PY 421/621 - Advanced Computing in Physics

Lecture notes.

1 The Fourier transform.

1.1 Eigenvalues and eigenvectors of the circular shift operator and the finite Fourier transform.

Let us consider vectors in a space of dimension N. The circular shift operator (as implemented, for instance, in the Fortran 90 intrinsic function CSHIFT) maps the vector $f = (f_0, f_1, \dots f_{N-2}, f_{N-1})$ into $f' = (f_1, f_2, \dots f_{N-1}, f_0)$. The transformation $f \to f' = Cf$ is linear and its matrix representation has the form:

$$C = \begin{pmatrix} 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1 & 0 \\ 0 & 0 & 0 & \dots & 0 & 1 \\ 1 & 0 & 0 & \dots & 0 & 0 \end{pmatrix} \tag{1}$$

C is unitary, since, obviously, $f'^2 = f^2$. Its inverse $C^{-1} = C^\dagger$ permutes the elements of f in the opposite order:

$$C^{-1} = \begin{pmatrix} 0 & 0 & \dots & 0 & 0 & 1 \\ 1 & 0 & \dots & 0 & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 & 0 & 0 \\ 0 & 0 & \dots & 0 & 1 & 0 \end{pmatrix} \tag{2}$$

After N circular shifts, the components of the vector f return to the original order. Thus $C^N = I$. It follows that, given any eigenvalue λ of C, $\lambda^N = 1$:

λ must be one of the N^{th} roots of unity. The possible eigenvalues of C will therefore be

$$\lambda_k = e^{\frac{2\pi \imath k}{N}}, \qquad k = 0 \dots N-1 \tag{3}$$

It is convenient to define

$$z = e^{\frac{2\pi \imath}{N}} \tag{4}$$

With this notation

$$\lambda_k = z^k, \qquad k = 0 \dots N-1 \tag{5}$$

We will soon see that each of these eigenvalues occurs exactly once.

Starting from the possible eigenvalue λ_k, we can try solving for the components of the corresponding eigenvector $f^{(k)}$. The eigenvalue equation

$$Cf^{(k)} = \lambda_k f^{(k)} = z^k f^{(k)} \tag{6}$$

gives (cfr. Eq. 1)

$$\begin{array}{rcl} f_1^{(k)} & = & z^k f_0^{(k)} \\ f_2^{(k)} & = & z^k f_1^{(k)} \\ \dots & & \dots \\ f_{N-1}^{(k)} & = & z^k f_{N-2}^{(k)} \\ f_0^{(k)} & = & z^k f_{N-1}^{(k)} \end{array} \tag{7}$$

The first $N-1$ equations determine

$$f_j^{(k)} = z^{jk} f_0^{(k)} \tag{8}$$

The last equation

$$f_0^{(k)} = z^k f_{N-1}^{(k)} = z^{kN} f_0^{(k)} \tag{9}$$

is a consistency condition, which is satisfied because $z^{kN} = 1$. Thus we see that all $\lambda_k = z^k, \quad k = 0 \dots N-1$, are indeed eigenvalues of C. Since there are N of them, they exhaust all the eigenvalues and each occurs exactly once.

We fix the eigenvectors uniquely by

$$f_0^{(k)} = \frac{1}{\sqrt{N}} \tag{10}$$

This gives

$$f_j^{(k)} = \frac{1}{\sqrt{N}} e^{\frac{2\pi \imath jk}{N}} \tag{11}$$

for the eigenvector components and the normalization

$$|f^{(k)}| = \sqrt{\sum_j f_j^{(k)*} f_j^{(k)}} = 1 \tag{12}$$

Eigenvectors corresponding to different values of k are orthogonal. This follows from the general properties of eigenvectors of unitary matrices and can also easily be checked explicitly: for $k \neq k'$

$$f^{(k)} f^{(k')} = \sum_j f_j^{(k)*} f_j^{(k')} = \sum_j z^{-kj} z^{k'j} = \sum_j z^{(k'-k)j} = \frac{1 - z^{(k'-k)N}}{1 - z^{(k'-k)}} = 0 \tag{13}$$

The eigenvectors $f^{(k)}$ thus form an orthonormal basis. Any vector f can be expanded into this basis:

$$f = \sum_k F_k f^{(k)} \tag{14}$$

Explicitly

$$f_j = \sum_k F_k f_j^{(k)} = \frac{1}{\sqrt{N}} \sum_k F_k \, e^{\frac{2\pi \imath jk}{N}} \tag{15}$$

Since this is an orthonormal change of basis, the norm of the vector is preserved

$$\sum_j |f_j|^2 = \sum_k |F_k|^2 \tag{16}$$

The change of basis expressed by Eqs. 14, 15 goes under the name of (finite) Fourier transform. The numbers $F_k, \quad k = 0 \ldots N-1$ are the Fourier components of f.

1.2 Properties of the finite Fourier transform, and its continuum limits.

The Fourier components of f can be found by using the orthonormality of the eigenvectors:

$$F_k = f^{*(k)} f = \frac{1}{\sqrt{N}} \sum_j f_j \, e^{\frac{-2\pi \imath jk}{N}} \tag{17}$$

The action of the circular shift operator C takes a particularly simple form when expressed in terms of the Fourier components. From the fact that C is diagonal in the basis of its eigenvectors it follows that, if $f' = Cf$,

$$F'_k = \lambda_k F_k = e^{\frac{2\pi\imath k}{N}} F_k \tag{18}$$

This can also be seen directly from the expansion 14

$$f' = Cf = \sum_k F_k C f^{(k)} = \sum_k \lambda_k F_k f^{(k)} \tag{19}$$

which shows that $F'_k = \lambda_k F_k$, as in Eq. 18 above. The fact that C is diagonal in Fourier space is especially relevant because C enters in most numerical approximations to differential operators. For example, the central difference approximation to the second derivative is the operator

$$D_2 = \frac{C + C^\dagger - 2I}{a^2} \tag{20}$$

where we assumed that the numbers f_j represent the value of some function $f(x)$ at the points $x_j = x_0 + ja$. In Fourier space, D_2 is also diagonal

$$(D_2 F)_k = \frac{e^{\frac{2\pi\imath k}{N}} + e^{\frac{-2\pi\imath k}{N}} - 2}{a^2} F_k = \frac{2\cos\left(\frac{2\pi k}{N}\right) - 2}{a^2} F_k \tag{21}$$

For several considerations that follow, it will be convenient to extend the range of indices j, k etc. beyond $0, N-1$. We will assume that all indices are defined mod N, so that

$$f_{j+mN} = f_j, \qquad F_{k+mN} = F_k \tag{22}$$

with integer m. This convention is consistent with the relation between f_j and F_k (Eqs. 15, 17) since $\exp[2\pi\imath(j+mN)k/N] = \exp[2\pi\imath j(k+mN)/N] = \exp[2\pi\imath jk/N]$.

Given three vectors f, f', f'' we say that f'' is the convolution of f and f' if

$$f''_i = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} f_{i-j} f'_j \tag{23}$$

where we use the convention just introduced above for the case when the index $i-j$ takes negative values. The vector with components f_{i-j} can be obtained from f by the action of C^{-j}. Thus

$$f'' = \frac{1}{\sqrt{N}} \sum_j f'_j \, C^{-j} f \tag{24}$$

In Fourier space this becomes

$$F''_k = \frac{1}{\sqrt{N}} \sum_j f'_j \, e^{\frac{-2\pi \imath jk}{N}} F_k \tag{25}$$

which, on account of Eq. 17, gives

$$F''_k = F_k F'_k \tag{26}$$

This is the "convolution theorem" for Fourier transforms. It shows that the convolution operation becomes just a component-by-component product in Fourier space.

In some applications the vector f will have real components. Its Fourier components F_k, in general, will still be complex. The implication of having real f_j can be derived by taking the complex conjugate of Eq. 17

$$F^*_k = \sum_j f^*_j e^{\frac{2\pi \imath jk}{N}} = \sum_j f_j e^{\frac{-2\pi \imath j(-k)}{N}} = F_{-k} \tag{27}$$

Thus the reality of f implies the constraints $F^*_k = F_{-k}$ on the its Fourier components. These constraints are equivalent to N equations on the $2N$ real variables $\mathrm{Re}F_k$, $\mathrm{Im}F_k$, so the total number of independent real variables is preserved.

The Fourier transform can be extended to functions of a continuous variable. We will proceed in two steps. We will consider first functions $f(x)$ defined over the finite range $-L/2 \leq x < L/2$ and will later extend the range to the whole real axis by letting $L \to \infty$.

Let us divide the interval $-L/2 \leq x < L/2$ into an even number N of subintervals of width $a = L/N$. (For convenience, we will take N to be even, although this is not crucial.) We approximate the function $f(x)$ by the values f_j it takes at the points $x_j = ja$:

$$f_j = f(ja), \qquad j = -\frac{N}{2}, \ldots \frac{N}{2} - 1 \tag{28}$$

We take the Fourier transform of f, but allow for a different normalization

$$F_k = \frac{\alpha}{\sqrt{N}} \sum_j f_j \, e^{\frac{-2\pi\imath jk}{N}} \tag{29}$$

where α is a normalization factor which will be specified below. (Remember that indices can be thought of as defined mod$N$, so that, although we are letting now j vary from $-N/2$ to $(N/2)-1$, this formula only differs from Eq. 17 in the normalization.) We rewrite Eq. 29 in terms of x and $f(x)$

$$F_k = \frac{\alpha}{\sqrt{N}} \sum_j f(x_j) \, e^{\frac{-2\pi\imath x_j k}{L}} \tag{30}$$

We set now $\alpha = \sqrt{a}$. With this normalization, Eq. 30 becomes

$$F_k = \frac{\sqrt{a}}{\sqrt{N}} \sum_j f(x_j) \, e^{\frac{-2\pi\imath x_j k}{L}} = \frac{1}{\sqrt{L}} \sum_j a f(x_j) \, e^{\frac{-2\pi\imath x_j k}{L}} \tag{31}$$

We recognize now in the r.h.s. of this equation the numerical approximation to the integral of $f(x)\exp(-2\pi\imath kx/L)$. We take the continuum limit by letting $N \to \infty$, $a \to 0$ and obtain

$$F_k = \frac{1}{\sqrt{L}} \int_{-L/2}^{L/2} f(x) \, e^{\frac{-2\pi\imath kx}{L}} \, dx \tag{32}$$

This is the Fourier transform for functions defined over a finite range. It maps $f(x)$ into an infinite set of Fourier coefficients $F_k, \quad k = -\infty, \ldots \infty$. The transformation preserves the norm, in the sense that

$$\int_{-L/2}^{L/2} |f(x)|^2 \, dx = \sum_k |F_k|^2 \tag{33}$$

This follows from

$$\sum_k |F_k|^2 = a \sum_j |f_j|^2 \tag{34}$$

(cfr. Eq. 16) and motivated our choice of normalization. The inverse of the transformation 32 can be obtained from

$$f_j = \frac{1}{\sqrt{aN}} \sum_k F_k \, e^{\frac{2\pi\imath jk}{N}} = \frac{1}{\sqrt{L}} \sum_k F_k \, e^{\frac{2\pi\imath x_j k}{L}} \tag{35}$$

(cfr. Eq. 15 and remember the change of normalization). In the limit $N \to \infty, \quad a \to 0$ this becomes

$$f(x) = \frac{1}{\sqrt{L}} \sum_k F_k \, e^{\frac{2\pi\imath kx}{L}} \tag{36}$$

Notice that Eq. 36 defines a function $f(x)$ for all values of x, not necessarily restricted to the interval $-L/2 \leq x < L/2$. Indeed, $f(x)$, as defined by Eq. 36, is a periodic function of x with period L:

$$f(x + mL) = f(x) \tag{37}$$

with integer m. On account of such periodicity, we will drop explicit mention of the range in all integrals involving $f(x)$, assuming that the integral is over one full period of f.

The continuum equivalent of the circular shift operator is the infinitesimal displacement $f(x) \to f(x + dx)$. The transformation of f under such infinitesimal transformation is captured by the derivative and it is therefore no surprise that the derivative operator becomes diagonal in Fourier space. The Fourier components of $f'(x) = df(x)/dx$ are

$$F'_k = \frac{2\pi\imath k}{L} F_k \tag{38}$$

The finite shift $f(x) \to \tilde{f}(x) = f(x + b)$ is also diagonal in Fourier space, with

$$\tilde{F}_k = e^{\frac{2\pi\imath kb}{L}} F_k \tag{39}$$

and the convolution theorem becomes the following. If $f(x)$, $g(x)$ and $h(x)$ are related by

$$h(x) = \frac{1}{\sqrt{L}} \int f(x - y) \, g(y) \, dy \tag{40}$$

then their Fourier components satisfy

$$H_k = F_k G_k \tag{41}$$

Notice that the Fourier transformation defined by Eqs. 32, 36 goes both ways, in the sense that, as it associates to a function $f(x)$ the infinite set of Fourier coefficients F_k, so, given an infinite sequence F_k, it can be used to associate with it the periodic function $f(x)$. Sometime relations involving sequences can thus be converted into simpler relations on the functions that are their Fourier transforms. For example, the action of shift operator on the sequence, $F_k \to F'_k = F_{k+1}$, becomes diagonal on $f(x)$: $f(x) \to f'(x) = \exp(-2\pi\imath x/L) f(x)$. There is also the equivalent of the convolution theorem: if F_k, F'_k and F''_k are related by

$$F''_k = \frac{1}{\sqrt{L}} \sum_{k'} F_{k-k'}\, F'_{k'} \tag{42}$$

then the associated functions $f(x), f'(x)$ and $f''(x)$ are related by

$$f''(x) = f(x) f'(x) \tag{43}$$

Finally, we can let x vary over an infinite range by taking the limit $L \to \infty$. In order to preserve a meaningful exponent in Eq. 36 we introduce a new variable p related to k by

$$p_k = \frac{2\pi}{L} k \tag{44}$$

We also allow again for a change of normalization. We thus rewrite Eq. 36 as

$$f(x) = \frac{\beta}{\sqrt{L}} \sum_k F(p_k)\, e^{\imath p_k x} \tag{45}$$

We notice that the spacing between subsequent values p_k and p_{k+1} of p is $\Delta p = 2\pi/L$. This spacing goes to zero when L goes to infinity and thus, by choosing β so that $\beta/\sqrt{L}$ is proportional to Δp, it becomes possible to interpret the r.h.s. of Eq. 45 as the approximation to an integral. It is convenient to set $\beta = \sqrt{2\pi/L}$. Equation 45 becomes then

$$f(x) = \frac{1}{\sqrt{2\pi}} \sum_k F(p_k)\, e^{\imath p_k x}\, \Delta p \tag{46}$$

In the limit $L \to \infty$, $\Delta p \to 0$ the r.h.s. becomes an integral and the limit of Eq. 46 gives the Fourier transform over a continuum infinite range:

$$f(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} F(p)\, e^{\imath p x}\, dp \tag{47}$$

The inverse transform can be obtained from Eq. 32 with the appropriate change of variables and normalization

$$F(p_k) = \lim_{L\to\infty} \frac{1}{\beta\sqrt{L}} \int_{-L/2}^{L/2} f(x)\, e^{-\imath x p_k}\, dx \tag{48}$$

i.e.

$$F(p) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(x)\, e^{-\imath p x}\, dx \tag{49}$$

Notice the symmetry between Eqs. 47 and 49. Our particular choice for the factor β in Eq. 45 was motivated by the desire of getting the same normalization factor $1/\sqrt{2\pi}$ in Eqs. 47 and 49.

The infinite-range, continuous Fourier transform has many useful properties, similar to those encountered with the finite-range transform. It preserves the norm:

$$\int |F(p)|^2\, dp = \int |f(x)|^2\, dx \tag{50}$$

The derivative and shift operators are diagonal in Fourier space:

$$\begin{aligned} g(x) = f'(x) \qquad &\Rightarrow \qquad G(p) = \imath p F(p) \\ g(x) = f(x+b) \qquad &\Rightarrow \qquad G(p) = e^{\imath p b} F(p) \end{aligned} \tag{51}$$

The convolution theorem holds:

$$h(x) = \frac{1}{\sqrt{2\pi}} \int f(x-y) g(y)\, dy \quad \Rightarrow \quad H(p) = F(p)G(p) \tag{52}$$

To conclude, by performing the Fourier transform followed by its inverse, we get

$$f(x) = \frac{1}{2\pi} \int \Big[\int f(y)\, e^{-\imath p y}\, dy \Big]\, e^{\imath p x}\, dp = \int \Big[\frac{1}{2\pi} \int e^{\imath p (x-y)}\, dp \Big] f(y)\, dy \tag{53}$$

This implies

$$\frac{1}{2\pi} \int_{-\infty}^{\infty} e^{\imath p (x-y)}\, dp = \delta(x-y) \tag{54}$$

This equation, and its finite-range and discrete counterparts

$$\frac{1}{L} \sum_{k=-\infty}^{\infty} e^{\frac{2\pi \imath k (x-y)}{L}} = \sum_{m=-\infty}^{\infty} \delta(x - y - mL) \tag{55}$$

$$\frac{1}{L}\int_{-L/2}^{L/2} e^{\frac{2\pi\imath(k-k')x}{L}} = \delta_{k,k'} \tag{56}$$

$$\frac{1}{N}\sum_{k=0}^{N-1} e^{\frac{2\pi\imath k(j-j')}{N}} = \sum_{m=-\infty}^{\infty} \delta_{j,j'+mN} \tag{57}$$

express the completeness of Fourier space and are quite useful to remember.

1.3 The fast Fourier transform (FFT) algorithm.

The fast Fourier transform (FFT) is an algorithm, discovered by Danielson and Lanczos in 1942, that allows one to calculate the finite Fourier transform (see Eq. 15)

$$f_j = \sum_{k=0}^{N-1} e^{2\pi\imath\frac{jk}{N}} F_k$$

where $N = 2^p$ is a power of 2, in $O(N\log_2 N)$ rather than $O(N^2)$ steps. (The FFT can be generalized to the case where N is not a power of 2, but its implementation is simpler if N is a power of 2 and we will restrict our considerations to this case.)

The algorithm is based on the identity

$$\begin{aligned} f_j = \sum_{k=0}^{N-1} e^{2\pi\imath\frac{jk}{N}} F_k &= \sum_{k=0}^{\frac{N}{2}-1} e^{2\pi\imath\frac{j(2k)}{N}} F_{2k} + \sum_{k=0}^{\frac{N}{2}-1} e^{2\pi\imath\frac{j(2k+1)}{N}} F_{2k+1} \\ &= \sum_{k=0}^{\frac{N}{2}-1} e^{2\pi\imath\frac{jk}{N/2}} F_{2k} + e^{2\pi\imath\frac{j}{N}} \sum_{k=0}^{\frac{N}{2}-1} e^{2\pi\imath\frac{jk}{N/2}} F_{2k+1} \end{aligned} \tag{58}$$

We see from this equation that the components of the Fourier transform $f_j,\ j = 0,\dots N-1$ can be calculated in $O(N)$ steps if we already know the Fourier transforms $f_j^0,\ f_j^1,\ j = 0\dots\frac{N}{2}-1$ taken separately over the even and odd sites. Clearly this gives origin to a recursive procedure, since the same argument can now be applied to f_j^0 and f_j^1 separately. Repeating the procedure $p = \log_2 N$ times, we will eventually get to the calculation of N Fourier transforms over a single site, which is of course trivial since the Fourier transform coincides then with the variable itself. Since at every iteration of the recursive procedure one must perform $O(N)$ operations (if one is at the stage where one deals with $M = 2^q$ Fourier transforms taken over $N/M = 2^{p-q}$ points, one must implement M equations similar to Eq. 58 above, involving $O(N/M)$ operations each), the total number of operations is indeed $O(N \log_2 N)$.

The implementation of equations like Eq. 58 is rather straightforward, but one must pay attention to the organization of the data. It is easier to explain the issue with a specific example, so let us consider the case where $N = 8$. According to Eq. 58, in order to calculate the Fourier transform of the variables F_0, $F_1, \ldots F_7$ we need the Fourier transforms f_j^0 and f_j^1, $k = 0 \ldots 3$, taken over the variables F_0, F_2, F_4, F_6 and F_1, F_3, F_5, F_7 respectively. So, to begin with, it makes sense to rearrange the data so that the F_k values follow in the order F_0, F_2, F_4, F_6, F_1, F_3, F_5, F_7. We will then have to calculate the Fourier transforms of the first 4 values in the array and of the last 4 values separately. But now the problem of calculating the Fourier transform of the variables F_0, F_2, F_4, F_6, always according to Eq. 58 which we implement in a recursive manner, can be reduced to the problem of calculating the Fourier transforms over the variables F_0, F_4 and F_2, F_6 separately. So we rearrange the first 4 elements of data in the order F_0, F_4, F_2, F_6. A similar reasoning applied to the last 4 elements of data leads us to rearrange them in the order F_1, F_5, F_3, F_7 so that the entire array of data becomes

$$F_0,\ F_4,\ F_2,\ F_6\, F_1,\ F_5,\ F_3,\ F_7 \tag{59}$$

Now we will have to take the Fourier transforms over the first pair of elements of the array, then the second, third and fourth pairs separately. Once again, we can use the recursive procedure to reexpress, for instance, the Fourier transform of the pair F_0, F_4 in terms of the Fourier transforms of F_0 and F_4. Of course, since we have finally arrived at sets formed by single elements, the Fourier transforms coincide with the elements themselves. So we do not need to perform any further rearrangements but, starting from the data in the order 59, we simply combine the subsequent pairs of data according to Eq. 58 to obtain their Fourier transforms, which we leave at the corresponding positions in the entire array. Then we combine the data in the subsequent groups of 4, always according to Eq. 58, to obtain the Fourier transforms of the various groups of 4 elements. Then, in a recursive manner, we combine the data in the subsequent groups of 8, etc. In our particular case, we have only one group of 8 elements, and so with the latter (third) application of

Eq. 58 the procedure stops and we find that the 8 elements of the array contain the Fourier transforms f_0, $f_1, \ldots f_7$ of our original data.

It should be apparent from this discussion that the non trivial algorithmic step is to rearrange the original data in the order 59 above, not in the sense that this is the step involving most computations, but the one where the rule to follow is not immediately apparent. However we get a clue on the rule underlying the rearrangement of data if we rewrite the sequences of indices in the original and final orders, namely 0, 1, 2, 3, 4, 5, 6, 7 and 0, 4, 2, 6, 1, 5, 3, 7, in binary notation. This leads to

$$000 \quad 001 \quad 010 \quad 011 \quad 100 \quad 101 \quad 110 \quad 111 \tag{60}$$

and

$$000 \quad 100 \quad 010 \quad 110 \quad 001 \quad 101 \quad 011 \quad 111 \tag{61}$$

But now we see that the bits in the corresponding elements of the two arrays are simply reversed! Thus, the initial step, which plays a crucial part in the implementation of the algorithm, consists simply in rearranging the elements of the array with indices in bit reversed order. One can readily convince oneself that this is true for any number N (equal to a power of two) of elements in the original array. Thus the FFT algorithm is seen to consist of two steps, the rearrangement of the data in bit reversed order, followed by $\log_2 N$ recombinations of the data according to Eq. 58. The latter step is the computationally intensive one, with $O(N \log_2 N)$ arithmetic operations, but the data motion of the first step is the one that gives origin to interesting issues of efficient implementation with any distributed memory machine.