1 Lomb's Algorithm and the Hilbert Transform

1.1 Introduction

This first chapter describes two of the more advanced techniques in signal processing: Lomb's algorithm and the Hilbert transform. Throughout this chapter (and the remainder of this text) we assume that you have a basic understanding of signal processing procedures; for those needing to refresh these skills, we include multiple references to van Drongelen (2007).

In the 1970s, the astrophysicist Lomb developed an algorithm for spectral analysis to deal with signals consisting of unevenly sampled data. You might comment that in astrophysics considering uneven sampling is highly relevant (you cannot observe the stars on a cloudy day), but in neuroscience data are always evenly sampled. Although this is true, one can consider the action potential (or its extracellular recorded equivalent, the spike) or neuronal burst as events that represent or sample an underlying continuous process. Since these events occur unevenly, the sampling of the underlying process is also uneven. In this context we will explore how to obtain spectral information from unevenly distributed events.

The second part of this chapter introduces the Hilbert transform that allows one to compute the instantaneous phase and amplitude of a signal. The fact that one can determine these two metrics in an instantaneous fashion is unique because usually this type of parameter can only be associated with an interval of the signal. For example, in spectral analysis the spectrum is computed for an epoch and the spectral resolution is determined by epoch length. Being able to determine parameters such as the phase instantaneously is especially useful if one wants to determine relationships between multiple signals generated within a neuronal network.

1.2 Unevenly Sampled Data

In most measurements we have evenly sampled data—for instance, the interval Δt between the sample points of the time series is constant, pixels in a picture have uniform interdistance, and so forth. Usually this is the case, but there are instances when uneven sampling cannot be avoided. Spike trains (chapter 14, van Drongelen,

Signal Processing for Neuroscientists, A Companion Volume. DOI: 10.1016/B978-0-12-384915-1.00001-2

2007) or time series representing heart rate (van Drongelen et al., 2009) are two such examples; in these cases one may consider the spike or the heartbeat to represent events that sample an underlying process that is invisible to the experimenter (Fig. 1.1A).

The heart rate signal is usually determined by measuring the intervals between peaks in the QRS complexes. The inverse value of the interval between pairs of subsequent QRS complexes can be considered a measure of the instantaneous rate (Fig. 1.1B). This rate value can be positioned in a time series at the instant of either the first or second QRS complex of the pair and, because the heartbeats do occur at slightly irregular intervals, the time series is sampled unevenly. This example for the heartbeat could be repeated, in a similar fashion, for determining the firing rate associated with a spike train.

When a signal is unevenly sampled, many algorithms that are based on a fixed sample interval (such as the direct Fourier transform [DFT] or fast Fourier transform [FFT]) cannot be applied. In principle there are several solutions to this problem:

(1) An evenly sampled time series can be constructed from the unevenly sampled one by using interpolation. In this approach the original signal is resampled at evenly spaced intervals. The interpolation technique (e.g., linear, cubic, spline) may vary with the application. In MATLAB resampling may be accomplished with the interp1 command or any of the other related functions. After resampling the time series one can use standard Fourier analysis methods. The disadvantage is that the interpolation algorithm may introduce frequency components that are not related to the underlying process.

(2) The measurements can be represented as the number of events in a binned trace; now our time series is a sequence of numbers, with one number for each bin. Since the bins are equally spaced, the standard DFT/FFT can be applied. In case of low-frequency activity, the bins must be relatively wide to avoid an overrepresentation of empty bins.

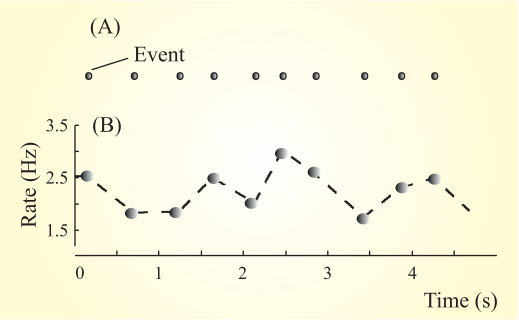

Figure 1.1 The QRS complexes in the ECG or extracellularly recorded spike trains can be considered as a series of events such as shown in (A). The rate of events can be depicted as the inverse of the interval between the events (B); here the inverse of the interval between each pair of events is plotted at the instant of the second event of the pair. The signal in (B) is unevenly sampled because the rate measure is available only at the occurrence of the events; the dashed line is a linear interpolation between these measures.

The disadvantage of this is that wide bins are associated with a low sample rate and thus a low Nyquist frequency, which limits the bandwidth of the spectral analysis.

(3) The most elegant solution is to use Lomb's algorithm for estimating the spectrum. This algorithm is specially designed to deal with unevenly sampled time series directly without the assumptions demanded by interpolation and resampling techniques (Lomb, 1976; Press et al., 1992; Scargle, 1982; van Drongelen et al., 2009). The background and application of this algorithm will be further described in Sections 1.2.1 and 1.2.2.

1.2.1 Lomb's Algorithm

The idea of Lomb's algorithm is similar to the development of the Fourier series, namely, to represent a signal by a sum of sinusoidal waves (see chapter 5 in van Drongelen, 2007). Lomb's procedure is to fit a demeaned time series x that **may be sampled unevenly** to a weighted pair of cosine and sine waves, where the cosine is weighted by coefficient a and the sine by coefficient b. The fitting procedure is performed over N samples of $x(n)$ obtained at times t_n and repeated for each frequency f.

$$\boxed{P(a,b,f,t_n) = a\cos(2\pi f t_n) + b\sin(2\pi f t_n)} \tag{1.1}$$

Coefficients a and b are unknown and must be obtained from the fitting procedure. For example, we can fit P to signal x by minimizing the squared difference between them over all samples: that is, minimize $\varepsilon^2 = \sum_{n=0}^{N-1}[P - X(n)]^2$. We repeat this minimization for each frequency f. To accomplish this, we follow the same procedure for developing the Fourier series (chapter 5 in van Drongelen, 2007) and set the partial derivative for each coefficient to zero to find the minimum of the error, that is:

$$\partial\varepsilon^2/\partial a = 0 \tag{1.2a}$$

and

$$\partial\varepsilon^2/\partial b = 0 \tag{1.2b}$$

For convenience, in the following we use a shorthand notation in addition to the full notation. In the shorthand notation: $C = \cos(2\pi f t_n)$, $S = \sin(2\pi f t_n)$, and $X = x(n)$.

For the condition in Equation (1.2a) we get:

$$\partial\varepsilon^2/\partial a = \sum 2[P - x(n)]\frac{\partial[P - x(n)]}{\partial a} = \sum 2(aC + bS - X)C$$
$$= \sum_{n=0}^{N-1} 2\left[\underbrace{a\cos(2\pi f t_n) + b\sin(2\pi f t_n)}_{P} - x(n)\right]\underbrace{\cos(2\pi f t_n)}_{\partial[P - x(n)]/\partial a} = 0$$

This and a similar expression obtained from the condition in Equation (1.2b) results in the following two equations:

$$\sum XC = a\sum C^2 + b\sum CS$$
$$\sum_{n=0}^{N-1} X(n)\cos(2\pi f t_n) = a\sum_{n=0}^{N-1}\cos^2(2\pi f t_n) + b\sum_{n=0}^{N-1}\cos(2\pi f t_n)\sin(2\pi f t_n) \tag{1.3a}$$

and

$$\sum XS = a\sum CS + b\sum S^2$$
$$\sum_{n=0}^{N-1} X(n)\sin(2\pi f t_n) = a\sum_{n=0}^{N-1}\cos(2\pi f t_n)\sin(2\pi f t_n) + b\sum_{n=0}^{N-1}\sin^2(2\pi f t_n) \tag{1.3b}$$

Thus far the procedure is similar to the standard Fourier analysis described in chapter 5 in van Drongelen (2007). The special feature in Lomb's algorithm is that for each frequency f, the sample times t_n are now shifted by an amount τ (Fig. 1.2). Thus, in Equations (1.3a) and (1.3b), t_n becomes $t_n - \tau$. The critical step is that for each frequency f, we select an optimal time shift τ so that the cosine−sine cross-terms ($\sum CS$) disappear, that is:

$$\sum CS = \sum_{n=0}^{N-1}\cos(2\pi f(t_n-\tau))\sin(2\pi f(t_n-\tau)) = 0 \tag{1.4}$$

Using the trigonometric identity $\cos(A)\sin(B) = \frac{1}{2}[\sin(A-B) - \sin(A+B)]$, this can be simplified into:

$$\frac{1}{2}\left[\sum_{n=0}^{N-1}\underbrace{\sin(0)}_{0} - \sin(4\pi f(t_n-\tau))\right] = 0 \rightarrow \sum_{n=0}^{N-1}\sin(4\pi f(t_n-\tau)) = 0$$

To separate the expressions for t_n and τ, we use the trigonometric relationship $\sin(A-B) = \sin(A)\cos(B) - \cos(A)\sin(B)$ to get the following expression:

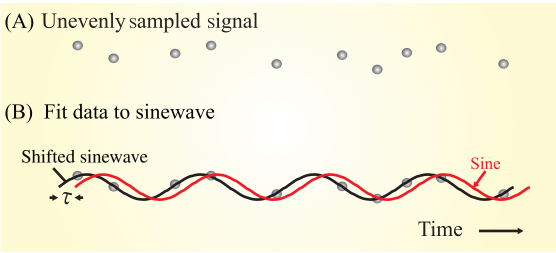

Figure 1.2 The Lomb algorithm fits sinusoidal signals to time series that may be unevenly sampled, as in the example in (A). The fit procedure (B) is optimized by shifting the sinusoidal signals by an amount τ.

$$\sum_{n=0}^{N-1}\sin(4\pi ft_n)\cos(4\pi f\tau) - \sum_{n=0}^{N-1}\cos(4\pi ft_n)\sin(4\pi f\tau)$$
$$= \cos(4\pi f\tau)\sum_{n=0}^{N-1}\sin(4\pi ft_n) - \sin(4\pi f\tau)\sum_{n=0}^{N-1}\cos(4\pi ft_n) = 0$$

This can be further simplified into:

$$\sin(4\pi f\tau)/\cos(4\pi f\tau) = \tan(4\pi f\tau) = \sum_{n=0}^{N-1}\sin(4\pi ft_n) \Big/ \sum_{n=0}^{N-1}\cos(4\pi ft_n)$$

Hence, condition (1.4) is satisfied if:

$$\boxed{\tau = \tan^{-1}\left[\sum_{n=0}^{N-1}\sin(4\pi ft_n) \Big/ \sum_{n=0}^{N-1}\cos(4\pi ft_n)\right] \Big/ 4\pi f} \tag{1.5}$$

The value of variable τ as a function of frequency f can be found with Equation (1.5), and by applying the appropriate shift $t_n \rightarrow (t_n - \tau)$, the cross-terms in Equations (1.3a) and (1.3b) become zero. Now we can determine the a and b coefficients for each frequency from the simplified expressions obtained from Equations (1.3a) and (1.3b) without the cross-terms:

$$\sum XC = a\sum C^2$$

$$\sum_{n=0}^{N-1}X(n)\cos(2\pi f(t_n - \tau)) = a\sum_{n=0}^{N-1}\cos^2(2\pi f(t_n - \tau)) \tag{1.6a}$$

$$\rightarrow \boxed{a = \sum XC/\sum C^2 = \sum_{n=0}^{N-1}x(n)\cos(2\pi f(t_n - \tau)) \Big/ \sum_{n=0}^{N-1}\cos^2(2\pi f(t_n - \tau))}$$

and

$$\sum XS = b\sum S^2$$

$$\sum_{n=0}^{N-1}x(n)\sin(2\pi f(t_n - \tau)) = b\sum_{n=0}^{N-1}\sin^2(2\pi f(t_n - \tau)) \tag{1.6b}$$

$$\rightarrow \boxed{b = \sum XS/\sum S^2 = \sum_{n=0}^{N-1}x(n)\sin(2\pi f(t_n - \tau)) \Big/ \sum_{n=0}^{N-1}\sin^2(2\pi f(t_n - \tau))}$$

Now we can compute the sum of $P^2(a, b, f, t_n)$—that is, the sum of squares of the sinusoidal signal in Equation (1.1) for all t_n—in order to obtain an expression that is proportional with the power spectrum S of $x(n)$ as a function of f:

$$
\begin{aligned}
S(f,a,b) &= \sum_{n=0}^{N-1} P^2(a,b,f,t_n) = \sum (aC+bS)^2 = \sum a^2C^2 + b^2S^2 + \overbrace{2abCS}^{cross\text{-}terms} \\
&= \sum_{n=0}^{N-1}\left[a^2\cos^2(2\pi f(t_n-\tau)) + b^2\sin^2(2\pi f(t_n-\tau)) + \underbrace{cross\text{-}terms}_{0}\right]
\end{aligned}
\tag{1.7}
$$

Since we shift by τ, all cross-terms vanish and by substitution of the expressions for the a and b coefficients in Equation (1.7) we get:

$$
\begin{aligned}
S(f) &= \frac{\left(\sum XC\right)^2}{\left(\sum C^2\right)^2}\sum C^2 + \frac{\left(\sum XS\right)^2}{\left(\sum S^2\right)^2}\sum S^2 \\
&= \frac{\left[\sum_{n=0}^{N-1} x(n)\cos(2\pi f(t_n-\tau))\right]^2}{\left[\sum_{n=0}^{N-1}\cos^2(2\pi f(t_n-\tau))\right]^2}\sum_{n=0}^{N-1}\cos^2(2\pi f(t_n-\tau)) \\
&\quad + \frac{\left[\sum_{n=0}^{N-1} x(n)\sin(2\pi f(t_n-\tau))\right]^2}{\left[\sum_{n=0}^{N-1}\sin^2(2\pi f(t_n-\tau))\right]^2}\sum_{n=0}^{N-1}\sin^2(2\pi f(t_n-\tau))
\end{aligned}
$$

This can be further simplified into:

$$
\begin{aligned}
S(f) &= \frac{\left(\sum XC\right)^2}{\sum C^2} + \frac{\left(\sum XS\right)^2}{\sum S^2} \\
&= \frac{\left[\sum_{n=0}^{N-1} x(n)\cos(2\pi f(t_n-\tau))\right]^2}{\sum_{n=0}^{N-1}\cos^2(2\pi f(t_n-\tau))} + \frac{\left[\sum_{n=0}^{N-1} x(n)\sin(2\pi f(t_n-\tau))\right]^2}{\sum_{n=0}^{N-1}\sin^2(2\pi f(t_n-\tau))}
\end{aligned}
\tag{1.8}
$$

The expression for the power spectrum in Equation (1.8) is sometimes divided by 2 (to make it equal to the standard power spectrum based on the Fourier transform; see Appendix 1.1), or by $2\sigma^2$ (σ^2—variance of x) for the determination of the statistical significance of spectral peaks. (Some of the background for this normalization is described in Appendix 1.1; for more details, see Scargle, 1982.) By applying the normalization we finally get:

$$\boxed{S(f) = \frac{1}{2\sigma^2}\left\{\frac{\left[\sum_{n=0}^{N-1} x(n)\cos(2\pi f(t_n-\tau))\right]^2}{\sum_{n=0}^{N-1}\cos^2(2\pi f(t_n-\tau))} + \frac{\left[\sum_{n=0}^{N-1} x(n)\sin(2\pi f(t_n-\tau))\right]^2}{\sum_{n=0}^{N-1}\sin^2(2\pi f(t_n-\tau))}\right\}} \tag{1.9}$$

From the above derivation, we can see that Lomb's procedure allows (but does not require) unevenly sampled data. Note that in Equations (1.7) and (1.8) we did not compute power as the square of the cosine and sine coefficients, a and b, as we would do in the standard Fourier transform; this is because in Lomb's approach the sinusoidal signals are not required to have a complete period within the epoch determined by the samples $x(n)$. Because we do not have this requirement, the frequency f is essentially a continuous variable and the spectral estimate we obtain by this approach is therefore not limited by frequency resolution (in the DFT/FFT, the frequency resolution is determined by the total epoch of the sampled data) and range (in the DFT/FFT, the maximum frequency is determined by the Nyquist frequency). However, to avoid misinterpretation, it is common practice to limit the bandwidth of the Lomb spectrum to less than or equal to half the average sample rate. Similarly, the commonly employed frequency resolution is the inverse of the signal's epoch.

1.2.2 A MATLAB Example

To test Lomb's algorithm we apply it to a signal that consists of two sinusoidal signals (50 and 130 Hz) plus a random noise component (this is the same example used in fig. 7.2A in van Drongelen, 2007). In this example (implemented in MATLAB script Pr1_1.m), we sample the signal with randomly distributed intervals (2000 points) and specify a frequency scale (f in the script) up to 500 Hz. Subsequently we use Equations (1.5), (1.8), and (1.9) to compute τ (tau in the script) and the unscaled and scaled versions of power spectrum $S(f)$ (Pxx in the script) of input $x(n)$ (x in the script). This script is available on http://www.elsevierdirect.com/companions/9780123849151.

The following script (Pr1_1.m) uses the Lomb algorithm to compute the spectrum from an unevenly sampled signal. The output of the script is a plot of the input (an unevenly sampled time domain) signal and its associated Lomb spectrum.

```
% Pr1_1.m
% Application of Lomb Spectrum
clear;

t=rand(2000,1); t=sort(t);        % An array of 2000 random sample intervals
f=[1:500];                        % The desired frequency scale
% frequencies same as pr7_1.m in van Drongelen (2007)
f1=50;
f2=130;
% data plus noise as in pr7_1.m in van Drongelen (2007)
x=sin(2*pi*f1*t)+sin(2*pi*f2*t);
x=x+randn(length(t),1);
var=(std(x))^2;                   % The signal's variance

% Main Loop
for i=1:length(f)
  h1=4*pi*f(i)*t;
  %Equation (1.5)
  tau=atan2(sum(sin(h1)), sum(cos(h1)))/(4*pi*f(i));
  h2=2*pi*f(i)*(t-tau);
  %Equation (1.8)
  Pxx(i)=(sum(x.*cos(h2)).^2)/sum(cos(h2).^2)+...
                                 (sum(x.*sin(h2)).^2)/sum(sin(h2).^2);
end;
% Normalize; Equation (1.9)
Pxx=Pxx/(2*var);
% Plot the Results
figure;
subplot(2,1,1), plot(t,x,'.-')
title('Irregularly Sampled Signal (USE ZOOM TO INSPECT UNEVEN
SAMPLING)')
xlabel('Time (s)');ylabel('Amplitude')
subplot(2,1,2),plot(f,Pxx);
title('Lomb Spectrum')
xlabel('Frequency (Hz)');ylabel('Normalized Power')
```

1.3 The Hilbert Transform

One of the current frontiers in neuroscience is marked by our lack of understanding of neuronal network function. A first step in unraveling network activities is to record from multiple neurons and/or networks simultaneously. A question that often arises in this context is which signals lead or lag; the underlying thought here is that the signals that lead cause the signals that lag. Although this approach is not foolproof, since one can only make reasonable inferences about causality if all connections between and activities of the neuronal elements are established, it is a first step in analyzing network function. Multiple techniques to measure lead and lag can be used. The simplest ones are cross-correlation and coherence (for an overview of these techniques, see chapter 8 in van Drongelen, 2007). A rather direct method to examine lead and lag is to determine the phase of simultaneously recorded signals. If the phase difference between two signals is not too big, one considers signal 1 to lead signal 2 if the phase of signal 1 (ϕ_1) is less than the phase of signal 2 (ϕ_2): $\phi_1 < \phi_2$. Of course this procedure should be considered as a heuristic approach to describe a causal sequence between the components in the network activity since there is no guarantee that a phase difference reflects a causal relationship between neural element 1 (generating signal 1) and neural element 2 (generating signal 2). In this example one could easily imagine alternatives where neural elements 1 and 2 are both connected to a common source causing both signals, or where element 2 is connected to element 1 via a significant number of relays; in both alternatives the condition $\phi_1 < \phi_2$ might be satisfied without a direct causal relationship from element 1 to element 2. A frequently used technique to compute a signal's phase is the Hilbert transform, which will be described in the remainder of this chapter. An alternative approach to study causality in multichannel data is discussed in Chapter 7.

The Hilbert transform is a useful tool to determine the amplitude and instantaneous phase of a signal. We will first define the transform before demonstrating the underlying mathematics. An easy way of introducing the application of the Hilbert transform is by considering Euler's equation multiplied with a constant A:

$$A\mathrm{e}^{\mathrm{j}\omega t} = A[\cos(\omega t) + \mathrm{j}\,\sin(\omega t)] = A\cos(\omega t) + \mathrm{j}A\sin(\omega t) \tag{1.10}$$

In this example we consider the first term in Equation (1.10), $f(t) = A\cos(\omega t)$, as the **signal** under investigation. This signal is ideal to demonstrate the Hilbert transform application because in this example we can see that the amplitude of $f(t)$ is A, and its instantaneous phase ϕ is ωt. The terminology for the Hilbert transform is as follows: the imaginary component, the second term, in Equation (1.10) $\tilde{f}(t) = A\sin(\omega t)$ is defined as the **Hilbert transform** of $f(t)$ (we will discuss further details in Sections 1.3.1 and 1.3.2 below), and the sum of both the signal and its Hilbert transform multiplied by j generates a complex signal:

$$f_a(t) = A\,\mathrm{e}^{\mathrm{j}\omega t} = A\cos(\omega t) + \mathrm{j}\,A\sin(\omega t) = f(t) + \mathrm{j}\tilde{f}(t)$$

in which $f_a(t)$ is defined as the **analytic signal**.

To summarize, the real part of the analytic signal is the signal under investigation $f(t)$ and its imaginary component is the Hilbert transform $\tilde{f}(t)$ of the signal. The analysis procedure is summarized in Fig. 1.3. As can be seen in Fig. 1.3A and B, we can use the analytic signal $A\,\mathrm{e}^{\mathrm{j}\omega t}$ to determine amplitude A and instantaneous phase ωt of any point, such as the one indicated by *. The amplitude is:

$$A = \sqrt{\text{real component}^2 + \text{imaginary component}^2}$$

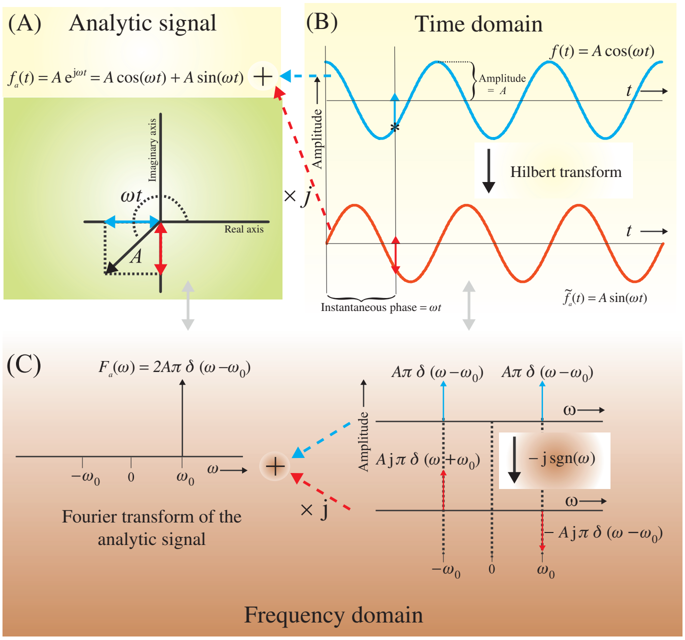

Figure 1.3 The signal amplitude A and instantaneous phase ωt of point * of the cosine function (B, $f(t)$, blue) can be determined with the so-called analytic signal (A). The analytic signal consists of a real part equal to the signal under investigation (the cosine) and an imaginary component (the sine). The imaginary component (red) is defined as the Hilbert transform $\tilde{f}(t)$ of the signal $f(t)$. The frequency domain equivalents of the cosine wave, the sine wave, the Hilbert transform procedure, and the analytic signal are shown in (C). See text for further explanation.

and the phase is:

$$\phi = \tan^{-1}\left[\frac{\text{imaginary component}}{\text{real component}}\right]$$

Again, in this example we did not need the analytic signal to determine phase and amplitude for our simple cosine signal, but our finding may be generalized to other signals where such a determination is not trivial.

1.3.1 The Hilbert Transform in the Frequency Domain

As can be seen in the earlier example (depicted in Fig. 1.3), the Hilbert transform can be considered as a phase shift operation on $f(t)$ to generate $\tilde{f}(t)$. In our example the signal $\cos(\omega t)$ is shifted by $-\pi/2$ rad (or $-90°$) to generate its Hilbert transform $\cos(\omega t - \pi/2) = \sin(\omega t)$. We may generalize this property and define a

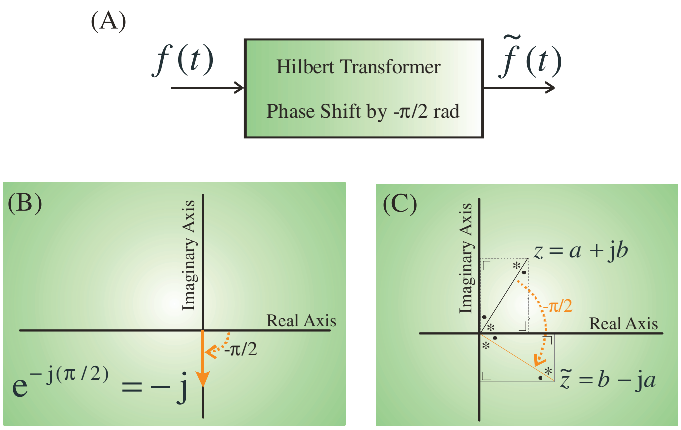

Figure 1.4 (A) The Hilbert transform can be represented as the operation of an LTI system (the Hilbert transformer). Input $f(t)$ is transformed into $\tilde{f}(t)$ by shifting it by $-\pi/2$ rad ($-90°$). (B) The Hilbert transform operation in the frequency domain can be represented as a multiplication with $e^{-j(\pi/2)} = -j$ (orange arrow). (C) Example of the Hilbert transform in the frequency domain—that is, multiplication of a complex number $z = a + jb$ with $-j$. The result is $\tilde{z} = b - ja$. As can be seen, the result is a $-90°$ rotation. Note in this panel that 90° angles are indicated by ∟ and that the angles indicated by • and * add up to 90°.

Hilbert transformer as a phase-shifting (linear time invariant, LTI) system that generates the Hilbert transform of its input (Fig. 1.4A). The generalization of this property associated with the cosine is not too far of a stretch if you recall that, with the real Fourier series, any periodic signal can be written as the sum of sinusoidal signals (the cosine and sine waves in equation (5.1) in van Drongelen, 2007) and that our above results can be applied to each of these sinusoidal components.

To further define the shifting property of the Hilbert transformer (see Fig. 1.4A), we begin to explore this operation in the frequency domain, because here the procedure of shifting the phase of a signal by $-\pi/2$ rad is relatively easy to define as a multiplication by $e^{-j(\pi/2)} = -j$ (Fig. 1.4B). If this is not obvious to you, consider the effect of this multiplication for any complex number $z = e^{j\phi}$ (representing phase ϕ) that can also be written as the sum of its real and imaginary parts $z = a + jb$. Multiplication by $-j$ gives its Hilbert transform $\tilde{z} = -j(a + jb) = b - ja$, indeed corresponding to a $-90°$ rotation of z (see Fig. 1.4C). Although the multiplication with $-j$ is correct for the positive frequencies, a $-90°$ shift for the negative frequencies in the Fourier transform (due to the negative values of ω) corresponds to multiplication with $e^{j(\pi/2)} = j$. Therefore, the operation of the Hilbert transform in the frequency domain can be summarized as:

$$\boxed{\text{multiplication by} - j\,\mathrm{sgn}(\omega)} \qquad (1.11)$$

Here we use the so-called signum function sgn (Appendix 1.2, Fig. A2.1) defined as:

$$\mathrm{sgn}(\omega)\begin{cases} -1 & \text{for } \omega < 0 \\ 0 & \text{for } \omega = 0 \\ 1 & \text{for } \omega > 0 \end{cases} \tag{1.12}$$

Let us go back to our phase-shifting system depicted in Fig. 1.4A and define its unit impulse response as $h(t)$ and its associated frequency response as $H(\omega)$. Within this approach, the Hilbert transform is the convolution of input $f(t)$ with $h(t)$. Using our knowledge about convolution (if you need to review this, see section 8.3.2 in van Drongelen, 2007), we can also represent the Hilbert transform in the frequency domain as the product of $F(\omega)$—the Fourier transform of $f(t)$—and $H(\omega)$. This is very convenient because we just determined above that the Hilbert transform in the frequency domain corresponds to a multiplication with $-\mathrm{j}\,\mathrm{sgn}(\omega)$. To summarize, we now have the following three Fourier transform pairs:

$$\begin{array}{ll} \text{System's input} \Leftrightarrow \text{Fourier transform:} & f(t) \Leftrightarrow F(\omega) \\ \text{System's unit impulse response} \Leftrightarrow \text{Fourier transform:} & h(t) \Leftrightarrow H(\omega) \\ \text{Hilbert transform} \Leftrightarrow \text{Fourier transform:} & f(t) \otimes h(t) \Leftrightarrow F(\omega)H(\omega) \end{array} \tag{1.13}$$

Using these relationships and Equation (1.11), we may state that the Fourier transform of the unit impulse response (i.e., the frequency response) of the Hilbert transformer is:

$$\boxed{H(\omega) = -\mathrm{j}\mathrm{sgn}(\omega)} \tag{1.14}$$

We can use the expression we found for $H(\omega)$ to examine the above example of Euler's equation

$$A\,\mathrm{e}^{\mathrm{j}\omega_0 t} = \overbrace{\underbrace{A\cos(\omega_0 t)}_{\text{Signal}} + \mathrm{j}\underbrace{A\sin(\omega_0 t)}_{\text{Hilbert transform}}}^{\text{Analytical signal}}$$

in the frequency domain. The Fourier transform of the cosine term (using equation (6.13) in van Drongelen, 2007) is:

$$A\pi[\delta(\omega + \omega_0) + \delta(\omega - \omega_0)] \tag{1.15}$$

Now, according to Equation (1.13), the Fourier transform of the cosine's Hilbert transform is the product of the Fourier transform of the input signal (the cosine) and the frequency response of the Hilbert transformer $H(\omega)$, that is:

$$\begin{array}{l} \{A\pi[\delta(\omega + \omega_0) + \delta(\omega - \omega_0)]\}\{-\mathrm{j}\,\mathrm{sgn}(\omega)\} \\ A\pi[\delta(\omega + \omega_0)(-\mathrm{j}\,\mathrm{sgn}(\omega)) + \delta(\omega - \omega_0)(-\mathrm{j}\,\mathrm{sgn}(\omega))] \end{array} \tag{1.16}$$

Because $\delta(\omega+\omega_0)$ is only nonzero for $\omega=-\omega_0$ and $\delta(\omega-\omega_0)$ is only nonzero for $\omega=\omega_0$, we may rewrite the $-\mathrm{j}\ \mathrm{sgn}(\omega)$ factors in Equation (1.16) and we get:

$$A\pi[\delta(\omega+\omega_0)(-\mathrm{j}\,\mathrm{sgn}(-\omega_0))+\delta(\omega-\omega_0)(-\mathrm{j}\,\mathrm{sgn}(\omega_0))]$$

Now we use the definition of sgn, $\mathrm{sgn}(-\omega_0)=-1$ and $\mathrm{sgn}(\omega_0)=1$ (Equation (1.12)), and simplify the expression to:

$$A\pi[\delta(\omega+\omega_0)(\mathrm{j})+\delta(\omega-\omega_0)(-\mathrm{j})]=A\mathrm{j}\pi[\delta(\omega+\omega_0)-\delta(\omega-\omega_0)] \tag{1.17}$$

As expected, Equation (1.17) is the Fourier transform of $A\sin(\omega t)$ (see equation (6.14) in van Drongelen, 2007), which is indeed the Hilbert transform $\tilde{f}(t)$ of $f(t)=A\cos(\omega t)$.

Combining the above results, we can find the Fourier transform of the analytic signal $f_a(t)=A\cos(\omega t)+\mathrm{j}A\sin(\omega t)=f(t)+\mathrm{j}\tilde{f}(t)$. If we define the following pairs:

$$\begin{aligned}
&f_a(t)\Leftrightarrow F_a(\omega)\\
&f(t)\Leftrightarrow F(\omega)\\
&\tilde{f}(t)\Leftrightarrow \tilde{F}(\omega)
\end{aligned}$$

the above expressions can be combined in the following Fourier transform pair:

$$f_a(t)=f(t)+\mathrm{j}\tilde{f}(t)\Leftrightarrow F_a(\omega)=F(\omega)+\mathrm{j}\tilde{F}(\omega)$$

In the above equation we substitute the expressions for $F(\omega)$ from Equation (1.15) and $\tilde{F}(\omega)$ from Equation (1.17) and get:

$$\begin{aligned}
F_a(\omega)&=F(\omega)+\mathrm{j}\tilde{F}(\omega)\\
&=\overbrace{\underbrace{A\pi[\delta(\omega+\omega_0)+\delta(\omega-\omega_0)]}_{\text{Fourier transform of signal}}+\mathrm{j}\underbrace{\{A\mathrm{j}\pi[\delta(\omega+\omega_0)-\delta(\omega-\omega_0)]\}}_{\text{Fourier transform of Hilbert transform}}}^{\text{Fourier transform of analytical signal}}
\end{aligned}$$

With a bit of algebra we obtain:

$$\boxed{F_a(\omega)=F(\omega)+\mathrm{j}\tilde{F}(\omega)=2\pi A\delta(\omega-\omega_0)} \tag{1.18}$$

This interesting finding shows that the Fourier transform of the analytic signal has zero energy at negative frequencies and only a peak at $+\omega_0$. The peak's amplitude at $+\omega_0$ is double the size of the corresponding peak in $F(\omega)$ (Fig. 1.3C). This finding may be generalized as: "The Fourier transform of the analytic signal $F_a(\omega)$ has no energy at negative frequencies $-\omega_0$, it only has energy at positive frequencies $+\omega_0$ and its amplitude is double the amplitude at $+\omega_0$ in $F(\omega)$."

1.3.2 The Hilbert Transform in the Time Domain

From the frequency response presented in Equation (1.14) and the relationship between convolution in the time and frequency domains (section 8.3.2, van Drongelen, 2007), we know that the unit impulse response $h(t)$ of the Hilbert transformer (Fig. 1.4A) is the inverse Fourier transform of $-\mathrm{j}\,\mathrm{sgn}(\omega)$. You can find details of $\mathrm{sgn}(t)$ and its Fourier transform in Appendix 1.2; using the signum's Fourier transform, we can apply the duality property (section 6.2.1, van Drongelen, 2007) to determine the inverse Fourier transform for $-\mathrm{j}\,\mathrm{sgn}(\omega)$. For convenience we restate the duality property as:

$$\text{if } f(t) \Leftrightarrow F(\omega), \text{ then } F(t) \Leftrightarrow 2\pi f(-\omega) \tag{1.19a}$$

Applying this to our signum function (see also Appendix 1.2), we can define the inverse Fourier transform of $\mathrm{sgn}(\omega)$:

$$\mathrm{sgn}(t) \Leftrightarrow \frac{2}{\mathrm{j}\omega}, \text{ therefore } \frac{2}{\mathrm{j}t} \Leftrightarrow 2\pi \underbrace{\mathrm{sgn}(-\omega)}_{-\mathrm{sgn}(\omega)} \tag{1.19b}$$

Note that we can substitute $-\mathrm{sgn}(\omega) = \mathrm{sgn}(-\omega)$ because the signum function (Fig. A2.1) has odd symmetry (defined in Appendix 5.2 in van Drongelen, 2007). Using the result from applying the duality property in Equation (1.19b), we can determine the inverse Fourier transform for the frequency response of the Hilbert transformer $H(\omega) = -\mathrm{j}\,\mathrm{sgn}(\omega)$ and find the corresponding unit impulse response $h(t)$. Because 2π and j are both constants, we can multiply both sides with j and divide by 2π; this generates the following Fourier transform pair:

$$\boxed{h(t) = \frac{1}{\pi t} \Leftrightarrow H(\mathrm{j}\omega) = -\mathrm{j}\,\mathrm{sgn}(\omega)} \tag{1.20}$$

In Equation (1.14) we found that the frequency response of the Hilbert transformer is $-\mathrm{j}\,\mathrm{sgn}(\omega)$. Because we know that multiplication in the frequency domain is equivalent to convolution in the time domain (chapter 8 in van Drongelen, 2007), we can use the result in Equation (1.20) to define the Hilbert transform $\tilde{f}(t)$ of signal $f(t)$ in both the time and frequency domains. We define the following Fourier transform pairs:

$$\begin{aligned} \text{the input: } & f(t) \Leftrightarrow F(\omega) \\ \text{the Hilbert transform of the input: } & \tilde{f}(t) \Leftrightarrow \tilde{F}(\omega) \\ \text{the unit impulse response of the Hilbert transformer: } & h(t) \Leftrightarrow H(\omega) \end{aligned}$$

Using the above pairs and Equation (1.20), the Hilbert transform and its frequency domain equivalent are:

$$\tilde{f}(t) = f(t) \otimes \underbrace{h(t)}_{\frac{1}{\pi t}} = \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{f(t)}{t-\tau} \mathrm{d}\tau \Leftrightarrow \tilde{F}(\omega) = F(\omega)H(\omega) \tag{1.21}$$

There is, however, a problem with our finding for the Hilbert transform expression in Equation (1.21), which is that there is a pole for $f(t)/(t-\tau)$ within the integration limits at $t=\tau$. The solution to this problem is to define the Hilbert transform as:

$$\boxed{\tilde{f}(t) = \frac{1}{\pi} \mathrm{CPV} \int_{-\infty}^{\infty} \frac{f(t)}{t-\tau} \mathrm{d}\tau} \tag{1.22}$$

in which CPV indicates the Cauchy principal value of the integral. The CPV is a mathematical tool to evaluate integrals that include poles within the integration limits. An example of such an application is given in Appendix 1.3. For those interested in the CPV procedure, we refer to a general mathematics text such as Boas (1966).

1.3.3 Examples

The Hilbert transform is available in MATLAB via the hilbert command. Note that this command produces the analytic signal $f(t) + \mathrm{j}\tilde{f}(t)$ and not the Hilbert transform itself; the Hilbert transform is the imaginary component of the output.

You can evaluate the example from Equation (1.10) by computing the Hilbert transform for the cosine and plot the amplitude and phase. Type the following in the MATLAB command window:

```
step=0.00001;                        % step size=1/sample rate
t=0:step:1;                          % timebase
x=cos(2*pi*4*t);                     % 4 Hz signal
xa=hilbert(x);                       % compute the analytic signal
Amplitude=abs(xa);                   % amplitude of the signal
Phase=atan2(imag(xa),real(xa));      % instantaneous phase
Ohmega=diff(Phase)/(2*pi*step);      % instantaneous frequency in Hz
figure;plot(t,x,'k');hold;
plot(t,Amplitude,'r');
plot(t,Phase,'g');
plot(t(1:length(t)-1),Ohmega,'m.')
axis([0 1 -5 5])
```

You will obtain a graph of a 4-Hz cosine function with an indication of its amplitude (a constant) in red, its instantaneous phase in green (note that we use the atan2 MATLAB command in the above example because we want to obtain phase angles between $-\pi$ and $+\pi$), and the frequency as the derivative of the phase in magenta.

You can now check both frequency characteristics we discussed by computing the Fourier transforms and plotting these in the same graph.

```
X=fft(x);                    % Fourier transform of the signal
XA=fft(xa);                  % Fourier transform of the analytic signal
figure;plot(abs(X),'k');hold;plot(abs(XA))
```

If you use the **zoom** function of the graph to study the peaks in the plot you will see that the peaks for the positive frequencies (far-left part of the graph) show a difference of a factor two between the Fourier transform of the analytic signal and the Fourier transform of the signal. The negative component (in the discrete version of the Fourier transform this is the far-right part of the graph) shows only a peak in the Fourier transform of the signal. Both observations are as expected from the theoretical considerations in Section 1.3.1.

Another property to look at is the phase shift between the signal and its Hilbert transform. This can be accomplished by typing the following lines:

```
figure; hold;
plot(t,imag(xa),'r');          % the imaginary part of the analytic signal=
                               % the Hilbert transform
plot(t,x,'k.')                 % the signal
plot(t,real(xa),'y')           % real part of the analytic signal=signal
```

Now you will get a figure with the signal (4-Hz cosine wave) in both black (thick line) and yellow (thin line); the Hilbert transform (the 4-Hz sine wave) is plotted in red.

Finally we will apply these techniques to an example in which we have two neural signals, one signal generated by a single neuron and one signal generated by the network in which the neuron is embedded. Our question here is how the phases of these two signals relate. First, the raw extracellular trace is rectified and sent through a low-pass filter with a 50-ms time constant (this technique of using the analytic signal to find the instantaneous phase usually works better with signals composed of a small band of frequencies and, in our case, we are only interested in the low-frequency behavior; see Pikovsky et al., 2001, for more details). For the cellular activity, we create a raster plot of the spike times and send it through the same low-pass filter. We now have two signals representing the low-pass-filtered spiking behavior of the cell and network (see Fig. 1.5Aiii). We can use the Hilbert

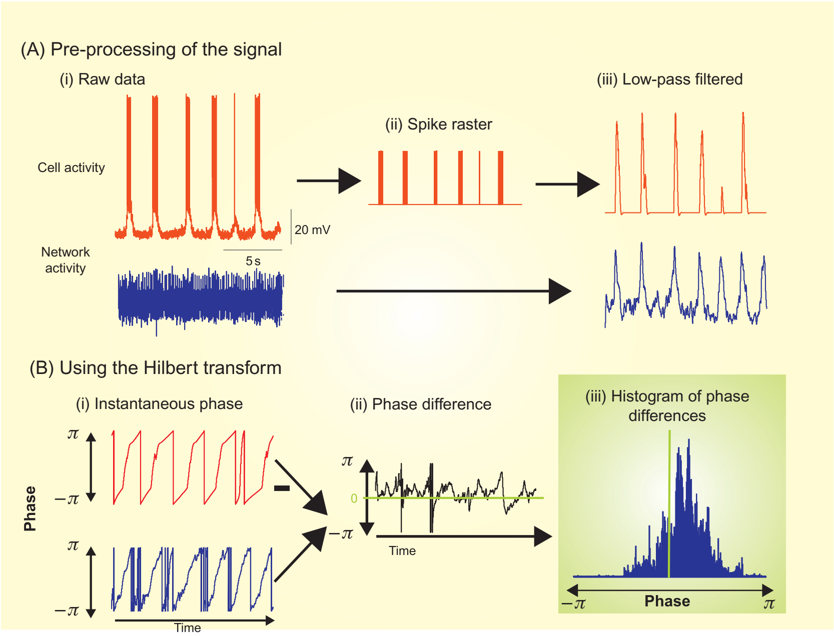

Figure 1.5 (A) Processing of a cellular and network activity (i) into a low-frequency index of spiking activity (iii) (see text for details). (B) The low-pass-filtered signals of (A) were transformed using the analytic signal technique to find the instantaneous phase over time (i). The relationship between the two signals was investigated by finding the difference between the phases over time (ii) and plotting these phase differences in a histogram (iii). In this example we observe that the overall effect is that the network activity leads and the cell activity lags—that is, the histogram (iii) of network activity phase minus cell activity phase is predominantly positive. (*From A. Martell, unpublished results, with permission.*)

transform technique to find the instantaneous phase of each signal (Fig. 1.5Bi). For our case, we are interested in how the phases of the cellular and network signals are related. To find this relationship, we calculate the difference between the two instantaneous phase signals at each point in time and then use this information to generate a histogram (see Fig. 1.5Bii−iii). This method has been used to compare how the phases of cellular and network signals are related for different types of cellular behavior (Martell et al., 2008).

Appendix 1.1

In the case of the standard power spectrum we have $S = XX^*/N$ (equation (7.1) in van Drongelen, 2007). The normalization by $1/N$ ensures that Parseval's conservation of energy theorem is satisfied (this theorem states that the sum of squares in the time domain and the sum of all elements in the power spectrum are equal; see Table 7.1 and Appendix 7.1 in van Drongelen, 2007). In the case of Lomb's algorithm we compute the sum of squares for each frequency by using the expression in

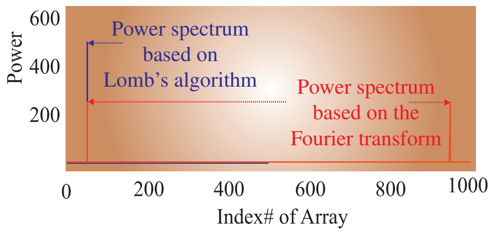

Figure A1.1 Spectral analysis of a 1-s epoch of a 50-Hz signal sampled at 1000 Hz. The graph depicts the superimposed results from a standard power spectrum (red) based on the Fourier transform and the power spectrum obtained with Lomb's algorithm (dark blue). Note that the total energy in both cases is identical. This figure can be created with Pr1_2.m.

Equation (1.8), which is based on Equation (1.7). Our expectation is therefore that Lomb's spectrum will also satisfy Parseval's theorem. However, there is a slight difference. In the standard Fourier transform the positive and negative frequencies each contain half the energy. Basically, this is due to the fact that the Fourier transform is based on the complex Fourier series, which includes negative frequencies. In contrast, if we compute the Lomb spectrum only up to the Nyquist frequency, we have all energy in the positive frequencies, and therefore its values are twice as large as compared to the standard power spectrum. An example for a single frequency is shown in Figure A1.1. This figure is based on a standard power spectrum and Lomb spectrum computed for the same input, a sine wave of 50 Hz. Thus, if we want the Lomb spectrum to have the same amplitudes as the standard power spectrum, we need to divide by two. Furthermore, if we want to normalize by the total power, we can divide by the variance σ^2. This normalization by $2\sigma^2$ is exactly the normalization commonly applied for Lomb's spectrum (see Equation (1.9) and Pr1_1.m).

Appendix 1.2

This appendix describes the signum function sgn(t), its derivative, and Fourier transform. The signum function is 1 for positive t and -1 for negative t (Fig. A2.1). Similar to the derivative of the unit step function $U(t)$ (section 2.2.2, fig. 2.4A in van Drongelen, 2007), the derivative of this function is only nonzero at $t = 0$. The only difference is that for sgn(t) the function increases by 2 units (from -1 to 1) instead of 1 unit (from 0 to 1) in $U(t)$. Since the derivative of the unit step is $\delta(t)$, the derivative of the signum function would be twice as large, that is:

$$\frac{\mathrm{d}[\mathrm{sgn}(t)]}{\mathrm{d}t} = 2\delta(t) \tag{A1.2.1}$$

The Fourier transform of the derivative of a function is equal to the Fourier transform of that function multiplied with jω. This property is similar to the relationship of the Laplace transform of a derivative of a function and the Laplace transform of the function itself (see section 9.3, equation (9.3) in van Drongelen, 2007). If we

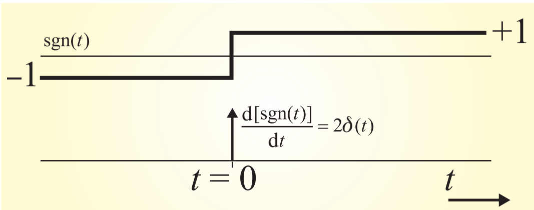

Figure A2.1 The signum function and its derivative, the unit impulse function with an amplitude of two.

now use this property and define the Fourier transform of sgn(t) as $S(\omega)$, we can apply the Fourier transform to both sides of Equation (A1.2.1):

$$\mathrm{j}\omega S(\omega) = 2 \tag{A1.2.2}$$

Recall that the Fourier transform of the unit impulse is 1 (see section 6.2.1, equation (6.9) in van Drongelen, 2007). Therefore, the Fourier transform pair for the signum function is:

$$\mathrm{sgn}(t) \Leftrightarrow S(\omega) = \frac{2}{\mathrm{j}\omega} \tag{A1.2.3}$$

Appendix 1.3

In Equation (1.22) we use the Cauchy principal value, CPV. This technique is used to approach integration of a function that includes a pole within the integration limits. We will not go into the mathematical details (for more on this subject please see a mathematics textbook such as Boas, 1966), but we will give an example to show the principle. For example, consider the integral $\int_{-\mathrm{d}}^{\mathrm{d}} (1/x)\mathrm{d}x$. The function $1/x$ in this integral has a pole (is unbounded) at $x = 0$. The Cauchy principal value technique approximates the integral as the sum of two separate integral:

$$\int_{-\mathrm{d}}^{\mathrm{d}} \frac{1}{x}\mathrm{d}x \approx \int_{-\mathrm{d}}^{-\varepsilon} \frac{1}{x}\mathrm{d}x + \int_{\varepsilon}^{\mathrm{d}} \frac{1}{x}\mathrm{d}x$$

where ε is a small positive value approaching zero. In this case the two integrals cancel and approach $\int_{-\mathrm{d}}^{\mathrm{d}} (1/x)\mathrm{d}x$. Our final result can be summarized as:

$$\mathrm{CPV} \int_{-\mathrm{d}}^{\mathrm{d}} \frac{1}{x}\mathrm{d}x = \lim_{\varepsilon \to 0} \left[\int_{-\mathrm{d}}^{-\varepsilon} \frac{1}{x}\mathrm{d}x + \int_{\varepsilon}^{\mathrm{d}} \frac{1}{x}\mathrm{d}x \right] = 0$$

Here the Cauchy principal value is indicated by CPV; in other texts you may also find PV or P.