Tutorial: The Discrete Fourier Transform

Roman Valiušenko
roman.valiusenko@gmail.com

Abstract

In this article we are going to look at the discrete Fourier transform. We start off with basics of complex numbers and algebra. Then we skim over basics of linear vector spaces, vectors spaces with a norm and vector spaces with an inner product. We describe what we mean by coordinate change (i.e. expressing vector coordinates in terms of a non-natural basis).

Next, we show how to derive Euler's identity. Then we introduce complex sinusoids, show their relation to complex roots of unity, observe their properties, and, finally, show the discrete Fourier transform.

Those who is fluent in algebra can skip right to the section on Euler's identity. If you know how to derive Euler's identity, you can skip right to the sinusoids section.

After we have understood DFT, we will show how to compute it efficiently, in $O(N \log N)$ time, by employing a divide-and-conquer approach and exploiting properties of complex roots of unity (see [13]). It is a great improvement over the naïve approach which takes $O(N^2)$ time.

We don't show a non-recursive way of computing DFT here. For non-recursive algorithm see [2] and [8].

Complex numbers

Imaginary roots Let's take a look at the quadratic function $f(x) = ax^2+bx+c$, $a \neq 0$, and $a, b, c \in \mathbb{R}$. This is parabola. The canonical parabola is also given by $f(x) = (x - x_0)^2 d + e$; note that x_0 determines shift along x-axis, e determines shift along y-axis, while d determines the width of the parabola.

Suppose we want to find roots of $f(x)$, i.e. find such xs that $f(x) = 0$. Let's rewrite

$$(x - x_0)^2 d + e = (x^2 - 2xx_0 + x_0^2)d + e = dx^2 - (2dx_0)x + (x_0^2 d + e) = 0.$$

So $a = d$, $b = -2dx_0 = -2ax_0$, and $c = x_0^2 d + e = x_0^2 a + e$.

Now we solve the equation $(x - x_0)^2 d + e = 0$ for x

$$x = x_0 \pm \sqrt{-e/a}.$$

But we already know that $x_0 = -\frac{b}{2a}$, and $e = c - x_0^2 a = c - (-\frac{b}{2a})^2 a = c - \frac{b^2}{4a} = \frac{4ac-b^2}{4a}$, we have

$$x = -\frac{b}{2a} \pm \sqrt{\frac{b^2 - 4ac}{4a^2}} = -\frac{b}{2a} \pm \frac{\sqrt{b^2 - 4ac}}{2a}.$$

This is something that we all know very well, the *quadratic formula*.

We now know how to find roots of the quadratic equation. Suppose our polynomial is *monic*, $f(x) = x^2 + bx + c$, that is it's leading coefficient is $a = 1$. Suppose we have found two roots r_1 and r_2, then we can write

$$f(r_1) = f(r_2) = r_1^2 + br_1 + c = r_2^2 + br_2 + c = (x - r_1)(x - r_2) = 0.$$

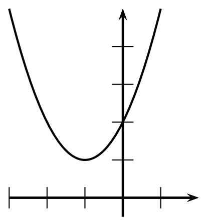

Figure 1: $y = x^2 + 2x + 2$.

But what happens if we try to solve, for example, $x^2+2x+2 = 0$? We get $x = -1 \pm \frac{\sqrt{-4}}{2}$. We can't take square root of -4, because there's no number whose square would give -4. We can write, though, $\sqrt{-1 \cdot 4} = 2\sqrt{-1}$. Or, in general, for any $\sqrt{-c} = \sqrt{|c|}\sqrt{-1}$.

We will define

$$\sqrt{-1} = i,$$

or, equivalently,

$$-1 = i^2.$$

We can now write roots formally as $r_0 = -1 + \sqrt{-1} = -1 + i$, $r_1 = -1 - i$. So

$$(x - (i - 1))(x - (-1 - i)) = (x - i + 1)(x + 1 + i) = 0.$$

Let's multiply these and see what happens:

$$x^2 + x + xi - xi - i - i^2 + x + 1 + i = x^2 + 2x - (-1) + 1 = x^2 + 2x + 2.$$

Complex numbers We will call numbers of the form

$$z = x + yi$$

complex numbers, here $x, y \in \mathbb{R}$; We will call x the *real part*, and y the *imaginary part*.

Multiplication, as we have just seen, can be defined as

$$\begin{aligned}(x_1 + y_1 i)(x_2 + y_2 i) &= (x_1x_2 + x_1y_2 i + y_1x_2 i + y_1y_2 i^2) \\ &= (x_1x_2 - y_1y_2) + (x_1y_2 + x_2y_1)i.\end{aligned}$$

And addition simply as $(x_1+y_1 i)+(x_2+y_2 i) = (x_1+x_2)+(y_1+y_2)i$. Complex numbers form a field, $\mathbb{C}$.

Complex plane and complex conjugate We can think of a complex number $z = x + yi$ as an ordered pair (x, y), and so we can plot it on a plane where "horizontal" axis is the real part, and "vertical" axis is the imaginary part.

Let's think of a Cartesian coordinate system. If we take a segment from the center of coordinates $(0, 0)$ to the point (x, y), then we can express our complex number as a pair (r, θ), where r is the length (or modulus, norm) of the segment, and θ is the angle (or phase) between the segment and x-axis. Using basic geometry we can find that $r = \sqrt{x^2 + y^2}$, and $\theta = \arctan \frac{y}{x}$.

Of course, it also follows that $x = r\cos\theta$, and $y = r\sin\theta$.

Complex conjugate is the "reflection" of a complex number "above" or "below" x-axis. That is, conjugate of $z = x + yi$ is $\overline{z} = x - yi$. Note that $z\overline{z} = (x + yi)(x - yi) = x^2 - (yi)^2 = x^2 + y^2 = |z|^2$.

Vector, Banach, Hilbert spaces

A *field* F is a set with two operations, denoted $\cdot$ and $+$ (addition and multiplication), and where the following axioms hold [7]: We require *closure* of F under $\cdot$ and $+$: $\forall x, y \in F$ we have that $x+y \in F$, and $x \cdot y \in F$. We also require operations to be *associative* and *commutative*, that is $\forall x, y, z \in F : x + (y+z) = (x+y)+z, x\cdot(y\cdot z) = (x\cdot y)\cdot z$ (associativity), and $\forall x, y \in F : x+y = y+x, x\cdot y = y \cdot x$ (commutativity). We also need an *identity* elements for both operations. This is 0 for addition: $\forall x \in F : x + 0 = x$. And 1 for multiplication: $\forall x \in F : x \cdot 1 = x$. Next, we require existence of *inverse* elements, that is $\forall x \in F \exists (-a) \in F : a + (-a) = 0$; similarly, for multiplication: $\forall x \in F \exists (x^{-1}) \in F : x \cdot x^{-1} = 1$. Finally, we need *distributivity* to hold in our structure, that is $\forall x, y, z \in F : x \cdot (y+z) = x \cdot y + x \cdot z$. It's easy to see that real numbers $\mathbb{R}$, rational numbers $\mathbb{Q}$, and complex numbers $\mathbb{C}$ are fields.

A N-length sequence x of *scalars* $(x_0, x_1, \ldots, x_{N-1})$, and where $x \in \mathbb{R}^N$ or $x \in \mathbb{C}^N$, or $x \in \mathbb{Q}^N$, is called a *vector*. Generally, scalars can be elements of any field. We can also regard a vector as a point in N-dimensional space.

Vector addition We can now define an addition of two vectors. If we are given two vectors $x = (x_0, x_1, \ldots x_N)$ and $y = (y_0, y_1, \ldots, y_N)$, their sum is defined as $x + y = (x_0 + y_0, x_1 + y_1, \ldots, x_N + y_N)$. Since scalars are elements of a field F (we will mean either $\mathbb{R}$ or $\mathbb{C}$), which is closed under $+$, $\forall x, y \in F : x_i + y_i \in F$, $i = 0, 1, \ldots, N-1$, definition makes sense. For instance, if a field is $\mathbb{R}$, and our vectors are 2-dimensional, $\mathbb{R}^2$, addition has natural geometric interpretation. It easily generalizes to N-dimensions. Also note, that addition of two vectors is commutative.

Vector subtraction Similarly, we can define a vector subtraction. Given two vectors $x = (x_0, x_1, \ldots x_N)$ and $y = (y_0, y_1, \ldots, y_N)$, their difference is defined as $x - y = (x_0 - y_0, x_1 - y_1, \ldots, x_N - y_N)$. Note that it is *not* commutative.

Scalar multiplication We can also "scale" vectors, i.e. multiply them by some value. For instance, if $\alpha \in \mathbb{R}$, and vector $x \in \mathbb{R}^N$, then $\alpha x = (\alpha x_0, \alpha x_1, \ldots, \alpha x_{N-1})$. That is we multiply each coordinate by some "scale factor."

Linear vector space We can also "combine" vectors, i.e. multiply them by some scalars and add together to get a new vector. For example, say we have M vectors x_i, $i = 0, 1, \ldots, M-1$, then their *linear combination* is the vector $y = \alpha_0 x_0 + \alpha_1 x_1 + \cdots + \alpha_{M-1} x_{M-1}$, here $\alpha_i \in F$ are scalars, and $x_i \in F^N$ are vectors. The result vector y is in F^N.

A set of vectors closed under linear combination is called *linear vector space* (see [7]). That is, it's a linear vector space if

$$\forall x_1, x_2 \in F^N, \forall \alpha_1, \alpha_2 \in F : \alpha_1 x_1 + \alpha_2 x_2 \in F^N.$$

Banach space Suppose we have a function $f : F^N \to F$ with the following properties: $\forall x \in F^N : f(x) \geq 0$, and $f(x) = 0$ only if $x = 0$; $\forall x, y \in F^N : f(x+y) \leq f(x) + f(y)$; and $\forall c \in F : f(cx) = |c| f(x)$. This function is called a *norm* on vector space F^N. In other words, a norm can be any function that assigns each nonzero vector some nonnegative value, satisfies the "triangle inequality", and which is *absolutely homogeneous* with respect to scalar multiplication [6][1].

A linear vector space with a norm is called a *normed space*, in case of a normed vector space over $\mathbb{C}$ or $\mathbb{R}$ it is also a *Banach space* [14].

For example, on an Euclidean space $\mathbb{R}^N$ the norm can be defined as a consequence of Pythagorean theorem:

$$\forall x = (x_0, x_1, \ldots, x_{N-1}) \in R^N : ||x|| = \sqrt{x_0^2 + x_1^2 + \cdots + x_{N-1}^2}.$$

This is rather natural, since we can think of it as the length of a vector.

In complex space $\mathbb{C}^N$ the norm can be defined similarly

$$\begin{aligned} ||x|| &= \sqrt{|x_0|^2 + |x_1|^2 + \cdots + |x_{N-1}|^2} \\ &= \sqrt{x_0\overline{x_0} + x_1\overline{x_1} + \cdots + x_{N-1}\overline{x_{N-1}}}, \end{aligned}$$

since $z\overline{z} = ||z||^2$, and $\overline{z}$ is the complex conjugate.

Hilbert space Let us take a look at the Euclidean space $\mathbb{R}^2$. Suppose we want to measure how much two vectors point in the same direction. We can achieve that by looking at the angle between them. The smaller the angle the more they point in the same direction. The larger the angle the less they point in the same direction. Alternatively, we can also think of the length of the projection of one vector onto another. The more the length of a given vector is equal to the length of the projected vector the more two vectors point in the same direction (assuming, though, that coefficient of projection is positive).

Suppose we have two vectors $x, y \in R^2$, then the scalar $||x||||y|| \cos\theta$, that is the length of the vector that we project on to multiplied by the coefficient of the projection, is called a *dot product*. We will denote such operation as $\cdot$, or enclose the vectors between a pair of brackets $\langle \cdot, \cdot \rangle$, so

$$x \cdot y = \langle x, y \rangle = ||x||||y|| \cos\theta \in R.$$

Note that when dealing with unit length vectors, this is just the coefficient of projection.

Suppose the angle between x and x-axis is θ_1, and the angle between y and x-axis is θ_2. Obviously, $\theta = \theta_1 - \theta_2$. We can write then,

$$\begin{aligned} x \cdot y = \langle x, y \rangle &= ||x||||y|| \cos(\theta_1 - \theta_2) \\ &= ||x||||y||(\cos\theta_1 \cos\theta_2 + \sin\theta_1 \sin\theta_2) \\ &= ||x|| \cos\theta_1 ||y|| \cos\theta_2 + ||x|| \sin\theta_1 ||y|| \sin\theta_2 = \\ &= x_1 y_1 + x_2 y_2. \end{aligned}$$

The dot product is generalized to N-dimensions:

$$\langle x, y \rangle = \sum_{i=0}^{N-1} x_i y_i.$$

Note that for our example $\mathbb{R}^2$ we have $||x|| = \sqrt{x \cdot x} = \sqrt{\sum_{i=0}^{N-1} x_i^2}$. That is, the length is induced by the dot product. We will see that dot product is also an *inner product*, which is a more general idea.

An inner product is a function that for a given pair of vectors assigns a scalar, and it has the following properties [6]:

Positivity $\forall x \neq 0 \in V : \langle x, x \rangle > 0$, and $\langle x, x \rangle = 0$ if $x = 0$
Additivity $\forall x, y, z \in V : \langle x + y, z \rangle = \langle x, z \rangle + \langle y, z \rangle$
Homogeneity $\forall x, y \in V, \alpha \in F : \langle \alpha x, y \rangle = \alpha \langle x, y \rangle$
Conjugate interchange $\forall x, y \in V : \langle x, y \rangle = \overline{\langle y, x \rangle}$

A vector space with an inner product is known as *inner product space*. So we can now define an inner product as

$$x, y \in V = F^N : \langle x, y \rangle = \sum_{i=0}^{N-1} x_i \overline{y_i} \in F.$$

Note that when applied to a real vector space $V = \mathbb{R}^N$ it becomes a dot product, since every real number equals its complex conjugate, and the last rule becomes simply $\forall x, y \in V : \langle x, y \rangle = \langle y, x \rangle$ [6].

An inner product space, where the norm is induced by the inner product, is called *Hilbert space* (see [15] for details).

Vector space basis Suppose we have some vector space V over the field F. Let us choose a minimal set of vectors $B \subset V$ such that linear combination of these vectors can produce any vector of our vector space. It is minimal in a sense that if we remove any vector from it, then the remaining vectors won't be able to produce all the vectors of the space. We call this set a *basis* of a vector space.

More formally: Suppose $B = \{v_0, v_1, \ldots, v_{N-1}\} \subset V$. If $\forall \alpha_0, \ldots, \alpha_{N-1} \in F$ we have $\alpha_0 v_0 + \cdots + \alpha_{N-1} v_{N-1} = 0$ only if $\alpha_0 = \alpha_1 = \ldots = \alpha_{N-1} = 0$, and $\forall x \in V \exists \alpha_0, \ldots, \alpha_{N-1} \in F : x = \alpha_0 v_0 + \cdots + \alpha_{N-1} v_{N-1}$, then B is a basis of V [9].

The scalars $\alpha_0, \ldots, \alpha_{N-1}$ are coordinates of the vector x defined in basis B. Each scalar α_i can be thought of as a coefficient of how much x points in the direction of the vector v_i, $i = 0, \ldots, N-1$.

Note that this set of scalars $\alpha_0, \ldots, \alpha_{N-1}$, the coordinates of the vector x, is unique in terms of the basis B. We can show this by supposing that there are two sets $\alpha_0, \ldots, \alpha_{N-1}$ and $\beta_0, \ldots, \beta_{N-1}$, and $\alpha_i \neq \beta_i$ for at least one $i = 0, 1, \ldots, N-1$, such that $x = \alpha_0 v_0 + \cdots + \alpha_{N-1} v_{N-1} = \beta_0 v_0 + \cdots + \beta_{N-1} v_{N-1}$. Then $(\alpha_0 - \beta_0) v_0 + \cdots + (\alpha_{N-1} - \beta_{N-1}) v_{N-1} = 0$. But since $\alpha_i \neq \beta_i$ for at least one $i \in [0, N-1]$ there's a nonzero term $(\alpha_i - \beta_i) v_i$. But it cannot be cancelled by some other term, since that would mean v_i are not linearly independent. So $\alpha_i = \beta_i$. We have proven that $\alpha_0, \ldots, \alpha_{N-1}$ uniquely represent x in terms of B.

A set of vectors in F^N

$$\{e_i = [\overbrace{0, 0, \ldots, \underbrace{1}_{\text{ith}}, \ldots, 0}^{N}] : i \in [0, N-1]\}$$

is called *natural basis*.

Orthogonality and projection If $\langle x, y\rangle = 0$ we say that x and y are *orthogonal*. We can also write $x \perp y$. For example, in $\mathbb{R}^2$ this would mean that they are perpendicular, that is, intersect at right angle. It would also mean, due to Pythagorean theorem, that $||x+y||^2 = ||x||^2 + ||y||^2$. It works for N-dimension too: $||x+y||^2 = \langle x+y, x+y\rangle = \langle x, x\rangle + \langle x, y\rangle + \langle y, x\rangle + \langle y, y\rangle = ||x||^2 + ||y||^2$, since $\langle x, y\rangle$ and $\langle y, x\rangle$ are 0. But note that $||x+y||^2 = ||x||^2 + ||y||^2$ doesn't necessary mean that $x \perp y$.

Let us find the *coefficient of projection* of x onto y in some inner product vector space V. Basically, the projection of x onto y is some vector $\Pr(x, y) = \alpha y$, and α is the coefficient of projection. By definition we have $x - \Pr(x, y) \perp y$, or $\langle x - \alpha y, y\rangle = 0$, so $\langle x, y\rangle = \langle \alpha y, y\rangle$, thus $\alpha = \frac{\langle x, y\rangle}{\langle y, y\rangle}$. So the projection of x onto y is

$$\Pr(x, y) = \langle x, y\rangle \frac{y}{||y||^2}.$$

Changing coordinates Suppose we have a vector x defined in terms of natural basis $B_1 = \{e_0, e_1, \dots, e_{N-1}\}$ of some vector space V. Suppose we have another basis $B_2 = \{v_0, v_1, \dots, v_{N-1}\}$. We can now define x in terms of basis B_2 by calculating coefficients of projection onto each vector in B_2, i.e.

$$x_{B_2} = \left(\frac{\langle x, v_0\rangle}{||v_0||^2}, \frac{\langle x, v_1\rangle}{||v_1||^2}, \dots, \frac{\langle x, v_{N-1}\rangle}{||v_{N-1}||^2} \right).$$

Thus, we have changed vector coordinates from one basis B_1 to another B_2.

We can easily reconstruct the original vector:

$$x = \frac{\langle x, v_0\rangle}{||v_0||^2} v_0 + \frac{\langle x, v_1\rangle}{||v_1||^2} v_1 + \dots + \frac{\langle x, v_{N-1}\rangle}{||v_{N-1}||^2} v_{N-1}.$$

Note that in order to be able to reconstruct the original vector the new basis must be a set of linearly independent and orthogonal vectors. Any linearly independent vectors in $\mathbb{C}^N$ can be orthogonalized by *Gram-Schmidt orthogonalization* process (see [1], [9] and [6]).

Euler's identity

Euler's identity relates complex exponential function with trigonometric functions.

Let us first define what we mean by writing a^x, when $a, x \in \mathbb{R}$.

Exponentials It's easy when a^n, and $a \in \mathbb{R}$, $n \in \mathbb{Z}$, since in that case we define

$$a^n = \underbrace{a \cdot a \cdot a \cdot \ldots \cdot a}_{n}.$$

By the definition given above, naturally, $a^{n_1} a^{n_2} = a^{n_1+n_2}$, $(a^{n_1})^{n_2} = a^{n_1 n_2}$. Note that $a^0 = 1$ because $a^0 a^1 = a^{0+1} = a^1$, and $a^{-1} = \frac{1}{a}$, because $a^{-1} a^1 = a^0 = 1$; generally, $a^{-n} = \frac{1}{a^n}$.

Now we can define what we mean when we write a^x, $x \in \mathbb{Q}$. Since $x \in \mathbb{Q}$, we can write $a^{\frac{m}{n}} = (a^{\frac{1}{n}})^m$, $x = \frac{m}{n}$, and $m, n \in \mathbb{Z}$. But note that $(a^{\frac{1}{n}})^n = a$, which means that $a^{\frac{1}{n}}$ is the n-th root of a. We also write $\sqrt[n]{a}$. So $a^{\frac{m}{n}}$ means n-th root of a to the power of m.

For a given real number we can find rational number that is close as much to the real one as we want, so we define a^x as the limit $a^x = \lim_{r \to x} b^r, x \in \mathbb{R}, r \in \mathbb{Q}$.

Euler's number We define constant e as the limit [5]

$$\lim_{n\to\infty} \left(1 + \frac{1}{n}\right)^n .$$

It's value to 40 decimal places is [4]

$$e = 2.7182818284590452353602874713526624977572\ldots$$

This constant is a transcendental number [16].

Taylor series Suppose we have a function $f(x)$. We want to approximate it using some n-th order polynomial:

$$f(x) = f_0 + f_1 x + f_2 x^2 + \cdots + f_n x^2 + R$$

where R is approximation error.

Actually what we want is to find f_i, $i \in 0, 1, \ldots, n$.

Suppose $R = 0$, then $f(0) = f_0$. Let us take now a derivation of $f(x)$

$$f'(x) = 0 + f_1 + 2f_2 x + 3f_3 x^2 + \cdots + n f_n x^{n-1}.$$

And $f'(0) = f_1$. Similarly,

$$f''(x) = 2f_2 + 3 \cdot 2f_3 x + \cdots + n(n-1) f_n x^{n-2}.$$

And $f''(0) = 2f_2$. In the same way we get $f^{(n)}(0) = n! f_n$.

Solving for the constants gives $f_0 = f(0), f_1 = f'(0), \ldots, f_n = \frac{f^{(n)}(0)}{n!}$. So our approximation is

$$f(x) \approx \sum_{k=0}^{n} \frac{f^{(n)}(0)}{n!} x^k.$$

To understand behaviour of the remainder R (which for simplicity we assumed to be 0) refer to any decent calculus book, or [1].

Derivative We now have a tool that allows us to approximate functions. We could try to approximate $f(x) = a^x$ using Taylor series. But first we need to find derivative of $f(x)$. We can rewrite $a^x = e^{\ln(a^x)} = e^{x \ln a}$, and, using the rule $f'(g(x)) = f'(g(x))g'(x)$, find $f'(x) = \frac{d}{dx} a^x = a^x \ln a$. Obviously, if $a = e$, we get $f'(x) = \frac{d}{dx} e^x = e^x$.

The rule $f'(g(x)) = f'(g(x))g'(x)$ can be derived as follows.

If $g(x)$ is differentiable at point x, then

$$\lim_{h\to 0} \frac{g(x+h) - g(h)}{h} - g'(x) = 0.$$

Let's define

$$\alpha = \frac{g(x+h) - g(h)}{h} - g'(x).$$

Similarly, if $f(x)$ is differentiable at point $\hat{x} = g(x)$, then

$$\lim_{\hat{h}\to 0} \frac{f(\hat{x}+\hat{h}) - f(\hat{h})}{\hat{h}} - f'(\hat{x}) = 0,$$

and let's define

$$\beta = \frac{f(\hat{x}+\hat{h}) - f(\hat{h})}{\hat{h}} - f'(\hat{x}).$$

So,

$$g(x+h) = g(x) + (g'(x) + \alpha)h_0,$$

and

$$f(\hat{x}+\hat{h}) = f(\hat{x}) + (f'(\hat{x}) + \beta)\hat{h}.$$

We now can write

$$f(g(x+h)) = f(g(x) + (g'(x)+\alpha)h) = f(g(x)) + (f'(g(x)) + \beta)(g'(x)+\alpha)h.$$

Using this result it's quite easy to show the rule:

$$\begin{aligned}
f'(g(x)) &= \lim_{h\to 0} \frac{f(g(x+h)) - f(g(h))}{h} \\
&= \lim_{h\to 0} \frac{f(g(x) + (g'(x)+\alpha)h) - g(h)}{h} \\
&= \lim_{h\to 0} \frac{f(g(x)) + (f'(g(x)) + \beta)(g'(x)+\alpha)h - g(h)}{h} \\
&= \lim_{h\to 0} (f'(g(x)) + \beta)(g'(x)+\alpha) \\
&= \left(\lim_{h\to 0} f'(g(x)) + \lim_{h\to 0} \beta\right)\left(\lim_{h\to 0} g'(x) + \lim_{h\to 0} \alpha\right) \\
&= f'(g(x))g'(x).
\end{aligned}$$

This rule is also known as the *chain rule* [5].
The Taylor series for e^x is quite simple then:

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \frac{x^5}{5!} + \frac{x^6}{6!} \cdots .$$

The series for $\cos(x)$ is

$$\cos(x) = 1 - \frac{1}{2}x^2 + \frac{1}{4!}x^4 - \frac{1}{6!}x^6 + \frac{1}{8!}x^8 \cdots .$$

$$\frac{d^n}{dx^n}\cos(x)|_{x=0} = \begin{cases} (-1)^{n/2} & \text{if } n \text{ is even} \\ 0 & \text{if } n \text{ is odd} \end{cases}$$

The series for $\sin(x)$ is

$$\sin(x) = x - \frac{1}{3!}x^3 + \frac{1}{5!}x^5 - \frac{1}{7!}x^7 + \frac{1}{9!}x^9 + \cdots .$$

$$\frac{d^n}{dx^n}\sin(x)|_{x=0} = \begin{cases} (-1)^{(n-1)/2} & \text{if } n \text{ is odd} \\ 0 & \text{if } n \text{ is even} \end{cases}$$

Note that the sum of $\sin(x)$ and $\cos(x)$ almost equals e^x:

$$\sin(x) + \cos(x) = 1 + x - \frac{1}{2!}x^2 - \frac{1}{3!}x^3 + \frac{1}{4!}x^4 + \frac{1}{5!}x^5 + \cdots .$$

The problem is that sings in the expansion of e^x don't match. However, if we use ix instead of simply x, ($i^2 = -1$), that is find Taylor series for e^{ix}, we get

$$e^{ix} = 1 + ix + \frac{1}{2!}(ix)^2 + \frac{1}{3!}(ix)^3 + \frac{1}{4!}(ix)^4 + \frac{1}{5!}(ix)^5 + \cdots .$$

Or, when simplified a bit

$$e^{ix} = 1 + ix - \frac{1}{2!}x^2 - \frac{1}{3!}ix^3 + \frac{1}{4!}x^4 + \frac{1}{5!}ix^5 + \cdots .$$

So, the Euler's Identity is

$$e^{ix} = \cos(x) + i\sin(x).$$

Sinusoids

A sinusoid (or a sine wave) is defined as

$$y(t) = A\sin(\omega t + \phi),$$

where A is the amplitude, ω is the angular frequency measured in radians per second ($\omega = \frac{2\pi}{T}$, T is the period in seconds), and ϕ is the phase.

Alternatively, we could write $y(t) = A\sin(2\pi f_s t + \phi)$, where $f_s = \frac{1}{T}$ is the frequency. Note that $\cos(\omega t)$ is the same as $\sin(\omega t + \frac{\pi}{2})$ (see the figure).

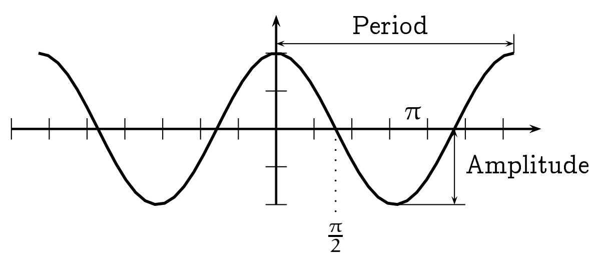

Figure 2: Sinusoid $f(x) = 2\sin(x - \frac{\pi}{2})$.

Complex sinusoids Multiplying Euler's identity by $A \geq 0$ we get the *complex sinusoid*

$$f(t) = Ae^{i(\omega t+\phi)} = A\left(\cos(\omega t + \phi) + i\sin(\omega t + \phi)\right).$$

A complex sinusoid has a *constant modulus*:

$$|Ae^{i(\omega t+\phi)}| = A,$$

because $\sin^2(x) + \cos^2(x) = 1$.

Since the modulus is constant, a complex sinusoid traces a circle in the complex plane [1]. As the parameter t increases in $f(t) = Ae^{i(\omega t+\phi)}$, the motion goes in the counter-clockwise direction; and for $\overline{f(t)} = Ae^{-i(\omega t+\phi)}$ we have clockwise motion.

Sampled complex sinusoids Since we are going to work with discrete data, we need a discrete version of a complex sinusoid. Suppose sampling rate is f_s, for example 44.1kHz. This means, that one second of our signal contains 44.1K samples. Let $T = \frac{1}{f_s}$ be the *sampling interval.*

A sampled complex sinusoid is a function

$$s(n) = Ae^{i(\omega nT+\phi)}, n = 0, 1, \ldots,$$

where A is the amplitude, n is the sample number.

Roots of unity & sinusoids Note that $e^{i2\pi k} = \cos(2\pi k) + i\sin(2\pi k) = 1$, so we can write

$$1^{\frac{k}{M}} = e^{i2\pi k\frac{1}{M}} = \left(e^{\frac{i2\pi}{M}}\right)^k,$$

where $k = 0, 1, \ldots, M-1$; these are the M-th roots of unity, because $\left(e^{\frac{i2\pi k}{M}}\right)^M = 1$.

When $k = 1$, we call $e^{\frac{i2\pi}{M}}$ *primitive* M*-th root of unity*, since it generates all of the others roots. We will denote primitive M-th root of unity as $W_M = e^{\frac{i2\pi}{M}}$, and $W_M^k = e^{\frac{i2\pi k}{M}}$, $k = 0, 1, \ldots, M-1$.[1]

We can think of roots of unity as complex sinusoids with amplitude $A = 1$ and phase $\phi = 0$.

We will use the sampled version of those sinusoids

$$s_k(n) = \left(W_N^k\right)^n = e^{\frac{i2\pi kn}{N}} = \cos\left(\frac{2\pi kn}{N}\right) + i\sin\left(\frac{2\pi kn}{N}\right),$$

where $n = 0, 1, \ldots, N-1$.

Note that the sinusoids $s_k(n)$ we generate have frequencies $f_k = kf_s/N$, $k = 0, 1, \ldots, N-1$, f_s is the sampling rate; these are the only frequencies that have whole number of periods in N samples [1].

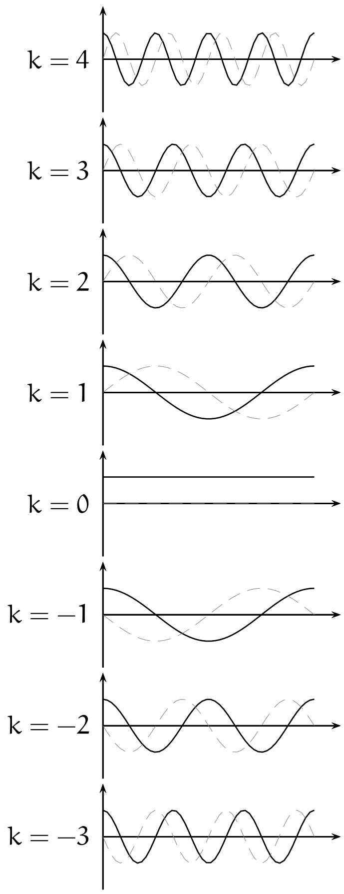

Figure 3: Sinusoids $s_k(x) = e^{\frac{i2\pi xk}{N}}$, $k = -3, -2, \ldots, 4$, $N = 8$. Gray dotted sinusoids are the imaginary parts of the complex sinusoids.

Orthogonality of sinusoids We can show that sinusoids generated by roots of unity $s_k(n) = \left(W_N^k\right)^n$, $n = 0, 1, \ldots, N-1$, are orthogonal:

$$\langle s_k, s_l\rangle = \sum_{n=0}^{N-1} s_k(n)\overline{s_l(n)} = \sum_{n=0}^{N-1} e^{\frac{i2\pi kn}{N}} e^{-\frac{i2\pi kl}{N}} = \sum_{n=0}^{N-1} e^{\frac{i2\pi n(k-l)}{N}} = \frac{1-e^{i2\pi n(k-l)}}{1-e^{\frac{i2\pi n(k-l)}{N}}}$$

It is zero when $k \neq l$, so $s_k \perp s_l$.

[1] Some authors use ω to denote roots of unity (for instance [2]), but here we already used it for angular frequnecy, so we will use W to avoid confusion.

This means that sinusoids generated by the roots of unity can be used as a basis in a vector space. So we can express coordinates of a given vector in terms of the basis formed by a set of sampled complex sinusoids generated by the roots of unity. Essentially that is the Fourier transform!

The Discrete Fourier Transform

Suppose we have a vector $x \in \mathbb{C}^N$, then

$$\mathrm{DFT}_N(x) = \{X_{\omega_0}, X_{\omega_1}, \ldots, X_{\omega_{N-1}}\}$$

is the *Discrete Fourier Transform*, where

$$X_{\omega_k} = \langle x, s_k \rangle = \sum_{n=0}^{N-1} x(n)\overline{s_k(n)} : k = 0, 1, \ldots, N-1,$$

and $s_k(n) = e^{\frac{i2\pi kn}{N}}$, $\omega_k = \frac{2\pi k f_s}{N}$, $f_s = \frac{1}{T}$; T is the sampling interval.
Note that

$$\langle s_k, s_k \rangle = \sum_{n=0}^{N-1} e^{\frac{i2\pi(k-k)n}{N}} = N.$$

The DFT is proportional to the coefficients of projection of a vector onto basis vectors s_k. In fact it is N times the coefficient of projection [1], because

$$\Pr(x, s_k) = \frac{\langle x, s_k \rangle}{||s_k||^2} s_k = \frac{X(\omega_k)}{N} s_k.$$

Having the coefficients of projection we can reconstruct the original vector:

$$x = \sum_{k=0}^{N-1} \frac{X(\omega_k)}{N} s_k = \frac{1}{N} \sum_{k=0}^{N-1} X(\omega_k) e^{\frac{i2\pi kn}{N}},$$

where $n = 0, 1, \ldots, N-1$. This is the *inverse Discrete Fourier Transform.*

Obviously, computing the DFT in a straightforward way takes $O(N^2)$ time which doesn't make it very practical. We will look at the algorithm known as Fast Fourier Transform which computes DFT in $O(N \log N)$ time.

The Fast Fourier Transform

Calculating the DFT is, in fact, evaluation of a polynomial $A(x) = \sum_{m=0}^{N-1} a_m x^m$ at the N complex N-th roots of unity, where $a = (a_0, a_1, \ldots, a_{N-1})$ is the vector whose discrete Fourier transform we want to calculate [2]. We will write $\mathrm{DFT}_N(a) = \big(A(W_N^0), A(W_N^k), \ldots, A(W_N^{N-1})\big)$.

The key idea of calculating DFT quickly is to sum separately on odd and even coefficients of the polynomial and employing properties of roots of unity. Assuming that N is exact power of 2

$$A(x) = \sum_{m=0}^{N/2-1} a_{2m} x^{2m} + \sum_{m=0}^{N/2-1} a_{2m+1} x^{(2m+1)},$$

we can define two new polynomials,

$$A^{[0]}(x) = \sum_{m=0}^{N/2-1} a_{2m} x^{m},$$

for even coefficients, and, similarly, for odd coefficients,

$$A^{[1]}(x) = \sum_{m=0}^{N/2-1} a_{2m+1} x^{m}.$$

Now we can define $A(x)$ in terms of $A^{[0]}(x)$ and $A^{[1]}(x)$ as

$$A(x) = A^{[0]}(x^2) + xA^{[1]}(x^2).$$

So evaluation of $A(x)$ can be computed by evaluating $A^{[0]}(x)$ and $A^{[1]}(x)$ at the points $(W_N^0)^2, (W_N^1)^2, \ldots, (W_N^{N-1})^2$.

Note, however, that $(W_N^k)^2 = W_{N/2}^k$, and $W_N^{N/2} = W_N = -1$. The latter implies that $W_N^{k+N/2} = -W_N^k$, so squares of $W_N^{k+N/2}$ and W_N^k are equal:

$$\left(W_N^{k+N/2}\right)^2 = \left(W_N^k\right)^2.$$

It means, that those squared N complex roots of unity will contain $N/2$ complex $N/2$-th roots of unity, with each root occurring twice.

The result here is that we divided the problem into two subproblems of the same form, but half the original size. Evaluation of N degree polynomial $A(x)$ reduces to evaluation of $N/2$ degree polynomials $A^{[0]}(x)$ and $A^{[1]}(x)$ at $N/2$ complex $N/2$-th roots of unity. We can divide each of these subproblems recursively in the same manner, until $N = 1$. So the algorithm becomes a *divide-and-conquer* algorithm.

Suppose we have calculated two $\mathrm{DFT}_{N/2}$-s for even and odd indices,

$$A^{[0]} = \left(A^{[0]}(W_{N/2}^0), A^{[0]}(W_{N/2}^1), \ldots, A^{[0]}(W_{N/2}^{N/2-1})\right)$$

$$A^{[1]} = \left(A^{[1]}(W_{N/2}^0), A^{[1]}(W_{N/2}^1), \ldots, A^{[1]}(W_{N/2}^{N/2-1})\right).$$

Let $y_k^{[0]}$ to denote k-th element of $A^{[0]}$, and $y_k^{[1]}$ to denote k-th element of $A^{[1]}$. Because $(W_N^k)^2 = W_{N/2}^k$, we also have that

$$y_k^{[0]} = A^{[0]}(W_N^{2k})$$

and

$$y_k^{[1]} = A^{[1]}(W_N^{2k}),$$

for $k = 0, 1, \ldots, N/2 - 1$. We want to calculate the resulting Fourier transform

$$\mathsf{DFT}_N = (y_0, y_1, \ldots, y_{N-1}) = \left(A(W_N^0), A(W_N^k), \ldots, A(W_N^{N-1})\right).$$

using $y_k^{[0]}$-s and $y_k^{[1]}$-s.

For $k = 0, 1, \ldots, N/2 - 1$ we have

$$y_k = A(W_N^k) = A^{[0]}(W_N^{2k}) + W_N^k A^{[1]}(W_N^{2k}) = y_k^{[0]} + W_N^k y_k^{[1]}.$$

For the second half of y_k-s it is

$$\begin{aligned} y_{k+\frac{N}{2}} = A(W_N^{k+\frac{N}{2}}) &= A^{[0]}(W_N^{2k+N}) + W_N^{k+\frac{N}{2}} A^{[1]}(W_N^{2k+N}) \\ &= A^{[0]}(W_N^{2k}) + W_N^{k+\frac{N}{2}} A^{[1]}(W_N^{2k}) \\ &= y_k^{[0]} + W_N^{k+\frac{N}{2}} y_k^{[1]} \\ &= y_k^{[0]} - W_N^k y_k^{[1]}. \end{aligned}$$

Now we can translate the above directly into code [2]:

The Fast Fourier Transform Algorithm

```
function DFT(a)
    N ← length(a)
    if N = 1 then
        return a
    end if
    W_N ← e^(2πi/N)
    W ← 1
    A^[0] ← DFT((a_0, a_2, ..., a_{N−2}))    ▷ Calculate DFT for even indexed coefficients
    A^[1] ← DFT((a_1, a_3, ..., a_{N−1}))    ▷ Calculate DFT for odd indexed coefficients
    for k = 0 to N/2 − 1 do
        y_k ← y_k^[0] + W · y_k^[1]
        y_{k+N/2} ← y_k^[0] − W · y_k^[1]
        W ← W · W_N
    end for
    return y                                 ▷ Returns y = (y_0, y_1, ..., y_{N−1})
end function
```

Running time Suppose $T(N)$ is the time needed to calculate DFT_N (N is exact power of 2). We have learned how to break DFT_N problem into two problems but half the size, which takes $T(N/2)$ time each. We used those two DFTs to get the final result. That takes 3 steps executed $N/2$ times (the **for**-loop in the above algorithm). Running time of the whole procedure is thus

$$T(N) = 2T(N/2) + 1.5N.$$

Now let's unfold[2] the recurrence above and see what happens:

$$\begin{aligned} T(N) &= 2T(N/2) + 1.5N = 2(\cdot 2T(N/4) + 1.5N/2) + 1.5N \\ &= 4T(N/4) + 2 \cdot 1.5N/2 + 1.5N \\ &= 4(2T(N/8) + 1.5N/4) + 2 \cdot 1.5N/2 + 1.5N \\ &= 8T(N/8) + 4 \cdot 1.5N/4 + 2 \cdot 1.5N/2 + 1.5N \\ &= 8(2T(N/16) + 1.5N/8) + 4 \cdot 1.5N/4 + 2 \cdot 1.5N/2 + 1.5N = \dots \end{aligned}$$

So the pattern is obvious, and once we have reached $\frac{N}{N} = 1$, we stop and get the exact sum:

$$T(N) = 1.5N(1 + 1 + \cdots + 1).$$

The sum in parentheses is equal to $\log_2 N$, so the running time is

$$T(N) = O(1.5N \log_2 N) = O(N \log N).$$

Notes

Good introduction to DFT mathematics with audio application is [1]. A good introduction to the algorithm can be found in [8].

The algorithm described above was apparently known to Gauss [11]. However, the algorithm became widely known only after Cooley and Tukey published their article in 1965 and re-discovered the algorithm[3] [13]. Interestingly enough, the algorithm had been known and used for many years before re-discovery without being regarded as having great importance [12].

For various practical considerations while implementing the algorithm see [10]. For non-recursive DFT calculation methods see [2] and [8].

References

[1] Julius O. Smith III, Mathematics of the Discrete Fourier Transform with Audio Applications, Second Edition, CCRMA, Department of Music, Stanford University, 2008

[2] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, Introduction to Algorithms, 2009

[3] Robert Sedgewick, Philippe Flajolet, An Introduction to the Analysis of Algorithms, 2001

[2] In this case finding the closed form of the recurrence is trivial. A great source of advanced algorithm analysis techniques is [3].

[3] For that reason the algorithm is also known as Cooley-Tukey algorithm.

[4] Donald E. Knuth, The Art of Computer Programming, Volume 2, Seminumerical Algorithms, 2007

[5] Gilbert Strang, Calculus, 1991

[6] Sheldon Axler, Linear Algebra Done Right, 1997

[7] Peter Petersen, Linear Algebra, 2012

[8] S. Dasgupta, C.H. Papadimitriou, and U.V. Vazirani, Algorithms, 2006

[9] K. R. Matthews, Elementary Linear Algebra, 2010

[10] Seven G. Johnson, Matteo Frigo, Implementing FFTs in Practice

[11] M. T. Heideman, D. H. Johnson and C. S. Burrus, Gauss and the history of the fast Fourier transform, Archive for History of Exact Sciences, 34, 1985

[12] J. W. Cooley, The re-discovery of the fast Fourier transform algorithm, Mikrochimica Actra, III, 1987

[13] J. W. Cooley, J. W. Tukey, An algorithm for machine calculation of complex Fourier series, Math. Comp., 19, 1965

[14] P. K. Jain, Khalil Ahmad, Functional analysis, 1995

[15] Gerald Teschl, Topics in Real and Functional Analysis, 2010

[16] Ivan Niven, Irrational Numbers, 2005