Discrete Fourier transform

We will work with vectors from $\mathbb{C}^M$.

Consider set of functions $q^{(k)}(\theta) = \frac{1}{\sqrt{M}} e^{ik\theta}$, $k = 0, ..., M-1$ on the interval $[0, 2\pi]$ (here $i = \sqrt{-1}$). Let us discretize $q^{(k)}(\theta)$ by computing their values at M equidistant points $\theta_j = j\Delta\theta$, with step $\Delta\theta = 2\pi/M$. Then the points are

$$\theta_j = \frac{2\pi j}{M}, \quad j = 0, 1, 2, .., M-1,$$

and each function $q^{(k)}(\theta)$ yields a column-vector $q^{(k)}$:

$$q^{(k)} = \frac{1}{\sqrt{M}} \begin{pmatrix} \exp(0) \\ \exp\left(k\frac{2\pi i}{M}\right) \\ \exp\left(2k\frac{2\pi i}{M}\right) \\ \exp\left(3k\frac{2\pi i}{M}\right) \\ ... \\ \exp\left((M-1)k\frac{2\pi i}{M}\right) \end{pmatrix}. \tag{1}$$

In other words, the j^{th} coordinate of vector $q^{(k)}$ is

$$q_j^{(k)} = \frac{1}{\sqrt{M}} \exp\left(jk\frac{2\pi i}{M}\right).$$

Proposition 1 *Vectors $q^{(k)}$, $k = 0, ..., M-1$ form an orthonormal set in $\mathbb{C}^M$.*

In order to prove this proposition we first prove two lemmas presented below.

Lemma 2 *For any $k = 0, ..., M-1$,*

$$||q^{(k)}||^2 =< q^{(k)}, q^{(k)} >= 1.$$

Prove it, please. (This is a part of a homework).

Lemma 3 *Suppose $\alpha \in \mathbb{C}$ is an M^{th} root of 1, i.e. $\alpha^M = 1$ and $\alpha \neq 1$. Then*

$$1 + \alpha + ... + \alpha^{M-1} = 0.$$

Prove it, please. (This is a part of a homework).

In order to prove Proposition 1, notice that $\exp(\left((k-l)\frac{2\pi i}{M}\right)$ is an M^{th} root of 1. Please, complete the proof of Proposition 1. (Homework).

Proposition 4 *Consider a square matrix Q whose columns are vectors $q^{(k)}$, $k = 0, ..., M-1$. Then Q is unitary, and, thus*

$$Q^{-1} = Q^*.$$

This follows directly from the material of Chapter 2 (Trefethen, Bau, "Numerical Linear Algebra").

Proposition 5 *Any vector $v \in \mathbb{C}^M$ can be represented (uniquely) as the linear combination of the columns of Q (i.e. vectors $q^{(k)}$, $k = 0, ..., M-1$):*

$$v = c_0 q^{(0)} + ... + c_{M-1} q^{(M-1)} = Q \begin{pmatrix} c_0 \\ c_1 \\ ... \\ c_{M-1} \end{pmatrix} = Q\vec{c}$$

with the coefficients c_j obtained by multiplying v with $Q^{-1} = Q^*$:

$$\vec{c} = Q^* v.$$

Alternatively, since rows of Q^* are the adjoints of the columns of Q:

$$c_j = \left(q^{(j)}\right)^* v = < q^{(j)}, v > .$$

This also follows directly from the material of Chapter 2.

Fast Fourier transform

There exist a Fast Fourier transform (FFT) algorithm (really, a family of algorithms) that computes matrix vector products $Q\vec{c}$ and Q^*v extremely fast. The conventional matrix-vector multiplication requires $\mathcal{O}(M^2)$ floating point operations. The FFT computes the result in $\mathcal{O}(M \ln M)$ operations. For very large values of M the speed-up is dramatic.

The FFT algorithm is pretty complicated. However, there exist a lot of freely available high quality implementations of this technique in most computational languages (including MATLAB, C, Fortran and so on). Many of these routines require the length of the vector M be a power of 2. However, more advanced techniques also exist that work very well if M can be represented as a product of small primes (2,3,5, sometimes 7 and 11).

Negative frequencies

In order to use the DFT and FFT as discrete versions of the Fourier series, one needs to utilize not only the positive but also the negative frequencies. On the interval $[0, 2\pi]$ let us consider M functions $q^{(k)}(\theta) = \frac{1}{\sqrt{M}} e^{ik\theta}$ with $k = -M/2, ..., M/2 - 1$ (we assume that M is an even number). We discretize these functions as before, and obtain vectors $q^{(k)}$ defined by the formula (1), with $k = -M/2, ..., M/2 - 1$. We notice that

$$q^{(k+M)} = q^{(k)},$$

since

$$\begin{aligned} q_j^{(k+M)} &= \frac{1}{\sqrt{M}} \exp\left(j(k+M)\frac{2\pi i}{M}\right) = \frac{1}{\sqrt{M}} \exp\left(jk\frac{2\pi i}{M}\right) \exp\left(j2\pi i\right) \\ &= \frac{1}{\sqrt{M}} \exp\left(jk\frac{2\pi i}{M}\right) = q_j^{(k)}. \end{aligned}$$

Therefore, the set of vectors $q^{(k)}$ with $k = -M/2, ..., M/2 - 1$ actually coincides with the set of vectors $q^{(k)}$ with $k = 0, M - 1$ and therefore all the formulas remain valid. In particular, any vector $v \in \mathbb{C}^M$ can be represented (uniquely) as the linear combination of the columns of matrix Q_{new} whose columns are vectors $q^{(k)}$, $-M/2, ..., M/2 - 1$):

$$v = c_{-M/2} q^{(-M/2)} + ... + c_{M/2-1} q^{(M/2-1)} = Q_{\text{new}} \begin{pmatrix} c_{-M/2} \\ c_1 \\ ... \\ c_{M/2-1} \end{pmatrix} = Q_{\text{new}} \vec{c}_{\text{new}}$$

with the coefficients c_j obtained by multiplying v with $Q_1^{-1} = Q_1^*$:

$$\vec{c}_{\text{new}} = Q_{\text{new}}^* v,$$

or

$$c_j = \left(q^{(j)}\right)^* v =< q^{(j)}, v >, \quad j = -M/2, ..., M/2 - 1.$$

In order to compute coefficients $c_j \quad j = -M/2, ..., M/2 - 1$ using existing FFT routines one first computes $\vec{c} = Q^* v$ and then obtains $\vec{c}_{\text{new}}$ by reordering the components, since the second half of components of $\vec{c} = Q^* v$ represents the higher frequency Fourier coefficients that are equal to the negative frequencies coefficients stored in the first half of the vector $\vec{c}_{\text{new}}$. MATLAB has a special routine called FFTSHIFT that does just that.

DFT and trigonometric interpolation

Let us consider an interpolation problem for a function $f(\theta)$ defined on $[0, 2\pi]$. Suppose we want to find a trigonometric polynomial

$$\sum_{k=-M/2}^{M/2-1} c_k q^{(k)}(\theta)$$

whose values coincide with values of $f(\theta)$ at the equispaced points $\theta_j = \frac{2\pi j}{M}, \quad j = 0, 1, 2, .., M-1$. In order to do this one forms a column vector

$$v = \begin{pmatrix} f(\theta_1) \\ f(\theta_2) \\ ... \\ f(\theta_{M-1}) \end{pmatrix}$$

and solves the linear system

$$Q_{\text{new}} \vec{c}_{\text{new}} = v,$$

whose solution is

$$\vec{c}_{\text{new}} = Q_{\text{new}}^* v.$$

In other words, DFT solves (exactly) the interpolation problem, and builds the trigonometric polynomial that equals v_j at the points θ_j.

It is interesting to understand the connection between the DFT and approximation of a function $f(\theta)$ by the Fourier series. In the latter case the approximating Fourier series can be written in the form

$$f(\theta) \approx \sum_{k=-M/2}^{M/2} a_k q^{(k)}(\theta),$$

where coefficients a_k are given by the formula

$$a_k = \frac{M}{2\pi} \int_0^{2\pi} \overline{q^{(k)}(\theta)} f(\theta) d\theta, \tag{2}$$

where the bar over $q^{(k)}$ represents complex conjugation. (The strange factor M appears in the above equation because we used scaled exponents $\frac{1}{\sqrt{M}} e^{ik\theta}$ instead of traditional $e^{ik\theta}$). In order to understand how a_k relates to c_k let us assume that one discretizes the integral (2) using the composite trapezoidal rule with nodes as the points $\theta_j = \frac{2\pi j}{M}, \quad j = 0, 1, 2, .., M$, and quadrature weights equal to the discretization step $2\pi/M$ at the internal points. Then

$$a_k \approx \frac{1}{2} \overline{q^{(k)}(0)} f(0) + \frac{1}{2} \overline{q^{(k)}(2\pi)} f(2\pi) + \sum_{j=1}^{M-1} \overline{q^{(k)}(\theta_j)} f(\theta_j).$$

Functions $q^{(k)}$ are periodic, so that $q^{(k)}(2\pi) = q^{(k)}(0)$. If, in addition, $f(\theta)$ is periodic then the above formula can be re-written as

$$a_k \approx \sum_{j=0}^{M-1} \overline{q^{(k)}(\theta_j)} f(\theta_j) = < q^{(k)}, v > = c_k.$$

Therefore, DFT can be used to approximately compute the coefficients of the Fourier series. Moreover, if $f(\theta)$ is periodic and smooth, such an approximation becomes extremely accurate (the smoother the function, the higher the order of accuracy of the trapezoid rule).