Notes regarding Integral and Discrete Fourier Transforms

C. J. Rennie

21st October 2010

1 Introduction

While DFTs are widely used to approximate integral transforms, there is a lot of scope for confusion. There are at least six sources of confusion, in my experience:

- convention: there is no standard for normalization and the sign of the exponent in transformations; nor is there a clear preference for frequencies to be expressed as angular (ω) or as Hertz (f),
- notation: transforms may be represented as $x(\omega)$, X_i, $\hat{y}(f)$, and so on,
- units: the DFT is agnostic regarding units, so there is an extra step involved, which demands care,
- assumptions: is the sampled time series considered to be periodic or aperiodic? Do we want to report *energy* or *power*?
- DFT $\leftrightarrow$ IFT: to give results generality it is best to report them as if no digitization was involved. Is the mapping being done correctly between domains, with respect both to scaling and units?
- implementations: Matlab, IDL, R, and Mathematica all do it their own way.

(Aside from these, there are more mundane matters, like the side-effects of windowing and zero-padding, baseline removal, and being vague about one- or two-sided spectra. They are not discussed here.)

This documents tries to deal with these many sources of confusion. It aims to do it by (a) choosing one convention and notation, (b) being explicit about assumptions, (c) being precise about expected values and units, and (d) being precise about the correspondence between integral and discrete Fourier transforms. The choice of convention and notation is based on that in Proakis and Manolakis [2007], which also provides crucial derivations.

The following sections restate the important relations, and apply them to three special cases:

- delta function
- sine wave
- white noise

The final section contains numerical examples. Throughout we assume the time series to has units of Volts.

2 Basic relations

2.1 Integral Fourier Transform, IFT

The location of the normalization factor ($1/2\pi$) and sign of exponent are poorly standardized. Here is one common convention:

	Angular	Hertz
Forward	$X(\omega)=\int_{-\infty}^{\infty}x(t)e^{-i\omega t}dt \quad (1)$	$X(f)=\int_{-\infty}^{\infty}x(t)e^{-2\pi ift}dt \quad (2)$
Inverse	$x(t)=\frac{1}{2\pi}\int_{-\infty}^{\infty}X(\omega)e^{i\omega t}d\omega \quad (3)$	$x(t)=\int_{-\infty}^{\infty}X(f)e^{2\pi ift}df \quad (4)$
	Ref: P+M 4.1.31 and 4.1.32 (p237)	Ref: P+M 4.1.24 (p235) and 4.1.28 (p236)

Note that the angular and Hertz forms of the forward transform, $X(\omega)$ and $X(f)$, could be written as $X_\omega(\omega)$ and $X_f(f)$, respectively, to disambiguate these two functions; however there should be no confusion in practice. The two functions are closely related: $X_\omega(\omega)=X_\omega(2\pi f)=X_f(f)=X_f(\omega/2\pi)$.

For continuous, aperiodic functions, Parseval's Theorem refers to *energy*:

$$\text{Energy}[V^2s]=\int_{-\infty}^{\infty}|x(t)|^2dt=\int_{-\infty}^{\infty}|X(\omega)|^2d\omega=\int_{-\infty}^{\infty}|X(f)|^2df. \tag{5}$$

Ref: P+M 4.1.38 (p238). The alternative formulation in terms of *power* involves dividing this by some period T, which gives a quantity in units of V^2.

2.2 Discrete Fourier Transform, DFT

The location of the normalization factor ($1/N$) and sign of exponent are poorly standardized. Here is what follows from the form of the IFT chosen above.

	Frequency indexed by k; time by n
Forward	$c_k=\frac{1}{N}\sum_{n=0}^{N-1}x(n)e^{-2\pi ikn/N} \quad (6)$
Inverse	$x(n)=\sum_{k=0}^{N-1}c_ke^{2\pi ikn/N} \quad (7)$
	Ref: P+M 4.2.7 and 4.2.8 (p242)

The Fourier coefficients, c_k, define the amplitude and phase for the corresponding basis functions

$$s_k(n)=e^{2\pi ikn/N}=e^{2\pi ik(n\Delta t)/(N\Delta t)}=e^{2\pi i(k\Delta f)(n\Delta t)}. \tag{8}$$

For discrete, periodic functions, Parseval's Theorem refers to energy/period, or *power*:

$$\text{Power}[V^2]=\frac{1}{N}\sum_{n=0}^{N-1}|x(n)|^2=\sum_{k=0}^{N-1}|c_k|^2. \tag{9}$$

Ref: P+M 4.2.11 (p246). Note the unexpected factor $1/N$. However Eq. (9) may be multiplied by $N\Delta t=T$ to give

$$\text{Energy}[V^2s]=\sum_{n=0}^{N-1}|x(n)|^2\Delta t=\sum_{k=0}^{N-1}\left|\frac{c_k}{\Delta f}\right|^2\Delta f, \tag{10}$$

while makes the correspondence with the energy equation, Eq. (5), somewhat clearer. Then, letting the duration of the period $\to\infty$ (i.e. $\Delta f\to df$), and $\Delta t\to dt$, further clarifies the link. [Note also the implication that $X(\omega)=X(f)\leftrightarrow c_k/\Delta f=c_kT$, which is borne out in what follows.]

2.3 Link between continuous-aperiodic and discrete-periodic cases

Starting from the continuous-aperiodic time domain, one can move towards the discrete-periodic case in either of two ways.

By imposing periodicity – which makes f discrete, with $\Delta f=1/T$ – in which case

	Transform	Notes
Forward	$c_k = \frac{1}{T}\int_T x(t)e^{-2\pi ikt/T}dt \qquad (11)$	$f = k\Delta f = k/T$
Inverse	$x(n) = x(t) = \sum_{k=-\infty}^{\infty} c_k e^{2\pi ikt/T} \qquad (12)$	$t = n\Delta t = nT/N$
	Ref: P+M 4.1.8 and 4.1.9 (p229)	

For continuous-periodic signals, Parseval's Theorem refers to energy/period, or *power*:

$$\text{Power}[V^2] = \frac{1}{T}\int_T |x(t)|^2 dt = \sum_{k=-\infty}^{\infty} |c_k|^2. \tag{13}$$

Ref: P+M 4.1.14 (p230)

By imposing discreteness in time – which makes f periodic, with period=sampling rate $F_s = 1/\Delta t$ – in which case

	Transform	Notes
Forward	$X(f) = \sum_{n=-\infty}^{\infty} x(n)e^{-2\pi in\Delta t f} \qquad (14)$	$F_s = 1/\Delta t$
Inverse	$x(n) = \frac{1}{F_s}\int_{F_s} X(f)e^{2\pi in\Delta t f}df \qquad (15)$	$F_s = 1/\Delta t$
	Ref: P+M 4.2.28 and 4.2.29 (p251)	

For discrete-aperiodic signals, Parseval's Theorem refers to *energy*:

$$\text{Energy}[V^2] = \sum_{n=-\infty}^{\infty} |x(n)|^2 = \frac{1}{F_s}\int_{F_s} |X(f)|^2 df. \tag{16}$$

Ref: P+M 4.2.41 (p255)

Chapter 6 of Proakis and Manolakis [2007] provides additional material regarding sampling and reconstruction, and Chapter 14 is devoted to power spectrum estimation.

3 Delta function, area=$A\Delta t$

Continuous, aperiodic version:

	Function	Dimension
Time	$x(t) = \frac{A\Delta t}{\lvert\Delta t\rvert}\delta\left(\frac{t-t_0}{\Delta t}\right)$	V
Angular	$X(\omega) = A\Delta t\ e^{-i\omega t_0}$	V s
Hz	$X(f) = A\Delta t\ e^{-2\pi i f t_0}$	V s
Checks	$\int x(t)dt = A\Delta t$	V s
	$\int \lvert x(t)\rvert^2 dt = A^2\Delta t$	V^2 s
	$\int \lvert X(\omega)\rvert^2 d\omega = \int \lvert X(f)\rvert^2 df = A^2\Delta t$	V^2 s

Discrete, periodic version:

	Function	Dimension
Time	$x(n) = A$ at $t = t_0$, zero elsewhere	V
Freq	$c_k = \frac{A}{N}e^{-2\pi ikt_0/T}$	V
Checks	$\Delta t\sum x(n) = A\Delta t$	V s
	$(1/N)\sum \lvert x(n)\rvert^2 = A^2/N$	V^2
	$\sum \lvert c_k\rvert^2 = A^2/N$	V^2

To obtain c_k, Eq. (11) was applied to $x(t)$. Then $x(n)$ was obtained using Eq. (12).

4 Sine wave

Continuous, aperiodic version:

	Function	Dimension
Time	$x(t) = A\cos(\omega_0 t) = A\sin(2\pi f_o t)$	V
Angular	$X(\omega) = \frac{A}{2}[\delta(\omega - \omega_0) + \delta^*(\omega + \omega_0)]$	V s
Hz	$X(f) = \frac{A}{2}[\delta(f - f_0) + \delta^*(f + f_0)]$	V s
Checks	$\frac{1}{T}\int_{T/2}^{T/2} \lvert x(t)\rvert^2 dt = A^2/2$	V^2
	$\frac{1}{T}\int_{-\infty}^{\infty} \lvert X(\omega)\rvert^2 d\omega = \frac{1}{T}\int_{-\infty}^{\infty} \lvert X(f)\rvert^2 df = A^2/2???$	V^2

[Evaluating $\int |X(f)|^2 df$ requires windowing the signal, and then letting the window width $T \to \infty$.]

Discrete, periodic version:

	Function	Dimension
Time	$x(n) = A\cos(2\pi f_0 nT/N)$	V
Freq	$c_k = \frac{A}{2}e^{-2\pi ikb/T}$	V
Checks	$(1/N)\sum \lvert x(n)\rvert^2 = A^2/2$	V^2
	$\sum \lvert c_k\rvert^2 = A^2/2$	V^2

To obtain c_k, Eq. (11) was applied to $x(t)$. Then $x(n)$ was obtained using Eq. (12).

5 White noise

Continuous, aperiodic version:

	Function	Dimension
Time	$x(t) = \text{white noise, with SD} = \sigma$	V
	$\gamma_{xx}(\tau) = \int_{-\infty}^{\infty} \Gamma_{xx}(f) e^{2\pi i f\tau} df$	V^2
Angular	$X(\omega) = \text{NA}$	V s
	$\Gamma_{xx}(\omega) = \int_{-\infty}^{\infty} \gamma_{xx}(\tau) e^{i\omega\tau} d\tau$	V^2 s
Hz	$X(f) = \text{NA}$	V s
	$\Gamma_{xx}(f) = \int_{-\infty}^{\infty} \gamma_{xx}(\tau) e^{-2\pi i f\tau} d\tau$	V^2 Hz^{-1}
Checks	$E(x^2(t)) = \frac{1}{T}\int_{T/2}^{T/2} \lvert x(t)\rvert^2 dt = \sigma^2$	V^2
	$\int_{-\infty}^{\infty} \Gamma_{xx}(\omega) d\omega = \int_{-\infty}^{\infty} \Gamma_{xx}(f) df = \gamma_{xx}(0) = \sigma^2$	V^2

A stationary random process is an infinite-energy signal and hence its Fourier transform does not exist. Its spectral characteristic is obtained according to the Wiener-Khintchine theorem, by computing the Fourier transform, $\Gamma_{xx}(f)$, of the autocorrelation function, $\gamma_{xx}(\tau)$. Ref: P+M 12.1.5 (p828).

Discrete, periodic version:

	Function	Dimension
Time	$x(n) = \texttt{rangau(0,}\sigma\texttt{)}$	V
Freq	$\lvert c_k\rvert = \sigma$, Arg random	V
Checks	$(1/N)\sum \lvert x(n)\rvert^2 = \sigma^2$	V^2
	$\sum \lvert c_k\rvert^2 = \sigma^2$	V^2

6 Explicit numerical tests

The following tests are all based on one time series covering $T = 4.0$ seconds, and with $N = 2000$ points.

6.1 Delta function at time $t_0 = 0.2$ s, with area $\int x\,dt = 3\Delta t$ V s

1. The implied offset within the time series is $t_0/(T/N) = 100$
2. The implied value at this offset is $(\int x\,dt)/(T/N) = 3$ V
3. The forward DFT should have $|c_k| = A/N = 0.0015$ V, for all k
4. The forward DFT should have $\arg(c_k)$ advancing by $-2\pi t_0/T = -0.31415$ rad/bin
5. The 1-sided SPD $|c_k|^2 + |c_{-k}|^2$ should have mean $2(A/N)^2 = 4.5 \times 10^{-6}$ V^2/bin
6. Power in time series: $(1/N)\sum |x_n|^2 = A^2/N = 0.0045$ V^2
7. Power in spectrum: $\sum |c_k|^2 = A^2/N = 0.0045$ V^2
8. The inverse DFT of the forward DFT should match the original time series

6.2 Sine wave, with amplitude $A = 3$ V, and frequency $f_0 = 10$ Hz

1. The implied time series is $x_n = A\cos(2\pi f_0 n(T/N) + \phi)$, where the phase ϕ is arbitrary
2. The forward DFT should contain $|c_k| = A/2 = 1.5$ V at offsets $k = \pm f_0 T = \pm 40$, 0 otherwise
3. The 1-sided SPD $|c_k|^2 + |c_{-k}|^2$ should be $A^2/2 = 4.5$ V^2/bin at f_0, and 0 otherwise
4. Power in time series: $(1/N)\sum |x_n|^2 = A^2/2 = 4.5$ V^2
5. Power in spectrum: $\sum |c_k|^2 = A^2/2 = 4.5$ V^2
6. The inverse DFT of the forward DFT should match the original time series

6.3 White noise, with zero mean and standard deviation $A = 3$ V

1. The implied time series is $x_n = \texttt{rangau}(0, A)$, where the distribution function is arbitrary
2. The forward DFT should contain $|c_k| \approx A/\sqrt{N} = 0.067$ V (average over all k)
3. The 1-sided SPD $|c_k|^2 + |c_{-k}|^2$ should have mean $2A^2/N = 0.009$ V^2/bin
4. Power in time series: $(1/N)\sum |x_n|^2 = A^2 = 9.0$ V^2
5. Power in spectrum: $\sum |c_k|^2 = A^2 = 9.0$ V^2
6. The inverse DFT of the forward DFT should match the original time series

6.4 Reporting results

When we have a periodic signal (and this is always implicitly assumed when performing DFTs) we have seen that the spectral power is $\sum |c_k|^2$, which has units of V^2. (For aperiodic signals it only makes sense to measure *energy*.)

The spectral power *density* could then be legitimately plotted as a bar graph, with heights $|c_k|^2$, and any band power evaluated by summing the appropriate bars. That plot would be of the spectral power density (per bin), and has units V^2. However we tend to think in terms of *integrals* – this is what is done for continuous spectra, and estimation of band power is easily done by eye by judging the average height and multiplying by the band width in Hz. Accordingly, we generally plot $|c_k|^2/\Delta f$ (instead of $|c_k|^2$) versus frequency. This plot represents spectral power density (per Hertz), and has units $\mathrm{V}^2\ \mathrm{Hz}^{-1}$. [Dividing by Δf is the same as multiplying by T; so we could label the result 'Energy (per period T) density'. But we don't.]

That is one rationale for dividing $|c_k|^2$ by Δf. But perhaps the corollary is more compelling. Plots of spectral power densities then become *standardized*. Given a wide bandwidth signal with spectral power density equal to 10 $\mathrm{V}^2\ \mathrm{Hz}^{-1}$, say, around some given frequency – then this SPD value remains the same whatever your choice of window length T. Doubling the window length doubles the 'Energy (per period T) density'; but now this energy is distributed amongst twice as many frequency bins. So your plots have a consistent height of 10 $\mathrm{V}^2\ \mathrm{Hz}^{-1}$, whatever your window length. *Plot heights are also independent of sampling rate.* From (Eq. 10) we see that doubling the sampling rate has no effect on energy (per period T): there are twice as many terms in the sum, but Δt is halved. The last bit of Sec. 2.2 reinforces this point.

So remember to plot the quantity $|c_k|^2/\Delta f$, call it 'Spectral power density', and remember its units are $\mathrm{V}^2\ \mathrm{Hz}^{-1}$. And the integral over some range of frequencies is 'Power', has units V^2, and may be computed by $\sum(|c_k|^2/\Delta f) \times \Delta f$ or $\sum |c_k|^2$ (whichever you like).

One more tweak: the spectral power density function spans both positive and negative frequencies; however it is symmetrical ($|c_k|^2 = |c_{-k}|^2$), so there is no need to show negative frequencies – and there is a temptation simply not to plot the left half of the spectrum. But rather than ignore power at negative frequencies, you should fold negative-frequency powers to positive frequencies, thus ensuring all components of power are explicit and properly accounted for. The result is a *one-sided* spectral power density function that has twice the height of the two-sided spectrum and $1/2$ the number of points.

In summary, after performing the DFT and obtaining c_k you should plot the values $(|c_k|^2 + |c_{-k}|^2)/\Delta f$, label the plot as having units $\mathrm{V}^2\ \mathrm{Hz}^{-1}$, and refer to it as a one-sided spectral power density plot. The power within a frequency band should then have units V^2.

Appendix A Listing of fourier.R

Below is a script that generates discrete approximations to each of the three test cases (delta, sine, white noise), performs a DFT, and checks the results against the expected values.

```
#! /usr/local/bin/R --vanilla
# Run from within R by typing          source("fourierTests.R")
# or from the Linux command line       Rscript fourierTests.R

# This program evaluates and checks the discrete Fourier transforms
# of several standard functions.  The emphasis is on correctness of
# scaling, self-consistancy, and clear correspondence with the case
# of Fourier integrals.

# The convention assumed here for continuous aperiodic signals is:
#   X(f) = Int_inf x(t) exp(-2 pi i f t) dt
#   x(t) = Int_inf X(f) exp( 2 pi i f t) dt
#   Energy = Int_inf |x(t)|^2 dt = Int_inf |X(f)|^2
# This corresponds to the discrete periodic signals:
#   c_k = (1/N) Sum_N x_n exp(-2 pi i k n/N)
#   x_n =       Sum_N c_n exp( 2 pi i k n/N)
#   Power = (1/N) Sum_N |x_n|^2 = Sum_N |c_k|^2
# The correspondence is proved in "Digital Signal Processing, 4th ed."
# by Proakis and Manolakis, pp 227 -- 245.

##################################################################
# This convention is implemented in R by
#     c <- (1/N) * fft(x)
#     x <- fft(c, inverse=T)
# SO omitting the factor (1/N) is always a danger with R!!!
```

```
##############################################################

# Common parameters.  Changing these will cause some results to
# change, but should not cause any tests to fail.
bigT <- 4.0               # T = duration of epoch, in seconds
bigN <- round(bigT*500) # T*samplingRate = values/epoch
time <- (bigT/bigN)*seq(0,bigN-1)
if(bigN %% 2 == 1) exit() # for simplicity ensure N is even

# Delta function
lagSecs <- 0.2                       # seconds
lagN <- 1 + lagSecs/(bigT/bigN)      # index
ampli = 3.0                          # Int x(t) dt = ampli * Delta t
x <- rep(0, bigN)
x[lagN] <- ampli                     # one non-zero element, with value=<ampli>
# Do FFT
c <- (1/bigN)*fft(x)
xPrime <- fft(c, inverse=T)
temp <- Mod(c)^2        # one-sided SPD, in uV^2/bin
spd <- c(temp[1:(bigN/2)],0) + c(0,rev(temp[(bigN/2+1):bigN]))
# Check FFT
cMag <- mean(abs(c))
cArg <- Arg(c)
cat("\nDelta function - alternative version:\n")
cat("Time domain contains",ampli,"V at t =",lagSecs,"s\n")
cat("Freq domain contains phasor with magnitude",cMag,"V: expecting",
    ampli/bigN,"\n")
cat("Freq domain contains phasor advancing by",round(cArg[2]-cArg[1],3),
    "rad/step: expecting",round(-2*pi*lagSecs/bigT,3),"\n")
cat("1-sided SPD has mean",mean(spd),"V^2/bin:",
    "expecting",2*(ampli/bigN)^2,"V^2/bin\n")
cat("Time domain power =",(1/bigN)*sum(Mod(x)^2),"V^2\n")
cat("Freq domain power =",sum(Mod(c)^2),"V^2\n")
cat("  Expecting power =",bigN*(ampli/bigN)^2,"V^2\n")
cat("Mean and SD of error in inverse =",mean(Re(xPrime)-x),"and",
    sd(Re(xPrime)-x),"V\n")

# Cosine function
ampli = 3.0           # V
f0 <- 10.0            # Hz
phase <- 0.3
x <- ampli * cos(2*pi*f0*time+phase)
# Do FFT
c <- (1/bigN)*fft(x)
xPrime <- fft(c, inverse=T)
temp <- Mod(c)^2        # one-sided SPD, in uV^2/bin
spd <- c(temp[1:(bigN/2)],0) + c(0,rev(temp[(bigN/2+1):bigN]))
# Check
cat("\nCosine function:\n")
cat("Time domain contains",ampli,"V x cos(2 pi",f0,"t +",phase,")\n")
cat("Freq domain contains zeros except",Mod(c[1+round(f0*bigT)]),"V at",f0,
    "Hz: expecting",ampli/2,"V\n")
cat("1-sided SPD contains zeros except",Mod(spd[1+round(f0*bigT)]),"V^2/bin at",f0,
    "Hz: expecting",ampli*ampli/2,"V^2/bin\n")
cat("Time domain power =",(1/bigN)*sum(Mod(x)^2),"V^2\n")
cat("Freq domain power =",sum(Mod(c)^2),"V^2\n")
cat("  Expecting power =",(1/2)*ampli^2,"V^2\n")
cat("Mean and SD of error in inverse =",mean(Re(xPrime)-x),"and",
    sd(Re(xPrime)-x),"V\n")

# White noise
```

```
sd <- 3.0            # V
x <- rnorm(bigN, mean=0, sd=sd)
# Do FFT
c <- (1/bigN)*fft(x)
xPrime <- fft(c, inverse=T)
temp <- Mod(c)^2         # one-sided SPD, in uV^2/bin
spd <- c(temp[1:(bigN/2)],0) + c(0,rev(temp[(bigN/2+1):bigN]))
# Check FFT
cMag <- mean(abs(c))
# Check
cat("\nWhite noise:\n")
cat("Time domain contains normally-distributed noise with sd =",sd,"V\n")
cat("Freq domain contains values with magnitude",cMag,"V: expecting",
    ampli/sqrt(bigN),"\n")
cat("1-sided SPD has mean",mean(spd),"V^2/bin:",
    "expecting",ampli*ampli/(bigN/2),"V^2/bin\n")
cat("Time domain power =",(1/bigN)*sum(Mod(x)^2),"V^2\n")
cat("Freq domain power =",sum(Mod(c)^2),"V^2\n")
cat("  Expecting power =",sd^2,"V^2\n")
cat("Mean and SD of error in inverse =",mean(Re(xPrime)-x),"and",
    sd(Re(xPrime)-x),"V\n")

# Example of output ...........................................

#Delta function:
#Time domain contains 3 V at t = 0.2 s
#Freq domain contains phasor with magnitude 0.0015 V: expecting 0.0015
#Freq domain contains phasor advancing by -0.314 rad/step: expecting -0.314
#1-sided SPD has mean 4.495504e-06 V^2/bin: expecting 4.5e-06 V^2/bin
#Time domain power = 0.0045 V^2
#Freq domain power = 0.0045 V^2
#  Expecting power = 0.0045 V^2
#Mean and SD of error in inverse = 6.52256e-19 and 2.383765e-17 V
#
#Cosine function:
#Time domain contains 3 V x cos(2 pi 10 t + 0.3 )
#Freq domain contains zeros except 1.5 V at 10 Hz: expecting 1.5 V
#1-sided SPD contains zeros except 4.5 V^2/bin at 10 Hz: expecting 4.5 V^2/bin
#Time domain power = 4.5 V^2
#Freq domain power = 4.5 V^2
#  Expecting power = 4.5 V^2
#Mean and SD of error in inverse = 8.399531e-18 and 7.658735e-16 V
#
#White noise:
#Time domain contains normally-distributed noise with sd = 3 V
#Freq domain contains values with magnitude 0.05877766 V: expecting 0.06708204
#1-sided SPD has mean 0.008780943 V^2/bin: expecting 0.009 V^2/bin
#Time domain power = 8.789724 V^2
#Freq domain power = 8.789724 V^2
#  Expecting power = 9 V^2
#Mean and SD of error in inverse = 5.679918e-18 and 1.921692e-15 V
```

References

John G. Proakis and Dimitris G. Manolakis. *Digital Signal Processing*. Pearson Prentice Hall, 4th edition, 2007.