n-2} + \cdots a_{n-1} z + a_n$$

with $a_1 \neq 0$ (so that $p(z)$ has degree $n-1$) can be completely factored as

$$p(z) = a_1(z - r_1)(z - r_2)\cdots(z - r_{n-1}).$$

The numbers $r_1, \ldots, r_{n-1}$ are called the roots of $p(z)$ and are the values of z for which $p(z) = 0$. Thus the equation $p(z) = 0$ always has solutions. There might not be $n-1$ distinct solutions, though, since it may happen that a given root r occurs more than once. If r occurs k times in the list, then we say r has multiplicty k. An important point is that the roots of a polynomial may be complex even when the coefficients $a_1, \ldots, a_n$ are real. For example $z^2 + 1 = (z + i)(z - i)$.

2. Complex exponential: The complex exponential function $e^{i\theta}$ is more convenient to use than $\cos(\theta)$ and $\sin(\theta)$ because it is easier to multiply, differentiate and integrate exponentials than trig functions.

Solving polynomial equations will be important when studying eigenvalues, while the complex exponential appears in Fourier series and the discrete Fourier transform.

III.2.2. Review of complex numbers

Complex numbers can be thought of as points on the (x, y) plane. The point $\begin{bmatrix} x \\ y \end{bmatrix}$, thought of as a complex number, is written $x + iy$ (or $x + jy$ if you are an electrical engineer).

If $z = x+iy$ then x is called the real part of z and is denoted $\mathrm{Re}(z)$ while y is called the imaginary part of z and is denoted $\mathrm{Im}(z)$.

Complex numbers are added just like vectors in two dimensions. If $z = x + iy$ and $w = s + it$, then

$$z + w = (x + iy) + (s + it) = (x + s) + i(y + t)$$

The rule for multiplying two complex numbers is

$$zw = (x + iy)(s + it) = (xs - yt) + i(xt + ys)$$

Notice that i is a square root of -1 since

$$i^2 = (0 + i)(0 + i) = (0 - 1) + i(0 + 0) = -1$$

This fact is all you need to remember to recover the rule for multiplying two complex numbers. If you multiply the expressions for two complex numbers formally, and then substitute -1 for i^2 you will get the right answer. For example, to multiply $1 + 2i$ and $2 + 3i$, compute

$$(1 + 2i)(2 + 3i) = 2 + 3i + 4i + 6i^2 = 2 - 6 + i(3 + 4) = -4 + 7i$$

Complex addition and multiplication obey the usual rules of algebra:

$$\begin{array}{cc} z_1 + z_2 = z_2 + z_1 & z_1 z_2 = z_2 z_1 \\ z_1 + (z_2 + z_3) = (z_1 + z_2) + z_3 & z_1(z_2 z_3) = (z_1 z_2) z_3 \\ 0 + z_1 = z_1 & 1 z_1 = z_1 \\ \multicolumn{2}{c}{z_1(z_2 + z_3) = z_1 z_2 + z_1 z_3} \end{array}$$

The negative of any complex number $z = x + iy$ is defined by $-z = -x + (-y)i$, and obeys $z + (-z) = 0$.

The modulus of a complex number, denoted $|z|$, is the length of the corresponding vector in two dimensions. If $z = x + iy$

$$|z| = |x + iy| = \sqrt{x^2 + y^2}$$

An important property is

$$|zw| = |z||w|$$

The complex conjugate of a complex number z, denoted $\bar{z}$, is the reflection of z across the x axis. Thus

$$\overline{x + iy} = x - iy.$$

The complex conjugate obeys

$$\overline{z + w} = \bar{z} + \bar{w}, \quad \overline{zw} = \bar{z}\bar{w}$$

This means that complex conjugate of an algebraic expression can be obtained by changing all the i's to $-i$'s, either before or after performing arithmetic operations. The complex conjugate also obeys

$$z\bar{z} = |z|^2.$$

This last equality is useful for simplifying fractions of complex numbers by turning the denominator into a real number, since

$$\frac{z}{w} = \frac{z\bar{w}}{|w|^2}$$

For example, to simplify $(1 + i)/(1 - i)$ we can write

$$\frac{1 + i}{1 - i} = \frac{(1 + i)^2}{(1 - i)(1 + i)} = \frac{1 - 1 + 2i}{2} = i$$

A complex number z is real (i.e. the y part in $x + iy$ is zero) whenever $\bar{z} = z$. We also have the following formulas for the real part and imaginary part of z. If $z = x+iy$ then $\mathrm{Re}(z) = x = (z+\bar{z})/2$ and $\mathrm{Im}(z) = y = (z - \bar{z})/(2i)$

We define the exponential, e^{it}, of a purely imaginary number it to be the number

$$e^{it} = \cos(t) + i\sin(t)$$

lying on the unit circle in the complex plane.

The complex exponential satisfies the familiar rule $e^{i(s+t)} = e^{is}e^{it}$ since by the addition formulas for sine and cosine

$$\begin{aligned}
e^{i(s+t)} &= \cos(s+t) + i\sin(s+t) \\
&= \cos(s)\cos(t) - \sin(s)\sin(t) + i(\sin(s)\cos(t) + \cos(s)\sin(t)) \\
&= (\cos(s) + i\sin(s))(\cos(t) + i\sin(t)) \\
&= e^{is}e^{it}
\end{aligned}$$

Any complex number can be written in polar form

$$z = re^{i\theta}$$

where r and θ are the polar co-ordinates of z. This means $r = |z|$ and θ is the angle that the line joining z to 0 makes with the real axis. The angle θ is called the *argument* of z, denoted $\arg(z)$. Since $e^{i(\theta+2\pi k)} = e^{i\theta}$ for $k \in \mathbb{Z}$ the argument is only defined up to an integer multiple of 2π. In other words, there are infinitely many choices for $\arg(z)$. We can always choose a value of the argument with $-\pi < \theta \le \pi$; this choice is called the principal value of the argument.

The polar form lets us understand the geometry of the multiplication of complex numbers. If $z_1 = r_1e^{i\theta_1}$ and $z_2 = r_2e^{i\theta_2}$ then

$$z_1z_2 = r_1r_2e^{i(\theta_1+\theta_2)}$$

This shows that when we multiply two complex numbers, their arguments are added.

The exponential of a number that has both a real and imaginary part is defined in the natural way.

$$e^{a+ib} = e^ae^{ib} = e^a(\cos(b) + i\sin(b))$$

The derivative of a complex exponential is given by the formula

$$\frac{d}{dt}e^{(a+ib)t} = (a+ib)e^{(a+ib)t}$$

while the anti-derivative, for $(a+ib) \neq 0$ is

$$\int e^{(a+ib)t}dt = \frac{1}{(a+ib)}e^{(a+ib)t} + C$$

If $(a+ib) = 0$ then $e^{(a+ib)t} = e^0 = 1$ so in this case

$$\int e^{(a+ib)t}dt = \int dt = t + C$$

III.2.3. Complex vector spaces and inner product

The basic example of a complex vector space is the space $\mathbb{C}^n$ of n-tuples of complex numbers. Vector addition and scalar multiplication are defined as before:

$$\begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{bmatrix} + \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} = \begin{bmatrix} z_1 + w_1 \\ z_2 + w_2 \\ \vdots \\ z_n + w_n \end{bmatrix}, \quad s \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{bmatrix} = \begin{bmatrix} sz_1 \\ sz_2 \\ \vdots \\ sz_n \end{bmatrix},$$

where now z_i, w_i and s are complex numbers.

For complex matrices (or vectors) we define the complex conjugate matrix (or vector) by conjugating each entry. Thus, if $A = [a_{i,j}]$, then

$$\overline{A} = [\overline{a}_{i,j}].$$

The product rule for complex conjugation extends to matrices and we have

$$\overline{AB} = \bar{A}\bar{B}$$

The inner product of two complex vectors $\mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix}$ and $\mathbf{z} = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{bmatrix}$ is defined by

$$\langle \mathbf{w}, \mathbf{z} \rangle = \overline{\mathbf{w}}^T \mathbf{z} = \sum_{i=1}^{n} \overline{w}_i z_i$$

When the entries of $\mathbf{w}$ and $\mathbf{z}$ are all real, then this is just the usual dot product. (In these notes we will reserve the notation $\mathbf{w} \cdot \mathbf{z}$ for the case when $\mathbf{w}$ and $\mathbf{z}$ are real.) When the vectors are complex it is important to remember the complex conjugate in this definition. Notice that for complex vectors the order of $\mathbf{w}$ and $\mathbf{z}$ in the inner product matters: we have $\langle \mathbf{z}, \mathbf{w} \rangle = \overline{\langle \mathbf{w}, \mathbf{z} \rangle}$.

With this definition for the inner product the norm of $\mathbf{z}$ is always positive since

$$\langle \mathbf{z}, \mathbf{z} \rangle = \|\mathbf{z}\|^2 = \sum_{i=1}^{n} |z_i|^2$$

For complex matrices and vectors we have to modify the rule for bringing a matrix to the other side of an inner product.

$$
\begin{aligned}
\langle \mathbf{w}, A\mathbf{z} \rangle &= \overline{\mathbf{w}}^T A\mathbf{z} \\
&= (A^T \overline{\mathbf{w}})^T \mathbf{z} \\
&= \left(\overline{(\overline{A}^T \mathbf{w})} \right)^T \mathbf{z} \\
&= \langle \overline{A}^T \mathbf{w}, \mathbf{z} \rangle
\end{aligned}
$$

This leads to the definition of the adjoint of a matrix

$$A^* = \overline{A}^T.$$

(In physics you will also see the notation $A^\dagger$.) With this notation

$$\langle \mathbf{w}, A\mathbf{z} \rangle = \langle A^* \mathbf{w}, \mathbf{z} \rangle.$$

MATLAB/Octave deals seamlessly with complex matrices and vectors. Complex numbers can be entered like this

```
>z= 1 + 2i

z =  1 + 2i
```

There is a slight danger here in that if `i` has be defined to be something else (*e.g.* `i =16`) then `z=i` would set `z` to be `16`. In this case, if you do want `z` to be equal to the number $0 + i$, you could use `z=1i` to get the desired result, or use the alternative syntax

```
>z= complex(0,1)

z =  0 + 1i
```

The functions `real(z)`, `imag(z)`, `conj(z)`, `abs(z)` compute the real part, imaginary part, conjugate and modulus of `z`.

The function `exp(z)` computes the complex exponential if `z` is complex.

If a matrix `A` has complex entries then `A'` is *not* the transpose, but the adjoint (conjugate transpose).

```
>z = [1; 1i]
```

```
z =

   1 + 0i
   0 + 1i


z'

ans =

   1 - 0i   0 - 1i
```

Thus the square of the norm of a complex vector is given by

```
>z'*z

ans =  2
```

This gives the same answer as

```
>norm(z)^2

ans =  2.0000
```

(Warning: the function `dot` in Octave does not compute the correct inner product for complex vectors (it doesn't take the complex conjugate). This has been fixed in the latest versions, so you should check. In MATLAB `dot` works correctly for complex vectors.)

III.3. Orthonormal bases, Orthogonal Matrices and Unitary Matrices

Prerequisites and Learning Goals

After completing this section, you should be able to

- Write down the definition of an orthonormal basis, and determine when a given set of vectors is an orthonormal basis.
- Compute the coefficients in the expansion of a vector in an orthonormal basis.
- Compute the norm of a vector from its coefficients in its expansion in an orthonormal basis.
- Write down the definition of an orthogonal (unitary) matrix; recognize when a matrix is orthogonal (unitary); describe the action of an orthogonal (unitary) matrix on vectors; describe the properties of the rows and columns of an orthogonal (unitary) matrix.

III.3.1. Orthonormal bases

A basis $\mathbf{q}_1, \mathbf{q}_2, \ldots$ is called orthonormal if

1. $\|\mathbf{q}_i\| = 1$ for every i (normal)
2. $\langle \mathbf{q}_i, \mathbf{q}_j \rangle = 0$ for $i \neq j$ (ortho).

The standard basis given by

$$\mathbf{e}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \end{bmatrix}, \quad \mathbf{e}_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \end{bmatrix}, \quad \mathbf{e}_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ \vdots \end{bmatrix}, \quad \cdots$$

is an orthonormal basis for $\mathbb{R}^n$. For example, $\mathbf{e}_1$ and $\mathbf{e}_2$ form an orthonormal basis for $\mathbb{R}^2$. Another orthonormal basis for $\mathbb{R}^2$ is

$$\mathbf{q}_1 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad \mathbf{q}_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

The vectors in a basis for $\mathbb{R}^n$ can also be considered to be vectors in $\mathbb{C}^n$. Any orthonormal basis for $\mathbb{R}^n$ is also an orthonormal basis for $\mathbb{C}^n$ if we are using complex scalars (homework problem). Thus the two examples above are also orthonormal bases for $\mathbb{C}^n$ and $\mathbb{C}^2$ respectively. On the other hand, the basis

$$\mathbf{q}_1 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ i \end{bmatrix}, \quad \mathbf{q}_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -i \end{bmatrix}$$

is an orthonormal basis for $\mathbb{C}^2$ but not for $\mathbb{R}^2$.

If you expand a vector in an orthonormal basis, it's very easy to find the coefficients in the expansion. Suppose

$$\mathbf{v} = c_1\mathbf{q}_1 + c_2\mathbf{q}_2 + \cdots + c_n\mathbf{q}_n$$

for some orthonormal basis $\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_n$. Then, if we take the inner product of both sides with $\mathbf{q}_k$, we get

$$\begin{aligned}\langle \mathbf{q}_k, \mathbf{v}\rangle &= c_1\langle \mathbf{q}_k, \mathbf{q}_1\rangle + c_2\langle \mathbf{q}_k, \mathbf{q}_2\rangle + \cdots + c_k\langle \mathbf{q}_k, \mathbf{q}_k\rangle \cdots + c_n\langle \mathbf{q}_k, \mathbf{q}_n\rangle \\ &= 0 + 0 + \cdots + c_k + \cdots + 0 \\ &= c_k\end{aligned}$$

This gives a convenient formula for each c_k. For example, in the expansion

$$\begin{bmatrix}1\\2\end{bmatrix} = c_1\frac{1}{\sqrt{2}}\begin{bmatrix}1\\1\end{bmatrix} + c_2\frac{1}{\sqrt{2}}\begin{bmatrix}-1\\1\end{bmatrix}$$

we have

$$\begin{aligned}c_1 &= \frac{1}{\sqrt{2}}\begin{bmatrix}1\\1\end{bmatrix}\cdot\begin{bmatrix}1\\2\end{bmatrix} = \frac{3}{\sqrt{2}}\\ c_2 &= \frac{1}{\sqrt{2}}\begin{bmatrix}-1\\1\end{bmatrix}\cdot\begin{bmatrix}1\\2\end{bmatrix} = \frac{1}{\sqrt{2}}\end{aligned}$$

Notice also that the norm of $\mathbf{v}$ is easily expressed in terms of the coefficients c_i. We have

$$\begin{aligned}\|\mathbf{v}\|^2 &= \langle \mathbf{v}, \mathbf{v}\rangle \\ &= \langle c_1\mathbf{q}_1 + c_2\mathbf{q}_2 + \cdots + c_n\mathbf{q}_n, c_1\mathbf{q}_1 + c_2\mathbf{q}_2 + \cdots + c_n\mathbf{q}_n\rangle \\ &= |c_1|^2 + |c_2|^2 + \cdots + |c_n|^2\end{aligned}$$

Another way of saying this is that the vector $\mathbf{c} = [c_1, c_2, \ldots, c_n]^T$ of coefficients has the same norm as $\mathbf{v}$.

III.3.2. Orthogonal matrices and Unitary matrices

If we put the vectors of an orthonormal basis into the columns of a matrix, the resulting matrix is called orthogonal (if the vectors are real) or unitary (if the vectors are complex). If $\mathbf{q}_1, \mathbf{q}_2, \ldots \mathbf{q}_n$ is an orthonormal basis then the expansion

$$\mathbf{v} = c_1\mathbf{q}_1 + c_2\mathbf{q}_2 + \cdots + c_n\mathbf{q}_n$$

can be expressed as a matrix equation $\mathbf{v} = Q\mathbf{c}$ where $\mathbf{c} = [c_1, c_2, \ldots, c_n]^T$ and Q is the orthogonal (or unitary) matrix

$$Q = \left[\begin{array}{c|c|c|c} \mathbf{q}_1 & \mathbf{q}_2 & \cdots & \mathbf{q}_n \end{array}\right]$$

The fact that the columns of Q are orthonormal means that $Q^*Q = I$ (equivalently $Q^* = Q^{-1}$). When the entries of Q are real, so that Q is orthogonal, then $Q^* = Q^T$. So for orthogonal matrices $Q^TQ = I$ (equivalently $Q^T = Q^{-1}$).

To see this, we compute

$$\begin{aligned}
Q^*Q &= \left[\begin{array}{c} \overline{\mathbf{q}}_1^T \\ \overline{\mathbf{q}}_2^T \\ \vdots \\ \overline{\mathbf{q}}_n^T \end{array}\right] \left[\begin{array}{c|c|c|c} \mathbf{q}_1 & \mathbf{q}_2 & \cdots & \mathbf{q}_n \end{array}\right] \\
&= \begin{bmatrix} \langle \mathbf{q}_1, \mathbf{q}_1 \rangle & \langle \mathbf{q}_1, \mathbf{q}_2 \rangle & \cdots & \langle \mathbf{q}_1, \mathbf{q}_n \rangle \\ \langle \mathbf{q}_2, \mathbf{q}_1 \rangle & \langle \mathbf{q}_2, \mathbf{q}_2 \rangle & \cdots & \langle \mathbf{q}_2, \mathbf{q}_n \rangle \\ \vdots & \vdots & \vdots & \vdots \\ \langle \mathbf{q}_n, \mathbf{q}_1 \rangle & \langle \mathbf{q}_n, \mathbf{q}_2 \rangle & \cdots & \langle \mathbf{q}_n, \mathbf{q}_n \rangle \end{bmatrix} \\
&= \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}.
\end{aligned}$$

Another way of recognizing unitary and orthogonal matrices is by their action on vectors. Suppose Q is unitary. We already observed in the previous section that if $\mathbf{v} = Q\mathbf{c}$ then $\|\mathbf{v}\| = \|\mathbf{c}\|$. We can also see this directly from the calculation

$$\|Q\mathbf{v}\|^2 = \langle Q\mathbf{v}, Q\mathbf{v} \rangle = \langle \mathbf{v}, Q^*Q\mathbf{v} \rangle = \langle \mathbf{v}, \mathbf{v} \rangle = \|\mathbf{v}\|^2$$

This implies that $\|Q\mathbf{v}\| = \|\mathbf{v}\|$. In other words, unitary matrices don't change the lengths of vectors.

The converse is also true. If a matrix Q doesn't change the lengths of vectors then it must be unitary (or orthogonal, if the entries are real). We can show this using the following identity, called the polarization identity, that expresses the inner product of two vectors in terms of norms.

$$\langle \mathbf{v}, \mathbf{w} \rangle = \frac{1}{4}\Big(\|\mathbf{v}+\mathbf{w}\|^2 - \|\mathbf{v}-\mathbf{w}\|^2 + i\|\mathbf{v}-i\mathbf{w}\|^2 - i\|\mathbf{v}+i\mathbf{w}\|^2\Big)$$

(You are asked to prove this in a homework problem.) Now suppose that Q doesn't change the length of vectors, that is, $\|Q\mathbf{v}\| = \|\mathbf{v}\|$ for every $\mathbf{v}$. Then, using the polarization identity, we find

$$\begin{aligned}
&\langle Q\mathbf{v}, Q\mathbf{w} \rangle \\
&\quad = \frac{1}{4}\Big(\|Q\mathbf{v}+Q\mathbf{w}\|^2 - \|Q\mathbf{v}-Q\mathbf{w}\|^2 + i\|Q\mathbf{v}-iQ\mathbf{w}\|^2 - i\|Q\mathbf{v}+iQ\mathbf{w}\|^2\Big) \\
&\quad = \frac{1}{4}\Big(\|Q(\mathbf{v}+\mathbf{w})\|^2 - \|Q(\mathbf{v}-\mathbf{w})\|^2 + i\|Q(\mathbf{v}-i\mathbf{w})\|^2 - i\|Q(\mathbf{v}+i\mathbf{w})\|^2\Big) \\
&\quad = \frac{1}{4}\Big(\|\mathbf{v}+\mathbf{w}\|^2 - \|\mathbf{v}-\mathbf{w}\|^2 + i\|\mathbf{v}-i\mathbf{w}\|^2 - i\|\mathbf{v}+i\mathbf{w}\|^2\Big) \\
&\quad = \langle \mathbf{v}, \mathbf{w} \rangle
\end{aligned}$$

Thus $\langle \mathbf{v}, Q^*Q\mathbf{w}\rangle = \langle Q\mathbf{v}, Q\mathbf{w}\rangle = \langle \mathbf{v}, \mathbf{w}\rangle$ for all vectors $\mathbf{v}$ and $\mathbf{w}$. In particular, if $\mathbf{v}$ is the standard basis vector $\mathbf{e}_i$ and $\mathbf{w} = \mathbf{e}_j$, then $\langle \mathbf{e}_i, Q^*Q\mathbf{e}_j\rangle$ is the i, jth entry of the matrix Q^*Q while $\langle \mathbf{e}_i, \mathbf{e}_j\rangle$ is the i, jth entry of the identity matrix I. Since these two quantities are equal for every i and j we may conclude that $Q^*Q = I$. Therefore Q is unitary.

Recall that for square matrices a left inverse is automatically also a right inverse. So if $Q^*Q = I$ then $QQ^* = I$ too. This means that Q^* is an unitary matrix whenever Q is. This proves the (non-obvious) fact that if the columns of an square matrix form an orthonormal basis, then so do the (complex conjugated) rows!

III.4. Fourier series

Prerequisites and Learning Goals

After completing this section, you should be able to

- Show that the functions $e_n(x) = e^{2\pi inx/L}$ for $n = 0, \pm1, \pm2, ...$, $a < x < b$ and $L = b - a$ form an orthonormal (scaled by $\sqrt{L}$) set in $L^2([a, b])$.
- Use the fact that the functions $e_n(x)$ form an infinite orthonormal basis to expand a L^2 function in a Fourier series; explain how this leads to a formula for the coefficients of the series, and compute the coefficients (in real and complex form).
- State and derive Parseval's formula and use it to sum certain infinite series.
- Use MATLAB/Octave to compute and plot the partial sums of Fourier series.
- Explain what an amplitude-frequency plot is and generate it for a given function using MATLAB/Octave; describe the physical interpretation of the plot when the function is a sound wave.

III.4.1. Vector spaces of complex-valued functions

Let $[a, b]$ be an interval on the real line. Recall that we introduced the vector space of real valued functions defined for $x \in [a, b]$. The vector sum $f + g$ of two functions f and g was defined to be the function you get by adding the values, that is, $(f + g)(x) = f(x) + g(x)$ and the scalar multiple sf was defined similarly by $(sf)(x) = sf(x)$.

In exactly the same way, we can introduce a vector space of complex valued functions. The independent variable x is still real, taking values in $[a, b]$. But now the values $f(x)$ of the functions may be complex. Examples of complex valued functions are $f(x) = x + ix^2$ or $f(x) = e^{ix} = \cos(x) + i\sin(x)$.

Now we introduce the inner product of two complex valued functions on $[a, b]$. In analogy with the inner product for complex vectors we define

$$\langle f, g\rangle = \int_a^b \overline{f}(x)g(x)dx$$

and the associated norm defined by

$$\|f\|^2 = \langle f, f\rangle = \int_a^b |f(x)|^2dx$$

For real valued functions we can ignore the complex conjugate.

Example: the inner product of $f(x) = 1 + ix$ and $g(x) = x^2$ over the interval $[0, 1]$ is

$$\langle 1 + ix, x^2 \rangle = \int_0^1 \overline{(1 + ix)} \cdot x^2 dx = \int_0^1 (1 - ix) \cdot x^2 dx = \int_0^1 x^2 - ix^3 dx = \frac{1}{3} - i\frac{1}{4}$$

It will often happen that a function, like $f(x) = x$ is defined for all real values of x. In this case we can consider inner products and norms for any interval $[a, b]$ including semi-infinite and infinite intervals, where a may be $-\infty$ or b may be $+\infty$. Of course the values of the inner product an norm depend on the choice of interval.

There are technical complications when dealing with spaces of functions. In this course we will deal with aspects of the subject where these complications don't play an important role. However, it is good to aware that they exist, so we will mention a few.

One complication is that the integral defining the inner product may not exist. For example for the interval $(-\infty, \infty) = \mathbb{R}$ the norm of $f(x) = x$ is infinite since

$$\int_{-\infty}^{\infty} |x|^2 dx = \infty$$

Even if the interval is finite, like $[0, 1]$, the function might have a spike. For example, if $f(x) = 1/x$ then

$$\int_0^1 \frac{1}{|x|^2} dx = \infty$$

too. To overcome this complication we agree to restrict our attention to square integrable functions. For any interval $[a, b]$, these are the functions $f(x)$ for which $|f(x)|^2$ is integrable. They form a vector space that is usually denoted $L^2([a, b])$. It is an example of a Hilbert space and is important in Quantum Mechanics. The L in this notation indicates that the integrals should be defined as Lebesgue integrals rather than as Riemann integrals usually taught in elementary calculus courses. This plays a role when discussing convergence theorems. But for any functions that come up in this course, the Lebesgue integral and the Riemann integral will be the same.

The question of convergence is another complication that arises in infinite dimensional vector spaces of functions. When discussing infinite orthonormal bases, infinite linear combinations of vectors (functions) will appear. There are several possible meanings for an equation like

$$\sum_{i=0}^{\infty} c_i \phi_i(x) = \phi(x).$$

since we are talking about convergence of an infinite series of functions. The most obvious interpretation is that for every fixed value of x the infinite sum of numbers on the left hand side equals the number on the right.

Here is another interpretation: the difference of ϕ and the partial sums $\sum_{i=0}^{N} c_i \phi_i$ tends to zero when measured in the L^2 norm, that is

$$\lim_{N \to \infty} \| \sum_{i=0}^{N} c_i \phi_i - \phi \| = 0$$

With this definition, it might happen that there are individual values of x where the first equation doesn't hold. This is the meaning that we will give to the equation.

III.4.2. An infinite orthonormal basis for $L^2([a,b])$

Let $[a,b]$ be an interval of length $L = b - a$. For every integer n, define the function

$$e_n(x) = e^{2\pi inx/L}.$$

Then infinite collection of functions

$$\{\ldots, e_{-2}, e_{-1}, e_0, e_1 e_2, \ldots\}$$

forms an orthonormal basis for the space $L^2([a,b])$, except that each function e_n has norm $\sqrt{L}$ instead of 1. (Since this is the usual normalization, we will stick with it. To get a true orthonormal basis, we must divide each function by $\sqrt{L}$.)

Let's verify that these functions form an orthonormal set (scaled by $\sqrt{L}$). To compute the norm we calculate

$$\begin{aligned}\|e_n\|^2 = \langle e_n, e_n\rangle &= \int_a^b \overline{e^{2\pi inx/L}} e^{2\pi inx/L} dx\\ &= \int_a^b e^{-2\pi inx/L} e^{2\pi inx/L} dx\\ &= \int_a^b 1 dx.\\ &= L\end{aligned}$$

This shows that $\|e_n\| = \sqrt{L}$ for every n. Next we check that if $n \neq m$ then e_n and e_m are orthogonal.

$$\begin{aligned}\langle e_n, e_m\rangle &= \int_a^b e^{-2\pi inx/L} e^{2\pi imx/L} dx\\ &= \int_a^b e^{2\pi i(m-n)x/L} dx\\ &= \frac{L}{2\pi i(m-n)} e^{2\pi i(m-n)x/L}\Big|_{x=a}^{b}\\ &= \frac{L}{2\pi i(m-n)}\left(e^{2\pi i(m-n)b/L} - e^{2\pi i(m-n)a/L}\right)\\ &= 0\end{aligned}$$

Here we used that $e^{2\pi i(m-n)b/L} = e^{2\pi i(m-n)(b-a+a)/L} = e^{2\pi i(m-n)} e^{2\pi i(m-n)a/L} = e^{2\pi i(m-n)a/L}$. This shows that the functions $\{\ldots, e_{-2}, e_{-1}, e_0, e_1 e_2, \ldots\}$ form an orthonormal set (scaled by $\sqrt{L}$).

To show these functions form a *basis* we have to verify that they span the space $L^2[a,b]$. In other words, we must show that any function $f \in L^2[a,b]$ can be written as an infinite linear combination

$$f(x) = \sum_{n=-\infty}^{\infty} c_n e_n(x) = \sum_{n=-\infty}^{\infty} c_n e^{2\pi i n x/L}.$$

This is a bit tricky, since it involves infinite series of functions. For a finite dimensional space, to show that an orthogonal set forms a basis, it suffices to count that there are the same number of elements in an orthogonal set as there are dimensions in the space. For an infinite dimensional space this is no longer true. For example, the set of e_n's with n even is also an infinite orthonormal set, but it doesn't span all of $L^2[a,b]$.

In this course, we will simply accept that it is true that $\{\ldots, e_{-2}, e_{-1}, e_0, e_1 e_2, \ldots\}$ span $L^2[a,b]$. Once we accept this fact, it is very easy to compute the coefficients in a Fourier expansion. The procedure is the same as in finite dimensions. Starting with

$$f(x) = \sum_{n=-\infty}^{\infty} c_n e_n(x)$$

we simply take the inner product of both sides with e_m. The only term in the infinite sum that survives is the one with $n = m$. Thus

$$\langle e_m, f \rangle = \sum_{n=-\infty}^{\infty} c_n \langle e_m, e_n \rangle = c_m L$$

and we obtain the formula

$$c_m = \frac{1}{L} \int_a^b e^{-2\pi i m x/L} f(x) dx$$

III.4.3. Real form of the Fourier series

Fourier series are often written in terms of sines and cosines as

$$f(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty} (a_n \cos(2\pi n x/L) + b_n \sin(2\pi n x/L))$$

To obtain this form, recall that

$$e^{\pm 2\pi i n x/L} = \cos(2\pi n x/L) \pm i \sin(2\pi n x/L)$$

Using this formula we find

$$\begin{aligned}
\sum_{n=-\infty}^{\infty} c_n e^{2\pi nx/L} &= c_0 + \sum_{n=1}^{\infty} c_n e^{2\pi nx/L} + \sum_{n=1}^{\infty} c_{-n} e^{-2\pi nx/L} \\
&= c_0 + \sum_{n=1}^{\infty} c_n(\cos(2\pi nx/L) + i\sin(2\pi nx/L)) \\
&\qquad\qquad + \sum_{n=1}^{\infty} c_{-n}(\cos(2\pi nx/L) - i\sin(2\pi nx/L)) \\
&= c_0 + \sum_{n=1}^{\infty} \left((c_n + c_{-n})\cos(2\pi nx/L) + i(c_n - c_{-n})\sin(2\pi nx/L)\right)
\end{aligned}$$

Thus the real form of the Fourier series holds with

$$\begin{aligned}
a_0 &= 2c_0 \\
a_n &= c_n + c_{-n} \quad \text{for} \quad n > 0 \\
b_n &= ic_n - ic_{-n} \quad \text{for} \quad n > 0.
\end{aligned}$$

Equivalently

$$\begin{aligned}
c_0 &= \frac{a_0}{2} \\
c_n &= \frac{a_n}{2} + \frac{b_n}{2i} \quad \text{for} \quad n > 0 \\
c_n &= \frac{a_{-n}}{2} - \frac{b_{-n}}{2i} \quad \text{for} \quad n < 0.
\end{aligned}$$

The coefficients a_n and b_n in the real form of the Fourier series can also be obtained directly. The set of functions

$$\{1/2, \cos(2\pi x/L), \cos(4\pi x/L), \cos(6\pi x/L), \ldots, \sin(2\pi x/L), \sin(4\pi x/L), \sin(6\pi x/L), \ldots\}$$

also form an orthogonal basis where each vector has norm $\sqrt{L/2}$. This leads to the formulas

$$a_n = \frac{2}{L}\int_a^b \cos(2\pi nx/L)f(x)$$

for $n = 0, 1, 2, \ldots$ and

$$b_n = \frac{2}{L}\int_a^b \sin(2\pi nx/L)f(x)$$

for $n = 1, 2, \ldots$. The desire to have the formula for a_n work out for $n = 0$ is the reason for dividing by 2 in the constant term $a_0/2$ in the real form of the Fourier series.

One advantage of the real form of the Fourier series is that if $f(x)$ is a real valued function, then the coefficients a_n and b_n are real too, and the Fourier series doesn't involve any complex numbers. However, it is often easier to calculate the coefficients c_n because exponentials are easier to integrate than sines and cosines.

III.4.4. An example

Let's compute the Fourier coefficients for the square wave function. In this example $L = 1$.

$$f(x) = \begin{cases} 1 & \text{if} \quad 0 \le x \le 1/2 \\ -1 & \text{if} \quad 1/2 < x \le 1 \end{cases}$$

If $n = 0$ then $e^{-i2\pi nx} = e^0 = 1$ so c_0 is simply the integral of f.

$$c_0 = \int_0^1 f(x)dx = \int_0^{1/2} 1dx - \int_{1/2}^1 1dx = 0$$

Otherwise, we have

$$\begin{aligned} c_n &= \int_0^1 e^{-i2\pi nx} f(x)dx \\ &= \int_0^{1/2} e^{-i2\pi nx} dx - \int_{1/2}^1 e^{-i2\pi nx} dx \\ &= \frac{e^{-i2\pi nx}}{-i2\pi n}\Big|_{x=0}^{x=1/2} - \frac{e^{-i2\pi nx}}{-i2\pi n}\Big|_{x=1/2}^{x=1} \\ &= \frac{2 - 2e^{i\pi n}}{2\pi i n} \\ &= \begin{cases} 0 & \text{if} \quad n \text{ is even} \\ 2/i\pi n & \text{if} \quad n \text{ is odd} \end{cases} \end{aligned}$$

Thus we conclude that

$$f(x) = \sum_{\substack{n=-\infty \\ n \text{ odd}}}^{\infty} \frac{2}{i\pi n} e^{i2\pi nx}$$

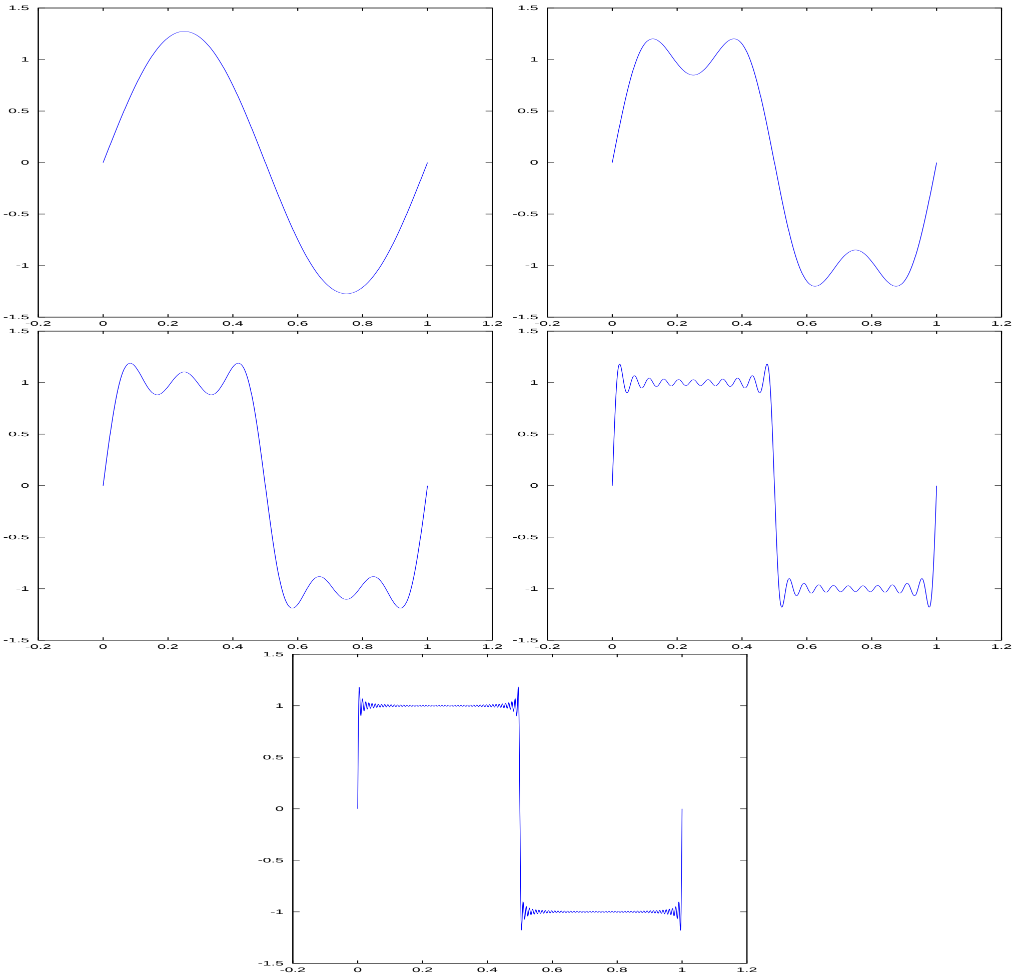
To see how well this series is approximating $f(x)$ we go back to the real form of the series. Using $a_n = c_n + c_{-n}$ and $b_n = ic_n - ic_{-n}$ we find that $a_n = 0$ for all n, $b_n = 0$ for n even and $b_n = 4/\pi n$ for n odd. Thus

$$f(x) = \sum_{\substack{n=1 \\ n \text{ odd}}}^{\infty} \frac{4}{\pi n} \sin(2\pi nx) = \sum_{n=0}^{\infty} \frac{4}{\pi(2n+1)} \sin(2\pi(2n+1)x)$$

We can use MATLAB/Octave to see how well this series is converging. The file `ftdemo1.m` contains a function that take an integer N as an argument and plots the sum of the first $2N + 1$ terms in the Fourier series above. Here is a listing:

```
function ftdemo1(N)

        X=linspace(0,1,1000);
        F=zeros(1,1000);

        for n=[0:N]
                F = F + 4*sin(2*pi*(2*n+1)*X)/(pi*(2*n+1));
        end

        plot(X,F)

end
```

Here are the outputs for $N = 0, 1, 2, 10, 50$:



III.4.5. Parseval's formula

If $\mathbf{v}_1$, $\mathbf{v}_2$, …, $\mathbf{v}_n$ is an orthonormal basis in a finite dimensional vector space and the vector $\mathbf{v}$ has the expansion

$$\mathbf{v} = c_1\mathbf{v}_1 + \cdots + c_n\mathbf{v}_n = \sum_{i=1}^{n} c_i\mathbf{v}_i$$

then, taking the inner product of $\mathbf{v}$ with itself, and using the fact that the basis is orthonormal, we obtain

$$\langle \mathbf{v}, \mathbf{v}\rangle = \sum_{i=1}^{n}\sum_{j=1}^{n} \overline{c}_i c_j \langle \mathbf{v}_i, \mathbf{v}_j\rangle = \sum_{i=1}^{n} |c_i|^2$$

The same formula is true in Hilbert space. If

$$f(x) = \sum_{n=-\infty}^{\infty} c_n e_n(x)$$

Then

$$\int_0^1 |f(x)|^2 dx = \langle f, f\rangle = \sum_{n=-\infty}^{\infty} |c_n|^2$$

In the example above, we have $\langle f, f\rangle = \int_0^1 1dx = 1$ so we obtain

$$1 = \sum_{\substack{n=-\infty \\ n \text{ odd}}}^{n=\infty} \frac{4}{\pi^2 n^2} = 2\sum_{\substack{n=0 \\ n \text{ odd}}}^{n=\infty} \frac{4}{\pi^2 n^2} = \frac{8}{\pi^2}\sum_{n=0}^{\infty} \frac{1}{(2n+1)^2}$$

or

$$\sum_{n=0}^{\infty} \frac{1}{(2n+1)^2} = \frac{\pi^2}{8}$$

III.4.6. Interpretation of Fourier series

What is the meaning of a Fourier series in a practical example? Consider the sound made by a musical instrument in a time interval $[0, T]$. This sound can be represented by a function $y(t)$ for $t \in [0, T]$, where $y(t)$ is the air pressure at a point in space, for example, at your eardrum.

A complex exponential $e^{2\pi i\omega t} = \cos(2\pi\omega t) \pm i\sin(2\pi\omega t)$ can be thought of as a pure oscillation with frequency ω. It is a periodic function whose values are repeated when t increases by ω^{-1}. If t has units of time (seconds) then ω has units of Hertz (cycles per second). In other words, in one second the function $e^{2\pi i\omega t}$ cycles though its values ω times.

The Fourier basis functions can be written as $e^{2\pi i\omega_n t}$ with $\omega_n = n/T$. Thus Fourier's theorem states that for $t \in [0, T]$

$$y(t) = \sum_{n=-\infty}^{\infty} c_n e^{2\pi i\omega_n t}.$$

In other words, the audio signal $y(t)$ can be synthesized as a superposition of pure oscillations with frequencies $\omega_n = n/T$. The coefficients c_n describe how much of the frequency ω_n is present in the signal. More precisely, writing the complex number c_n as $c_n = |c_n|e^{2\pi i\tau_n}$ we have $c_n e^{2\pi i\omega_n t} = |c_n|e^{2\pi i(\omega_n t+\tau_n)}$. Thus $|c_n|$ represents the amplitude of the oscillation with frequency ω_n while τ_n represents a phase shift.

A frequency-amplitude plot for $y(t)$ is a plot of the points $(\omega_n, |c_n|)$. It should be thought of as a graph of the amplitude as a function of frequency and gives a visual representation of how much of each frequency is present in the signal.

If $y(t)$ is defined for all values of t we can use any interval that we want and expand the restriction of $y(t)$ to this interval. Notice that the frequencies $\omega_n = n/T$ in the expansion will be different for different values of T.

Example: Let's illustrate this with the function $y(t) = e^{2\pi it}$ and intervals $[0, T]$. This function is itself a pure oscillation with frequency $\omega = 1$. So at first glance one would expect that there will be only one term in the Fourier expansion. This will turn out to be correct if number 1 is one of the available frequencies, that is, if there is some value of n for which $\omega_n = n/T = 1$. (This happens if T is an integer.) Otherwise, it is still possible to reconstruct $y(t)$, but more frequencies will be required. In this case we would expect that $|c_n|$ should be large for ω_n close to 1. Let's do the calculation. Fix T. Let's first consider the case when T is an integer. Then

$$\begin{aligned} c_n &= \frac{1}{T}\int_0^T e^{-2\pi int/T}e^{2\pi it}dt \\ &= \frac{1}{T}\int_0^T e^{2\pi i(1-n/T)t}dt \\ &= \begin{cases} 1 & n = T \\ \frac{1}{2T\pi i(1-n/T)}\left(e^{2\pi i(T-n)} - e^0\right) = 0 & n \neq T, \end{cases} \end{aligned}$$

as expected. Now let's look at what happens when T is not an integer. Then

$$\begin{aligned} c_n &= \frac{1}{T}\int_0^T e^{-2\pi int/T}e^{2\pi it}dt \\ &= \frac{1}{2\pi i(T-n)}\left(e^{2\pi i(T-n)} - 1\right) \end{aligned}$$

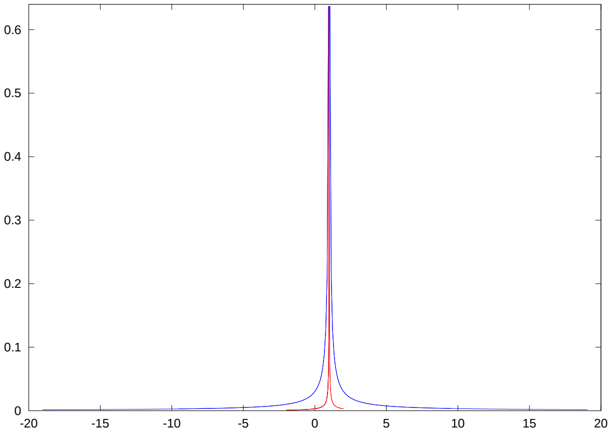
A calculation (that we leave as an exercise) results in

$$|c_n| = \frac{\sqrt{2 - 2\cos(2\pi T(1-\omega_n))}}{2\pi T|1-\omega_n|}$$

We can use MATLAB/Octave to do an amplitude-frequency plot. Here are the commands for $T = 10.5$ and $T = 100.5$

```
N=[-200:200];
T=10.5;
omega=N/T;
abs c=sqrt(2-2*cos(2*pi*T*(1-omega)))./(2*pi*T*abs(1-omega));
plot(omega,absc)
T=100.5;
omega=N/T;
absc=sqrt(2-2*cos(2*pi*T*(1-omega)))./(2*pi*T*abs(1-omega));
hold on;
plot(omega,absc, 'r')
```

Here is the result



As expected, the values of $|c_n|$ are largest when ω_n is close to 1.

Let us return to the sound made by a musical instrument, represented by a function $y(t)$ for $t \in [0, T]$. The frequency content of the sound is captured by the infinite Fourier series and can be displayed using a frequency-amplitude plot. In practical situations, though, we cannot measure $y(t)$ for infinitely many t values, but must sample this functions for a discrete set of t values. How can we perform a frequency analysis with this finite sample? To do this, we will use the discrete Fourier transform, which is the subject of the next section.

III.5. The Discrete Fourier Transform

Prerequisites and Learning Goals

After completing this section, you should be able to

- Explain why the vectors in $\mathbb{C}^n$ obtained by sampling the exponential Fourier bases $e_n(t)$ form an orthogonal basis for $\mathbb{C}^n$ (discrete Fourier bases).
- Use the discrete Fourier basis to expand a vector in $\mathbb{C}^n$ obtaining the discrete Fourier transform of the vector; recognize the matrix that implements the discrete Fourier transform as an unitary matrix.
- Use the Fast Fourier transform (fft) algorithm to compute the discrete Fourier transform, and explain why the Fast Fourier transform algorithm is a faster method. You should be able to perform Fourier transform computations by executing and interpreting the output of the MATLAB/Octave `fft` command.
- Explain the relation between the coefficients in the Fourier series of a function f defined on $[0, L]$ and the coefficients in the discrete Fourier transform of the corresponding sampled values of f, and discuss its limitations.
- Construct a frequency-amplitude plot for a sampled signal using MATLAB/Octave; give a physical interpretation of the resulting plot; explain the relation between this plot and the infinite frequency-amplitude plot.

III.5.1. Definition

In the previous section we saw that the functions $e_k(x) = e^{2\pi ikx}$ for $k \in \mathbb{Z}$ form an infinite orthonormal basis for the Hilbert space of functions $L^2([0,1])$. Now we will introduce a discrete, finite dimensional version of this basis.

To motivate the definition of this basis, imagine taking a function defined on the interval $[0,1]$ and sampling it at the N points $0, 1/N, 2/N, \ldots, j/N, \ldots, (N-1)/N$. If we do this to the basis functions $e_k(x)$ we end up with vectors $\mathbf{e}_k$ given by

$$\mathbf{e}_k = \begin{bmatrix} e^{2\pi i0k/N} \\ e^{2\pi ik/N} \\ e^{2\pi i2k/N} \\ \vdots \\ e^{2\pi i(N-1)k/N} \end{bmatrix} = \begin{bmatrix} 1 \\ \omega_N^k \\ \omega_N^{2k} \\ \vdots \\ \omega_N^{(N-1)k} \end{bmatrix}$$

where

$$\omega_N = e^{2\pi i/N}$$

The complex number ω_N lies on the unit circle, that is, $|\omega_N| = 1$. Moreover ω_N is a primitive Nth root of unity. This means that $\omega_N^N = 1$ and $\omega_N^j \ne 1$ unless j is a multiple of N.

Because $\omega_N^{k+N} = \omega_N^k \omega_N^N = \omega_N^k$ we see that $\mathbf{e}_{k+N} = \mathbf{e}_k$. Thus, although the vectors $\mathbf{e}_k$ are defined for every integer k, they start repeating themselves after N steps. Thus there are only N distinct vectors, $\mathbf{e}_0, \mathbf{e}_1, \ldots, \mathbf{e}_{N-1}$.

These vectors, $\mathbf{e}_k$ for $k = 0, \ldots, N-1$ form an orthogonal basis for $\mathbb{C}^N$. To see this we use the formula for the sum of a geometric series:

$$\sum_{j=0}^{N-1} r^j = \begin{cases} N & r = 1 \\ \dfrac{1 - r^N}{1 - r} & r \ne 1 \end{cases}$$

Using this formula, we compute

$$\langle \mathbf{e}_k, \mathbf{e}_l \rangle = \sum_{j=0}^{N-1} \overline{\omega_N}^{kj} \omega_N^{lj} = \sum_{j=0}^{N-1} \omega_N^{(l-k)j} = \begin{cases} N & l = k \\ \dfrac{1 - \omega_N^{(l-k)N}}{1 - \omega_N^{l-k}} = 0 & l \ne k \end{cases}$$

Now we can expand any vector $\mathbf{f} \in \mathbb{C}^N$ in this basis. Actually, to make our discrete Fourier transform agree with MATLAB/Octave we divide each basis vector by N. Then we obtain

$$\mathbf{f} = \frac{1}{N} \sum_{j=0}^{N-1} c_j \mathbf{e}_j$$

where

$$c_k = \langle \mathbf{e}_k, \mathbf{f} \rangle = \sum_{j=0}^{N-1} e^{-2\pi i k j / N} f_j$$

The map that sends the vector $\mathbf{f}$ to the vector of coefficients $\mathbf{c} = [c_0, \ldots, c_{N-1}]^T$ is the discrete Fourier transform. We can write this in matrix form as

$$\mathbf{c} = F\mathbf{f}, \quad \mathbf{f} = F^{-1}\mathbf{c}$$

where the matrix F^{-1} has the vectors $\mathbf{e}_k$ as its columns. Since this vectors are an orthogonal basis, the inverse is the transpose, up to a factor of N. Explicitly

$$F = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \overline{\omega}_N & \overline{\omega}_N^2 & \cdots & \overline{\omega}_N^{N-1} \\ 1 & \overline{\omega}_N^2 & \overline{\omega}_N^4 & \cdots & \overline{\omega}_N^{2(N-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & \overline{\omega}_N^{N-1} & \overline{\omega}_N^{2(N-1)} & \cdots & \overline{\omega}_N^{(N-1)(N-1)} \end{bmatrix}$$

and

$$F^{-1} = \frac{1}{N}\begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & {\omega_N} & {\omega_N}^2 & \cdots & {\omega_N}^{N-1} \\ 1 & {\omega_N}^2 & {\omega_N}^4 & \cdots & {\omega_N}^{2(N-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & {\omega_N}^{N-1} & {\omega_N}^{2(N-1)} & \cdots & {\omega_N}^{(N-1)(N-1)} \end{bmatrix}$$

The matrix $\tilde{F} = N^{-1/2}F$ is a unitary matrix ($\tilde{F}^{-1} = \tilde{F}^*$). Recall that unitary matrices preserve the length of complex vectors. This implies that the lengths of the vectors $\mathbf{f} = [f_0, f_1, \ldots, f_{N-1}]$ and $\mathbf{c} = [c_0, c_1, \ldots, c_{N-1}]$ are related by

$$\|\mathbf{c}\|^2 = N\|\mathbf{f}\|^2$$

or

$$sum_{k=0}^{N-1}|c_k|^2 = N\sum_{k=0}^{N-1}|f_k|^2$$

This is the discrete version of Parseval's formula.

III.5.2. The Fast Fourier transform

Multiplying an $N \times N$ matrix with a vector of length N normally requires N^2 multiplications, since each entry of the product requires N, and there are N entries. It turns out that the discrete Fourier transform, that is, multiplication by the matrix F, can be carried out using only $N\log_2(N)$ multiplications (at least if N is a power of 2). The algorithm that achieves this is called the Fast Fourier Transform, or FFT. This represents a tremendous saving in time: calculations that would require weeks of computer time can be carried out in seconds.

The basic idea of the FFT is to break the sum defining the Fourier coefficients c_k into a sum of the even terms and a sum of the odd terms. Each of these turns out to be (up to a factor we can compute) a discrete Fourier transform of half the length. This idea is then applied recursively. Starting with $N = 2^n$ and halving the size of the Fourier transform at each step, it takes $n = \log_2(N)$ steps to arrive at Fourier transforms of length 1. This is where the $\log_2(N)$ comes in.

To simplify the notation, we will ignore the factor of $1/N$ in the definition of the discrete Fourier transform (so one should divide by N at the end of the calculation.) We now also assume that

$$N = 2^n$$

so that we can divide N by 2 repeatedly. The basic formula, splitting the sum for c_k into a sum over odd and even j's is

$$\begin{aligned}
c_k &= \sum_{j=0}^{N-1} e^{-i2\pi kj/N} f_j \\
&= \sum_{\substack{j=0 \\ j \text{ even}}}^{N-1} e^{-i2\pi kj/N} f_j + \sum_{\substack{j=0 \\ j \text{ odd}}}^{N-1} e^{-i2\pi kj/N} f_j \\
&= \sum_{j=0}^{N/2-1} e^{-i2\pi k2j/N} f_{2j} + \sum_{j=0}^{N/2-1} e^{-i2\pi k(2j+1)/N} f_{2j+1} \\
&= \sum_{j=0}^{N/2-1} e^{-i2\pi kj/(N/2)} f_{2j} + e^{-i2\pi k/N} \sum_{j=0}^{N/2-1} e^{-i2\pi kj/(N/2)} f_{2j+1}
\end{aligned}$$

Notice that the two sums on the right are discrete Fourier transforms of length $N/2$.

To continue, it is useful to write the integers j in base 2. Lets assume that $N = 2^3 = 8$. Once you understand this case, the general case $N = 2^n$ will be easy. Recall that

$$\begin{aligned}
0 &= 000 \quad \text{(base 2)} \\
1 &= 001 \quad \text{(base 2)} \\
2 &= 010 \quad \text{(base 2)} \\
3 &= 011 \quad \text{(base 2)} \\
4 &= 100 \quad \text{(base 2)} \\
5 &= 101 \quad \text{(base 2)} \\
6 &= 110 \quad \text{(base 2)} \\
7 &= 111 \quad \text{(base 2)}
\end{aligned}$$

The even j's are the ones whose binary expansions have the form $**0$, while the odd j's have binary expansions of the form $**1$.

For any pattern of bits like $**0$, I will use the notation $F^{<pattern>}$ to denote the discrete Fourier transform where the input data is given by all the f_j's whose j's have binary expansion fitting the pattern. Here are some examples. To start, $F_k^{***} = c_k$ is the original discrete Fourier transform, since every j fits the pattern $***$. In this example k ranges over $0, \ldots, 7$, that is, the values start repeating after that.

Only even j's fit the pattern $**0$, so F^{**0} is the discrete Fourier transform of the even j's given by

$$F_k^{**0} = \sum_{j=0}^{N/2-1} e^{-i2\pi kj/(N/2)} f_{2j}.$$

Here k runs from 0 to 3 before the values start repeating. Similarly, F^{*00} is a transform of length $N/4 = 2$ given by

$$F_k^{*00} = \sum_{j=0}^{N/4-1} e^{-i2\pi kj/(N/4)} f_{4j}.$$

In this case $k = 0, 1$ and then the values repeat. Finally, the only j matching the pattern 010 is $j = 2$, so F_k^{010} is a transform of length one term given by

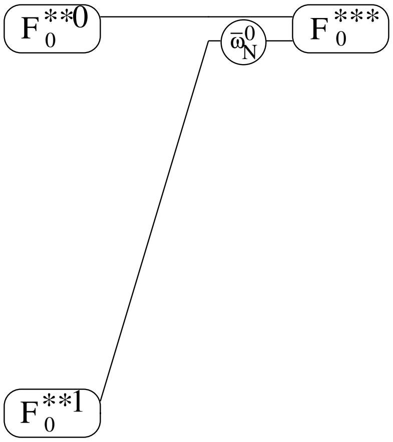
$$F_k^{010} = \sum_{j=0}^{N/8-1} e^{-i2\pi kj/(N/8)} f_2 = \sum_{j=0}^{0} e^0 f_2. = f_2$$

With this notation, the basic even–odd formula can be written

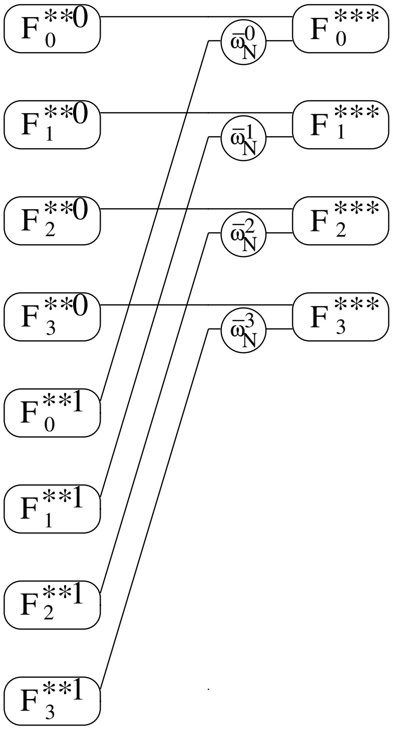
$$F_k^{***} = F_k^{**0} + \overline{\omega}_N^k F_k^{**1}.$$

Recall that $\omega_N = e^{i2\pi/N}$, so $\overline{\omega}_N = e^{-i2\pi/N}$.

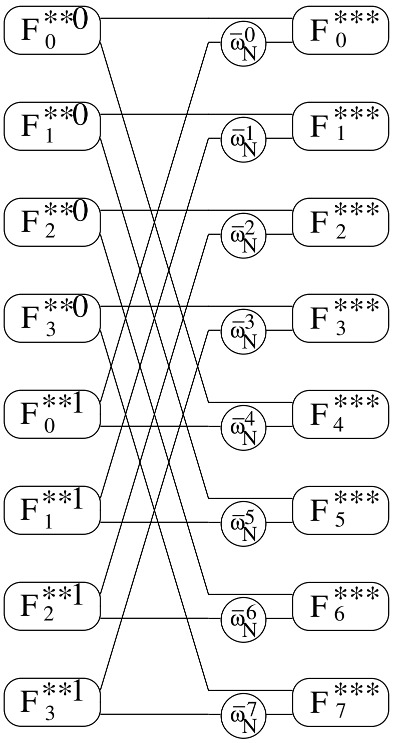
Lets look at this equation when $k = 0$. We will represent the formula by the following diagram.



This diagram means that F_0^{***} is obtained by adding F_0^{**0} to $\overline{\omega}_N^0 F_0^{**1}$. (Of course $\overline{\omega}_N^0 = 1$ so we could omit it.) Now lets add the diagrams for $k = 1, 2, 3$.



Now when we get to $k = 4$, we recall that F^{**0} and F^{**1} are discrete transforms of length $N/2 = 4$. Therefore, by periodicity $F_4^{**0} = F_0^{**0}$, $F_5^{**0} = F_1^{**0}$, and so on. So in the formula $F_4^{***} = F_4^{**0} + \overline{\omega}_N^4 F_4^{**1}$ we may replace F_4^{**0} and F_4^{**1} with F_0^{**0} and F_0^{**1} respectively. Making such replacements, we complete the first part of the diagram as follows.



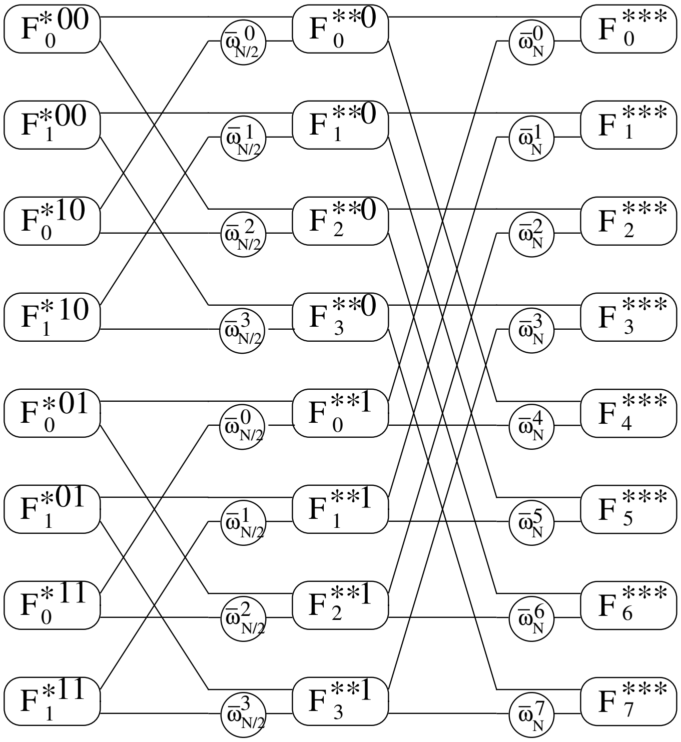
To move to the next level we analyze the discrete Fourier transforms on the left of this diagram in the same way. This time we use the basic formula for the transform of length $N/2$, namely

$$F_k^{**0} = F_k^{*00} + \overline{\omega}_{N/2}^k F_k^{*10}$$

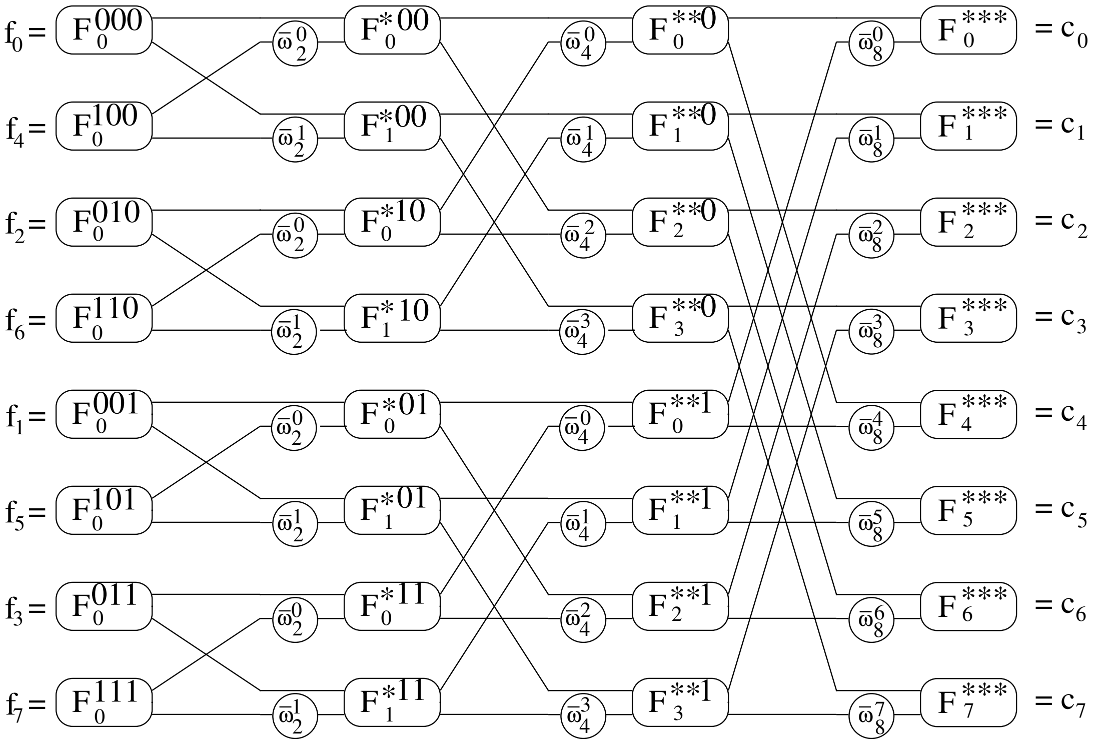
and

$$F_k^{**1} = F_k^{*01} + \overline{\omega}_{N/2}^k F_k^{*11}.$$

The resulting diagram shows how to go from the length two transforms to the final transform on the right.



Now we go down one more level. Each transform of length two can be constructed from transforms of length one, i.e., from the original data in some order. We complete the diagram as follows. Here we have inserted the value $N = 8$.



$f_0=$ F_0^{000} $\overline{\omega}_2^0$ F_0^{*00} $\overline{\omega}_4^0$ F_0^{**0} $\overline{\omega}_8^0$ F_0^{***} $= c_0$
$f_4=$ F_0^{100} $\overline{\omega}_2^1$ F_1^{*00} $\overline{\omega}_4^1$ F_1^{**0} $\overline{\omega}_8^1$ F_1^{***} $= c_1$
$f_2=$ F_0^{010} $\overline{\omega}_2^0$ F_0^{*10} $\overline{\omega}_4^2$ F_2^{**0} $\overline{\omega}_8^2$ F_2^{***} $= c_2$
$f_6=$ F_0^{110} $\overline{\omega}_2^1$ F_1^{*10} $\overline{\omega}_4^3$ F_3^{**0} $\overline{\omega}_8^3$ F_3^{***} $= c_3$
$f_1=$ F_0^{001} $\overline{\omega}_2^0$ F_0^{*01} $\overline{\omega}_4^0$ F_0^{**1} $\overline{\omega}_8^4$ F_4^{***} $= c_4$
$f_5=$ F_0^{101} $\overline{\omega}_2^1$ F_1^{*01} $\overline{\omega}_4^1$ F_1^{**1} $\overline{\omega}_8^5$ F_5^{***} $= c_5$
$f_3=$ F_0^{011} $\overline{\omega}_2^0$ F_0^{*11} $\overline{\omega}_4^2$ F_2^{**1} $\overline{\omega}_8^6$ F_6^{***} $= c_6$
$f_7=$ F_0^{111} $\overline{\omega}_2^1$ F_1^{*11} $\overline{\omega}_4^3$ F_3^{**1} $\overline{\omega}_8^7$ F_7^{***} $= c_7$

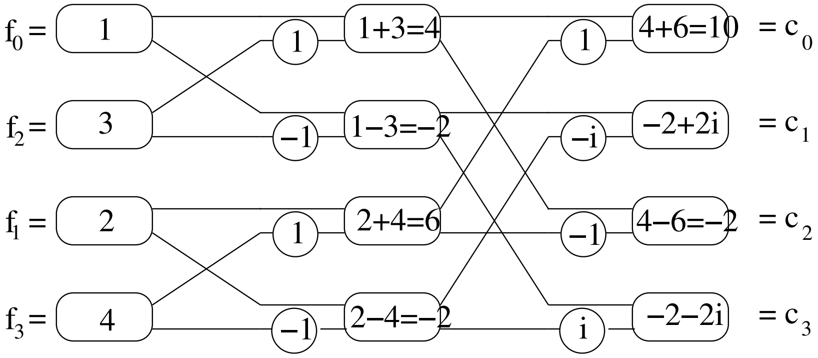
Notice that the f_j's on the left of the diagram are in *bit reversed* order. In other words, if we reverse the order of the bits in the binary expansion of the j's, the resulting numbers are ordered from 0 (000) to 7 (111).

Now we can describe the algorithm for the fast Fourier transform. Starting with the original data $[f_0, \ldots, f_7]$ we arrange the values in bit reversed order. Then we combine them pairwise, as indicated by the left side of the diagram, to form the transforms of length 2. To do this we we need to compute $\overline{\omega}_2 = e^{-i\pi} = -1$. Next we combine the transforms of length 2 according to the middle part of the diagram to form the transforms of length 4. Here we use that $\overline{\omega}_4 = e^{-i\pi/2} = -i$. Finally we combine the transforms of length 4 to obtain the transform of length 8. Here we need $\overline{\omega}_8 = e^{-i\pi/4} = 2^{-1/2} - i2^{-1/2}$.

The algorithm for values of N other than 8 is entirely analogous. For $N = 2$ or 4 we stop at the first or second stage. For larger values of $N = 2^n$ we simply add more stages. How many multiplications do we need to do? Well there are $N = 2^n$ multiplications per stage of the algorithm (one for each circle on the diagram), and there are $n = \log_2(N)$ stages. So the number of multiplications is $2^n n = N \log_2(N)$

As an example let us compute the discrete Fourier transform with $N = 4$ of the data $[f_0, f_1, f_2, f_3] = [1, 2, 3, 4]$. First we compute the bit reversed order of $0 = (00), 1 = (01), 2 = (10), 3 = (11)$ to be

$(00) = 0, (10) = 2, (01) = 1, (11) = 3$. We then do the rest of the computation right on the diagram as follows.



The MATLAB/Octave command for computing the fast Fourier transform is `fft`. Let's verify the computation above.

```
> fft([1 2 3 4])
ans =

   10 +  0i   -2 +  2i   -2 +  0i   -2 -  2i
```

The inverse fft is computed using `ifft`.

III.5.3. A frequency-amplitude plot for a sampled audio signal

Recall that a frequency-amplitude plot for the function $y(t)$ defined on the interval $[0, T]$ is a plot of the points $(\omega_n, |c_n|)$, where $\omega_n = n/T$ and c_n are the numbers appearing in the Fourier series

$$y(t) = \sum_{n=-\infty}^{\infty} c_n e^{2\pi i \omega_n t} = \sum_{n=-\infty}^{\infty} c_n e^{2\pi i n t/T}$$

If $y(t)$ represents the sound of a musical instrument, then the frequency-amplitude plot gives a visual representation of the strengths of the various frequencies present in the sound.

Of course, for an actual instrument there is no formula for $y(t)$ and the best we can do is to sample this function at a discrete set of points. Let $t_j = jT/N$ for $j = 0, \ldots, N-1$ be N equally spaced points, and let $y_j = y(t_j)$ be the sampled values of $y(t)$. Put the results in a vector $\mathbf{y} = [y_0, y_2, \ldots, y_{N-1}]^T$. How can we make an approximate frequency-amplitude plot with this information?

The key is to realize that the coefficients in the discrete Fourier transform of $\mathbf{y}$ can be used to approximate the Fourier series coefficients c_n. To see this, do a Riemann sum approximation of the integral in the formula for c_n. Using the equally spaced points t_j with $\Delta t_j = T/N$ we obtain

$$\begin{aligned} c_n &= \frac{1}{T}\int_0^T e^{-2\pi i n t/T} y(t) dt \\ &\approx \frac{1}{T}\sum_{j=0}^{N-1} e^{-2\pi i n t_j/T} y(t_j)\Delta t_j \\ &= \frac{T}{TN}\sum_{j=0}^{N-1} e^{-2\pi i n j/N} y_j \\ &= \frac{1}{N}\tilde{c}_n \end{aligned}$$

where $\tilde{c}_n$ is the nth coefficient in the discrete Fourier transform of $\mathbf{y}$

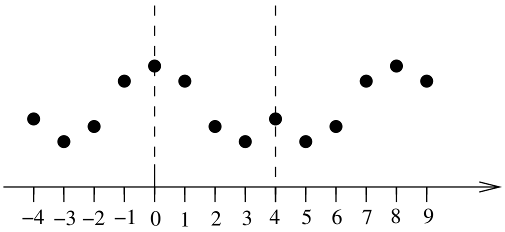
The frequency correpsonding to c_n is n/T. So, for an approximate frequency-amplitude plot, we can plot the points $(n/T, |\tilde{c}_n|/N)$. Typically we are not given T but rather the vector $\mathbf{y}$ of samples, from which we may determine the number N of samples, and the sampling frequency $F_s = N/T$. Then the points to be plotted can also be written as $(nF_s/N, |\tilde{c}_n|/N)$.

It is important to realize that the approximation $c_n \approx \tilde{c}_n/N$ is only good for small n. The reason is that the Riemann sum will do a worse job in approximating the integral when the integrand is oscillating rapidly, that is, when n is large. So we should only plot a restricted range of n. In fact, it never makes sense to plot more than $N/2$ points. To see why, recall the formula

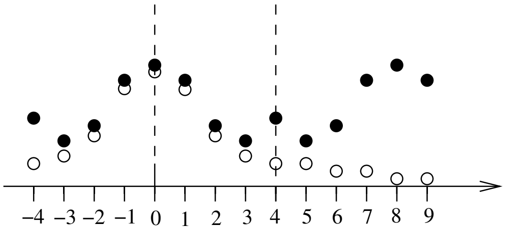
$$\tilde{c}_n = \sum_{j=0}^{N-1} e^{-2\pi i n j/N} y_j.$$

Notice that, although in the discrete Fourier transform n ranges from 0 to $N-1$, the formula for $\tilde{c}_n$ makes sense for any integer n. With this extended definition of $\tilde{c}_n$, (1) $\tilde{c}_{n+N} = \tilde{c}_n$, and (2) for $\mathbf{y}$ real valued, $\tilde{c}_{-n} = \overline{\tilde{c}_n}$. Relation (2) implies that $|\tilde{c}_{-n}| = |\tilde{c}_n|$ so that the plot of $|\tilde{c}_n|$ is symmetric about $n = 0$. But there is also a symmetry about $n = N/2$, since using (2) and then (1) we find $|\tilde{c}_{N/2+k}| = |\tilde{c}_{-N/2-k}| = |\tilde{c}_{N/2-k}|$

Here is a typical plot of $|\tilde{c}_n|$ for $N = 8$ illustrating the two lines of symmetry.



The coefficients c_n for the Fourier series obey the symmetry (2) but not (1) so if we were to add these to the plot (using the symbol $\circ$) the result might look like this:



So we see that $|\tilde{c}_7|$ should be thought of as an approximation for $|c_{-1}|$ rather than for $|c_7|$

To further compare the meanings of the coefficients c_n and $\tilde{c}_n$ it is instructive to consider the formulas (both exact) for the Fourier series and the discrete Fourier transform for $y_j = y(t_j)$:

$$y_j = \frac{1}{N}\sum_{n=0}^{N-1} \tilde{c}_n e^{2\pi i n j/N}$$

$$y(t_j) = \sum_{n=-\infty}^{\infty} c_n e^{2\pi i n t_j/T} = \sum_{n=-\infty}^{\infty} c_n e^{2\pi i n j/N}$$

The coefficients c_n and $\tilde{c}_n/N$ are close for n close to 0, but then their values must diverge so that the infinite sum and the finite sum above both give the same answer.

Now let's try and make a frequency amplitude plot using MATLAB/Octave for a sampled flute contained in the audio file `F6.baroque.au` available at

`http://www.phys.unsw.edu.au/music/flute/baroque/sounds/F6.baroque.au`.

This file contains a sampled baroque flute playing the note F_6, which has a frequency of 1396.91 Hz. The sampling rate is $F_s = 22050$ samples/second.

Audio processing is one area where MATLAB and Octave are different. The Octave code to load the file `F6.baroque.au` is

```
y=loadaudio('F6.baroque','au',8);
```

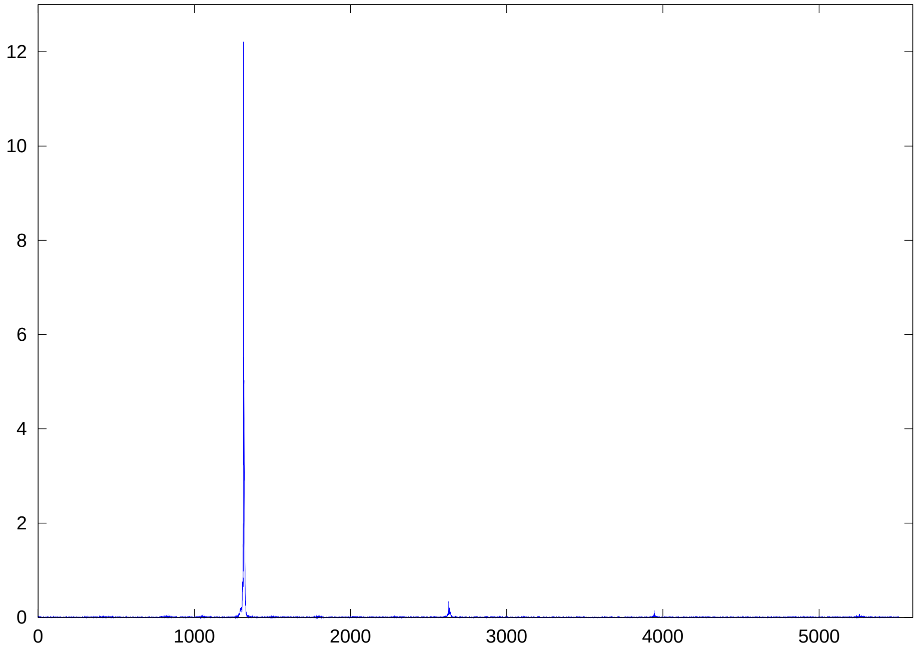
while the MATLAB code is

```
y=auread('F6.baroque.au');
```

After this step the sampled values are loaded in the vector `y`. Now we compute the FFT of `y` and store the resulting values $\tilde{c}_n$ in a vector `tildec`. Then we compute a vector `omega` containing the frequencies and make a plot of these frequencies against $|\tilde{c}_n|/N$. We plot the first `Nmax=N/4` values.

```
tildec = fft(y);
N=length(y);
Fs=22050;
omega=[0:N-1]*Fs/N;
Nmax=floor(N/4);
plot(omega(1:Nmax), abs(tildec(1:Nmax)/N));
```

Here is the result.



Notice the large spike at $\omega \approx 1396$ corresponding to the note F_6. Smaller spikes appear at the overtone series, but evidently these are quite small for a flute.