4.9 Dynamic Fourier Analysis and Wavelets

If a time series, x_t, is stationary, its second-order behavior remains the same, regardless of the time t. It makes sense to match a stationary time series with sines and cosines because they, too, behave the same forever. Indeed, based on the Spectral Representation Theorem (Appendix C, §C.1), we may regard a stationary series as the superposition of sines and cosines that oscillate at various frequencies. As seen in this text, however, many time series are not stationary. Typically, the data are coerced into stationarity via transformations, or we restrict attention to parts of the data where stationarity appears to adhere. In some cases, the nonstationarity of a time series is of interest. That is to say, it is the local behavior of the process, and not the global behavior of the process, that is of concern to the investigator. As a case in point, we mention the explosion and earthquake series first presented in Example 1.7 (see Figure 1.7). The following example emphasizes the importance of dynamic (or time-frequency) Fourier analysis.

Example 4.21 Dynamic Spectral Analysis of Seismic Traces

Consider the earthquake and explosion series displayed in Figure 1.7; it should be apparent that the dynamics of the series are changing with time. The goal of this analysis is to summarize the spectral behavior of the signal as it evolves over time.

First, a spectral analysis is performed on a short section of the data. Then, the section is shifted, and a spectral analysis is performed on the new section. This process is repeated until the end of the data, and the results are shown an image in Figure 4.17 and Figure 4.18; in the images, darker areas correspond to higher power. Specifically, in this example, let x_t, for $t = 1, \ldots, 2048$, represent the series of interest. Then, the sections of the data that were analyzed were $\{x_{t_k+1}, \ldots, x_{t_k+256}\}$, for $t_k = 128k$, and $k = 0, 1, \ldots, 14$; e.g., the first section analyzed is $\{x_1, \ldots, x_{256}\}$, the second section analyzed is $\{x_{129}, \ldots, x_{384}\}$, and so on. Each section of 256 observations was tapered using a cosine bell, and spectral estimation was performed using a repeated Daniell kernel with weights $\frac{1}{9}\{1, 2, 3, 2, 1\}$; see page 204. The sections overlap each other, however, this practice is not necessary and sometimes not desirable.[17]

The results of the dynamic analysis are shown as the estimated spectra for frequencies up to 10 Hz (the folding frequency is 20 Hz) for each starting location (time), $t_k = 128k$, with $k = 0, 1, \ldots, 14$. The S component for the

[17] A number of technical problems exist in this setting because the process of interest is nonstationary and we have not specified the nature of the nonstationarity. In addition, overlapping intervals complicate matters by introducing another layer of dependencies among the spectra. Consequently, the spectral estimates of contiguous sections are dependent in a non-trivial way that we have not specified. Nevertheless, as seen from this example, dynamic spectral analysis can be a helpful tool in summarizing the local behavior of a time series.

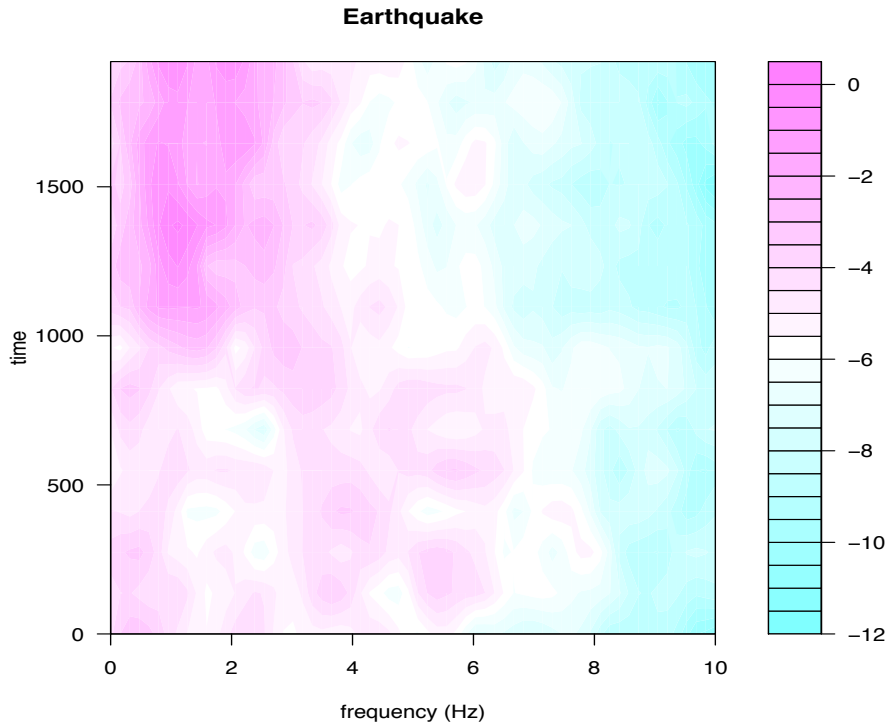

Fig. 4.17. Time-frequency image for the dynamic Fourier analysis of the earthquake series shown in Figure 1.7.

earthquake shows power at the low frequencies only, and the power remains strong for a long time. In contrast, the explosion shows power at higher frequencies than the earthquake, and the power of the signals (P and S waves) does not last as long as in the case of the earthquake.

The following is an R session that corresponds to the analysis of the explosion series. The images are generated using `filled.contour()` on the log of the power; ~~this, as well as using a gray scale and limiting the number of levels was done to produce a decent black-and-white graphic. The images look better in color, so we advise removing the `nlevels=...` and the `col=gray(...)` parts of the code.~~ We also include the code for obtaining a three-dimensional graphic to display the information, however, the graphic is not exhibited in the text.

```
nobs = length(EXP6)  # number of observations
wsize = 256   # window size
overlap = 128  # overlap
ovr = wsize-overlap
nseg = floor(nobs/ovr)-1;  # number of segments
krnl = kernel("daniell", c(1,1))  # kernel
ex.spec = matrix(0, wsize/2, nseg)
 for (k in 1:nseg) {
  a = ovr*(k-1)+1
```

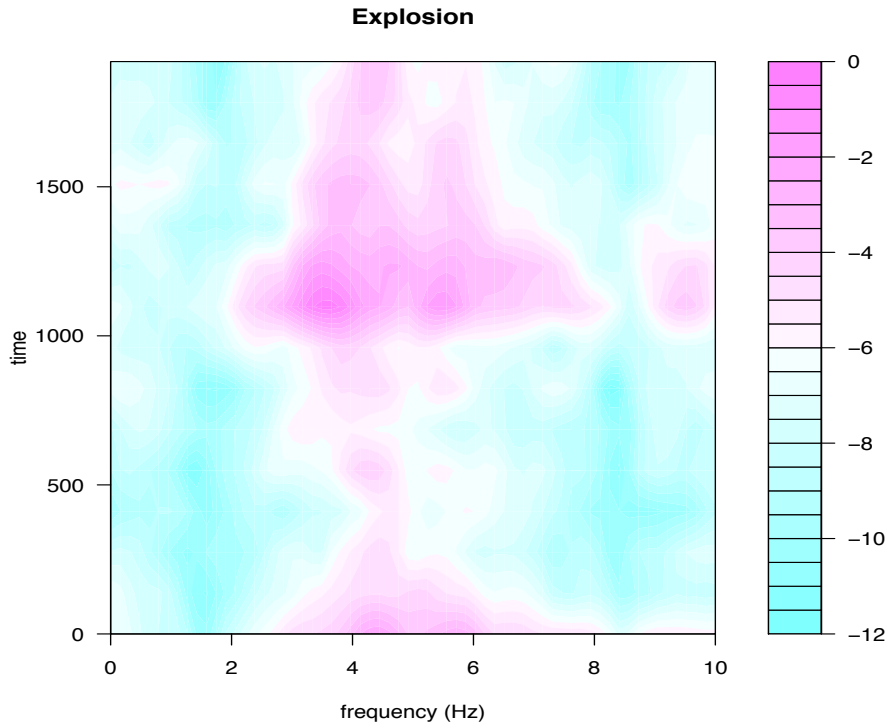

Fig. 4.18. Time-frequency image for the dynamic Fourier analysis of the explosion series shown in Figure 1.7.

```
  b = wsize+ovr*(k-1)
  ex.spec[,k] = spectrum(EXP6[a:b], krnl, taper=.5, plot=F)$spec }
x = seq(0, 10,  len = nrow(ex.spec)/2)
y = seq(0, ovr*nseg, len = ncol(ex.spec))
z = ex.spec[1:(nrow(ex.spec)/2),]
filled.contour(x, y, log(z), ylab="time", xlab="frequency (Hz)",
    main="Explosion")
persp(x, y, z, zlab="Power", xlab="frequency (Hz)", ylab="time",
    ticktype="detailed", theta=25,d=2, main="Explosion")  # not shown
```

One way to view the time-frequency analysis of Example 4.21 is to consider it as being based on local transforms of the data x_t of the form

$$d_{j,k} = n^{-1/2} \sum_{t=1}^{n} x_t \psi_{j,k}(t), \tag{4.112}$$

where

$$\psi_{j,k}(t) = \begin{cases} (n/m)^{1/2} h_t \, e^{-2\pi i t j/m} & t \in [t_k + 1, t_k + m], \\ 0 & \text{otherwise}, \end{cases} \tag{4.113}$$

where h_t is a taper and m is some fraction of n. In Example 4.21, $n = 2048$, $m = 256$, $t_k = 128k$, for $k = 0, 1, \ldots, 14$, and h_t was a cosine bell taper

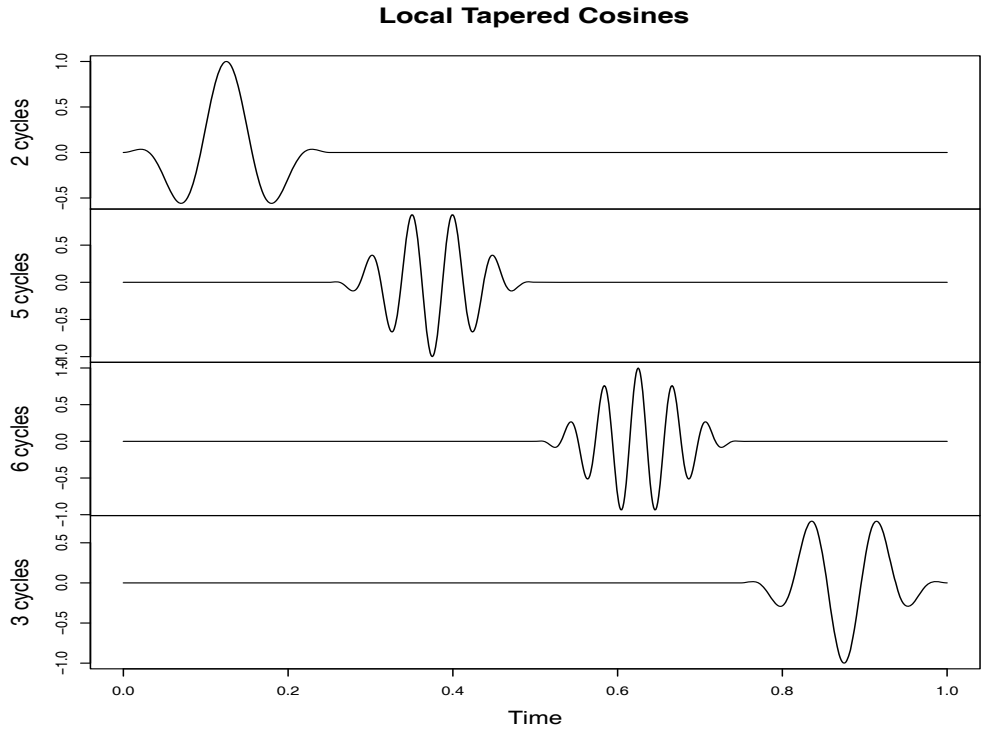

Fig. 4.19. Local, tapered cosines at various frequencies.

over 256 points. In (4.112) and (4.113), j indexes frequency, $\omega_j = j/m$, for $j = 1, 2, \ldots, [m/2]$, and k indexes the location, or time shift, of the transform. In this case, the transforms are based on tapered cosines and sines that have been zeroed out over various regions in time. The key point here is that the transforms are based on *local* sinusoids. Figure 4.19 shows an example of four local, tapered cosine functions at various frequencies. In that figure, the length of the data is considered to be one, and the cosines are localized to a fourth of the data length.

In addition to dynamic Fourier analysis as a method to overcome the restriction of stationarity, researchers have sought various alternative methods. A recent, and successful, alternative is wavelet analysis. The website `http://www.wavelet.org` is devoted to wavelets, which includes information about books, technical papers, software, and links to other sites. In addition, we mention the monograph on wavelets by Daubechies (1992), the text by Percival and Walden (2000), and we note that many statistical software manufacturers have wavelet modules that sit on top of their base package. In this section, we rely primarily on the S-PLUS wavelets module (with a manual written by Bruce and Gao, 1996), however, we will present some R code where possible. The basic idea of wavelet analysis is to imitate dynamic Fourier analysis, but with functions (wavelets) that may be better suited to capture the local behavior of nonstationary time series.

Wavelets come in families generated by a father wavelet, ϕ, and a mother wavelet, ψ. The father wavelets are used to capture the smooth, low-frequency nature of the data, whereas the mother wavelets are used to capture the detailed, and high-frequency nature of the data. The father wavelet integrates to one, and the mother wavelet integrates to zero

$$\int \phi(t)dt = 1 \quad \text{and} \quad \int \psi(t)dt = 0. \tag{4.114}$$

For a simple example, consider the Haar function,

$$\psi(t) = \begin{cases} 1, & 0 \le t < 1/2, \\ -1, & 1/2 \le t < 1, \\ 0, & \text{otherwise.} \end{cases} \tag{4.115}$$

The father in this case is $\phi(t) = 1$ for $t \in [0, 1)$ and zero otherwise. The Haar functions are useful for demonstrating properties of wavelets, but they do not have good time-frequency localization properties. Figure 4.20 displays two of the more commonly used wavelets that are available with the S-PLUS wavelets module, the *daublet4* and *symmlet8* wavelets, which are described in detail in Daubechies (1992). The number after the name refers to the width and smoothness of the wavelet; for example, the symmlet10 wavelet is wider and smoother than the symmlet8 wavelet. Daublets are one of the first type of continuous orthogonal wavelets with compact support, and symmlets were constructed to be closer to symmetry than daublets. In general, wavelets do not have an analytical form, but instead they are generated using numerical methods.

Figure 4.20 was generated in S-PLUS using the wavelet module as follows:[18]

```
d4f <- wavelet("d4", mother=F)
d4m <- wavelet("d4")
s8f <- wavelet("s8", mother=F)
s8m <- wavelet("s8")
par(mfrow=c(2,2))
plot(d4f);  plot(d4m)
plot(s8f);  plot(s8m)
```

It is possible to draw some wavelets in R using the `wavethresh` package. In that package, daublets are called DaubExPhase and symmlets are called DaubLeAsymm. The following R session displays some of the available wavelets (this will produce a figure similar to Figure 4.20) and it assumes the `wavethresh` package has been downloaded and installed (see Appendix R, §R.2, for details on installing packages). The `filter.number` determines the width and smoothness of the wavelet.

[18] At this time, the R packages available for wavelet analysis are not extensive enough for our purposes, hence we will rely on S-PLUS for some of the demonstrations. We will provide R code when possible, and that will be based on the `wavethresh` package (version 4.2-1) that accompanies Nason (2008).

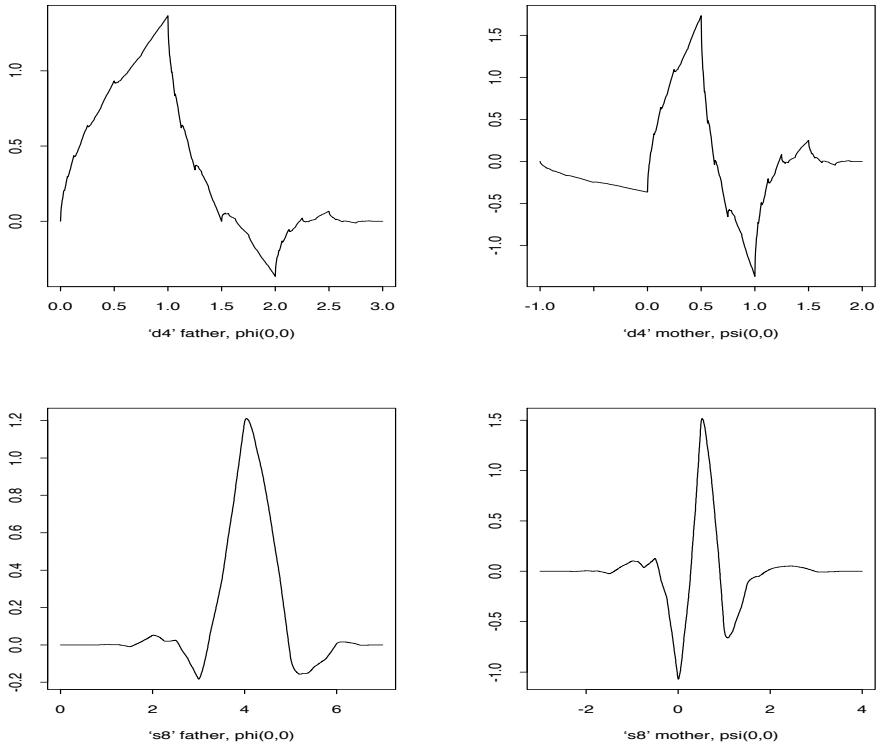

Fig. 4.20. Father and mother daublet4 wavelets (top row); father and mother symmlet8 wavelets (bottom row).

```
library(wavethresh)
par(mfrow=c(2,2))
draw(filter.number=4, family="DaubExPhase", enhance=FALSE, main="")
draw(filter.number=8, family="DaubExPhase", enhance=FALSE, main="")
draw(filter.number=4, family="DaubLeAsymm", enhance=FALSE, main="")
draw(filter.number=8, family="DaubLeAsymm", enhance=FALSE, main="")
```

When we depart from periodic functions, such as sines and cosines, the precise meaning of frequency, or cycles per unit time, is lost. When using wavelets, we typically refer to scale rather than frequency. The orthogonal wavelet decomposition of a time series, x_t, for $t = 1, \ldots, n$ is

$$x_t = \sum_k s_{J,k}\phi_{J,k}(t) + \sum_k d_{J,k}\psi_{J,k}(t) + \sum_k d_{J-1,k}\psi_{J-1,k}(t) + \cdots + \sum_k d_{1,k}\psi_{1,k}(t), \tag{4.116}$$

where J is the number of scales, and k ranges from one to the number of coefficients associated with the specified component (see Example 4.22). In (4.116), the wavelet functions $\phi_{J,k}(t), \psi_{J,k}(t), \psi_{J-1,k}(t), \ldots, \psi_{1,k}(t)$ are generated from the father wavelet, $\phi(t)$, and the mother wavelet, $\psi(t)$, by translation (shift) and scaling:

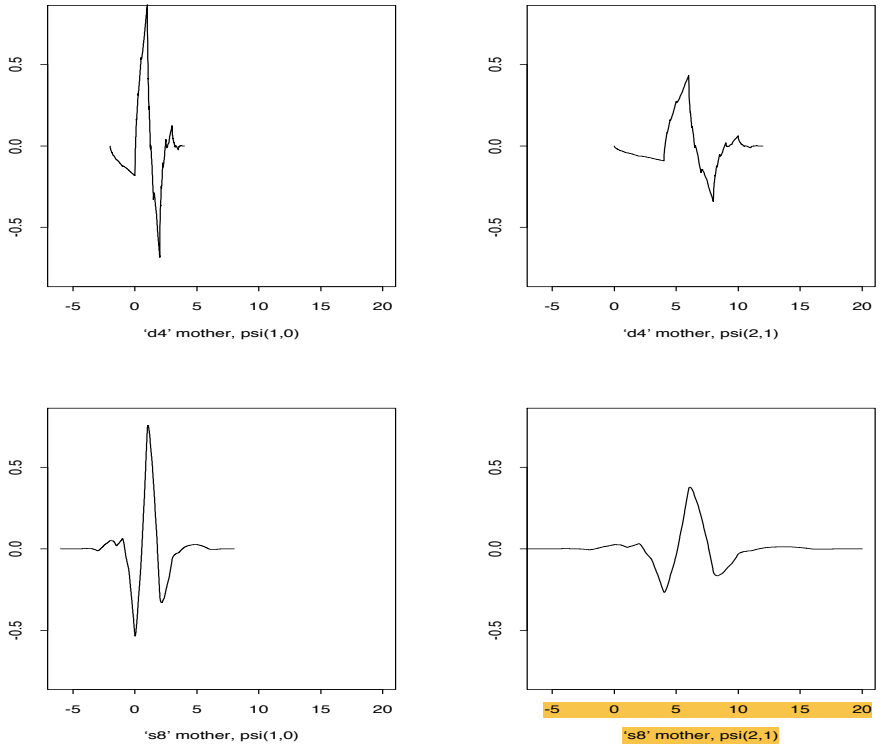

Fig. 4.21. Scaled and translated daublet4 wavelets, $\psi_{1,0}(t)$ and $\psi_{2,1}(t)$ (top row); scaled and translated symmlet8 wavelets, $\psi_{1,0}(t)$ and $\psi_{2,1}(t)$ (bottom row).

$$\phi_{J,k}(t) = 2^{-J/2}\phi\left(\frac{t - 2^J k}{2^J}\right), \tag{4.117}$$

$$\psi_{j,k}(t) = 2^{-j/2}\psi\left(\frac{t - 2^j k}{2^j}\right), \quad j = 1, \ldots, J. \tag{4.118}$$

The choice of dyadic shifts and scales is arbitrary but convenient. The shift or translation parameter is $2^j k$, and scale parameter is 2^j. The wavelet functions are spread out and shorter for larger values of j (or scale parameter 2^j) and tall and narrow for small values of the scale. Figure 4.21 shows $\psi_{1,0}(t)$ and $\psi_{2,1}(t)$ generated from the daublet4 (top row), and the symmlet8 (bottom row) mother wavelets. We may think of $1/2^j$ (or 1/scale) in wavelet analysis as being the analogue of frequency ($\omega_j = j/n$) in Fourier analysis. For example, when $j = 1$, the scale parameter of 2 is akin to the Nyquist frequency of 1/2, and when $j = 6$, the scale parameter of 2^6 is akin to a low frequency ($1/2^6 \approx 0.016$). In other words, larger values of the scale refer to slower, smoother (or coarser) movements of the signal, and smaller values of the scale refer to faster, choppier (or finer) movements of the signal. Figure 4.21 was generated in S-PLUS using the wavelet module as follows:

```
d4.1 <- wavelet("d4", level=1, shift=0)
d4.2 <- wavelet("d4", level=2, shift=1)
```

```
s8.1 <- wavelet("s8", level=1, shift=0)
s8.2 <- wavelet("s8", level=2, shift=1)
par(mfrow=c(2,2))
plot(d4.1, ylim=c(-.8,.8), xlim=c(-6,20))
plot(d4.2, ylim=c(-.8,.8), xlim=c(-6,20))
plot(s8.1, ylim=c(-.8,.8), xlim=c(-6,20))
plot(s8.2, ylim=c(-.8,.8), xlim=c(-6,20))
```

The discrete wavelet transform (DWT) of the data x_t are the coefficients $s_{J,k}$ and $d_{j,k}$ for $j = J, J-1, \ldots, 1$, in (4.116). To some degree of approximation, they are given by[19]

$$s_{J,k} = n^{-1/2} \sum_{t=1}^{n} x_t \phi_{J,k}(t), \tag{4.119}$$

$$d_{j,k} = n^{-1/2} \sum_{t=1}^{n} x_t \psi_{j,k}(t) \quad j = J, J-1, \ldots, 1. \tag{4.120}$$

It is the magnitudes of the coefficients that measure the importance of the corresponding wavelet term in describing the behavior of x_t. As in Fourier analysis, the DWT is not computed as shown but is calculated using a fast algorithm. The $s_{J,k}$ are called the smooth coefficients because they represent the smooth behavior of the data. The $d_{j,k}$ are called the detail coefficients because they tend to represent the finer, more high-frequency nature, of the data.

Example 4.22 Wavelet Analysis of Earthquake and Explosion

Figure 4.22 and Figure 4.23 show the DWTs, based on the symmlet8 wavelet basis, for the earthquake and explosion series, respectively. Each series is of length $n = 2^{11} = 2048$, and in this example, the DWTs are calculated using $J = 6$ levels. In this case, $n/2 = 2^{10} = 1024$ values are in $d1 = \{d_{1,k};\ k = 1, \ldots, 2^{10}\}$, $n/2^2 = 2^9 = 512$ values are in $d2 = \{d_{2,k};\ k = 1, \ldots, 2^9\}$, and so on, until finally, $n/2^6 = 2^5 = 32$ values are in $d6$ and in $s6$. The detail values $d_{1,k}, \ldots, d_{6,k}$ are plotted at the same scale, and hence, the relative importance of each value can be seen from the graph. The smooth values $s_{6,k}$ are typically larger than the detail values and plotted on a different scale. The top of Figure 4.22 and Figure 4.23 show the inverse DWT (IDWT) computed from all of the coefficients. The displayed IDWT is a reconstruction of the data, and it reproduces the data except for round-off error.

Comparing the DWTs, the earthquake is best represented by wavelets with larger scale than the explosion. One way to measure the importance of each level, $d1, d2, \ldots, d6, s6$, is to evaluate the proportion of the total power (or energy) explained by each. The total power of a time series x_t, for

[19] The actual DWT coefficients are defined via a set of filters whose coefficients are close to what you would get by sampling the father and mother wavelets, but not exactly so; see the discussion surrounding Figures 471 and 478 in Percival and Walden (2000).

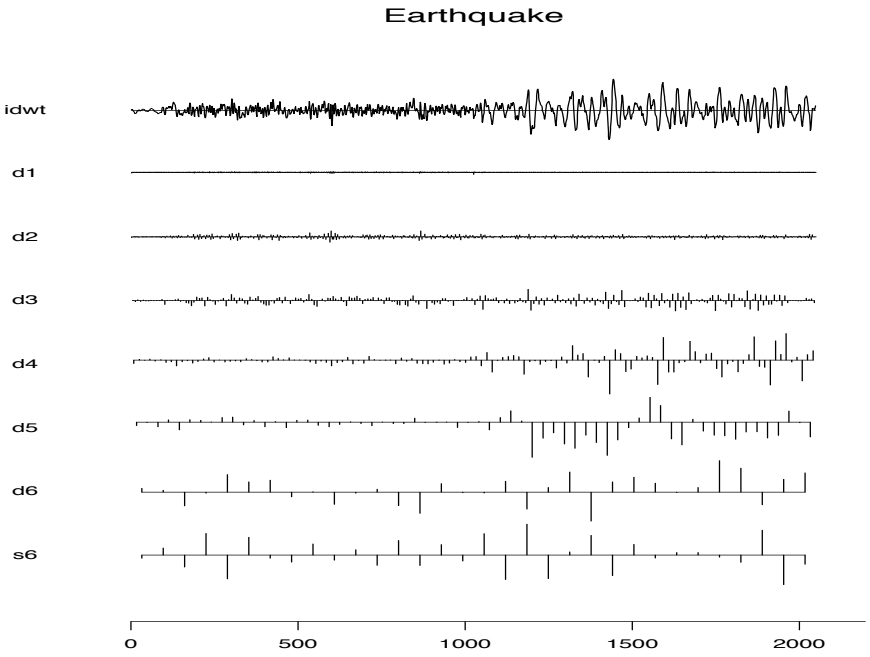

Fig. 4.22. Discrete wavelet transform of the earthquake series using the symmlet8 wavelets, and $J = 6$ levels of scale.

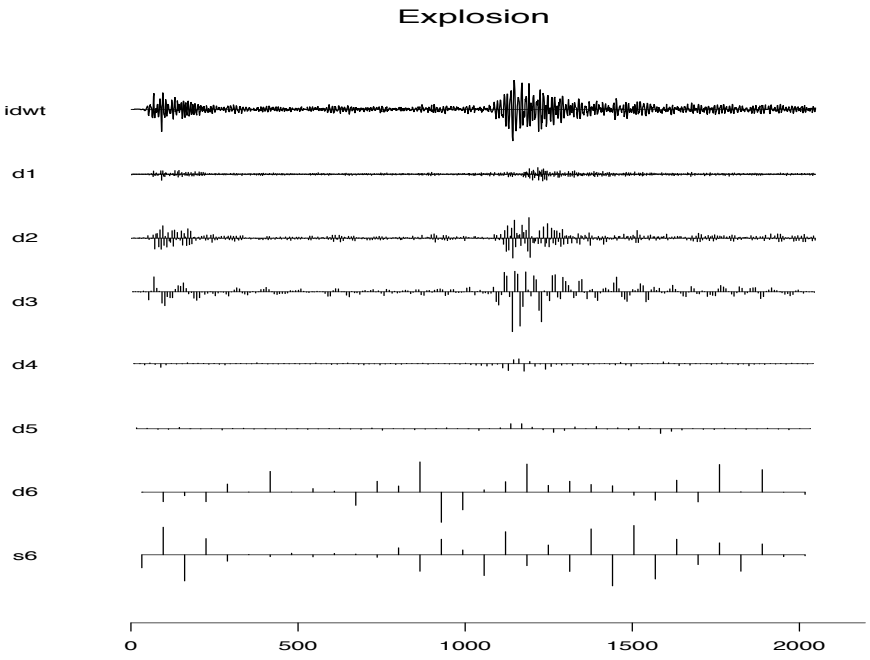

Fig. 4.23. Discrete wavelet transform of the explosion series using the symmlet8 wavelets and $J = 6$ levels of scale.

Table 4.2. Fraction of Total Power

Component	Earthquake	Explosion
s6	0.009	0.002
d6	0.043	0.002
d5	0.377	0.007
d4	0.367	0.015
d3	0.160	0.559
d2	0.040	0.349
d1	0.003	0.066

$t = 1, \ldots, n$, is $TP = \sum_{t=1}^{n} x_t^2$. The total power associated with each level of scale is (recall $n = 2^{11}$),

$$TP_6^s = \sum_{k=1}^{n/2^6} s_{6,k}^2 \quad \text{and} \quad TP_j^d = \sum_{k=1}^{n/2^j} d_{j,k}^2, \quad j = 1, \ldots, 6.$$

Because we are working with an orthogonal basis, we have

$$TP = TP_6^s + \sum_{j=1}^{6} TP_j^d,$$

and the proportion of the total power explained by each level of detail would be the ratios TP_j^d/TP for $j = 1, \ldots, 6$, and for the smooth level, it would be TP_6^s/TP. These values are listed in Table 4.2. From that table nearly 80% of the total power of the earthquake series is explained by the higher scale details $d4$ and $d5$, whereas 90% of the total power is explained by the smaller scale details $d2$ and $d3$ for the explosion.

Figure 4.24 and Figure 4.25 show the time-scale plots (or scalograms) based on the DWT of the earthquake series and the explosion series, respectively. These figures are the wavelet analog of the time-frequency plots shown in Figure 4.17 and Figure 4.18. The power axis represents the magnitude of each value d_{jk} or $s_{6,k}$. The time axis matches the time axis in the DWTs shown in Figure 4.22 and Figure 4.23, and the scale axis is plotted as 1/scale, listed from the coarsest scale to the finest scale. On the 1/scale axis, the coarsest scale values, represented by the smooth coefficients $s6$, are plotted over the range $[0, 2^{-6})$, the coarsest detail values, $d6$, are plotted over $[2^{-6}, 2^{-5})$, and so on. In these figures, we did not plot the finest scale values, $d1$, so the finest scale values exhibited in Figure 4.24 and Figure 4.25 are in $d2$, which are plotted over the range $[2^{-2}, 2^{-1})$.

The conclusions drawn from these plots are the same as those drawn from Figures Figure 4.17 and Figure 4.18. That is, the S wave for the earthquake shows power at the high scales (or low 1/scale) only, and the power remains strong for a long time. In contrast, the explosion shows power at smaller

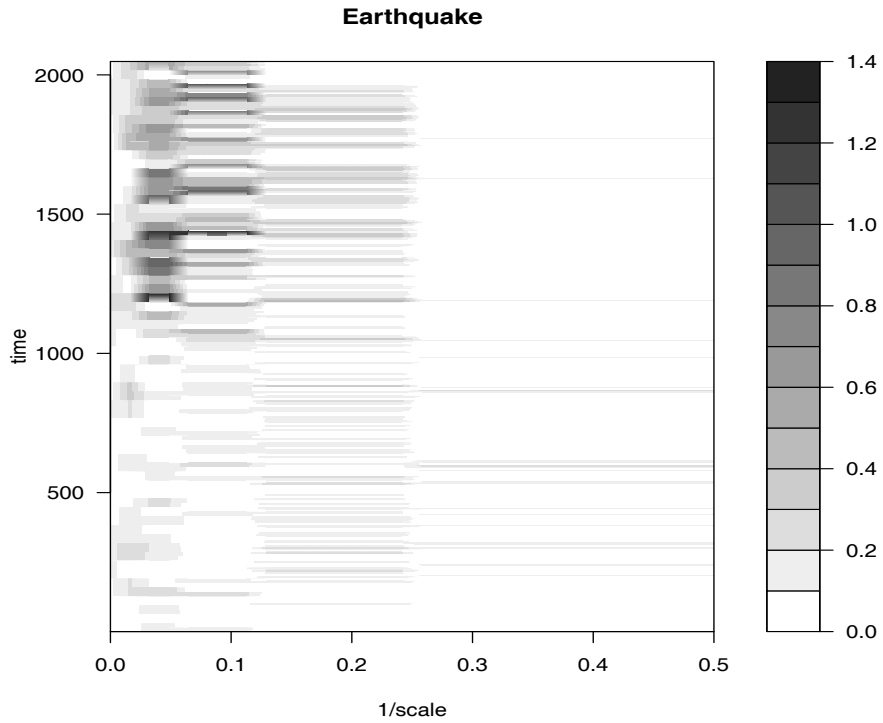

Fig. 4.24. Time-scale image (scalogram) of the earthquake series.

scales (or higher 1/scale) than the earthquake, and the power of the signals (P and S waves) do not last as long as in the case of the earthquake.

Assuming the data files EQ5 and EXP6 have been read into S-PLUS, the analyses of this example can performed using the S-PLUS wavelets module (which must be loaded prior to the analyses) as follows:

```
eq <- scale(EQ5)
ex <- scale(EXP6)
eq.dwt <- dwt(eq)
ex.dwt <- dwt(ex)
plot(eq.dwt)
plot(ex.dwt)
# energy distributions (Table 4.2)
dotchart(eq.dwt)  # a graphic
summary(eq.dwt)  # numerical details
dotchart(ex.dwt)
summary(ex.dwt)
# time scale plots
time.scale.plot(eq.dwt)
time.scale.plot(ex.dwt)
```

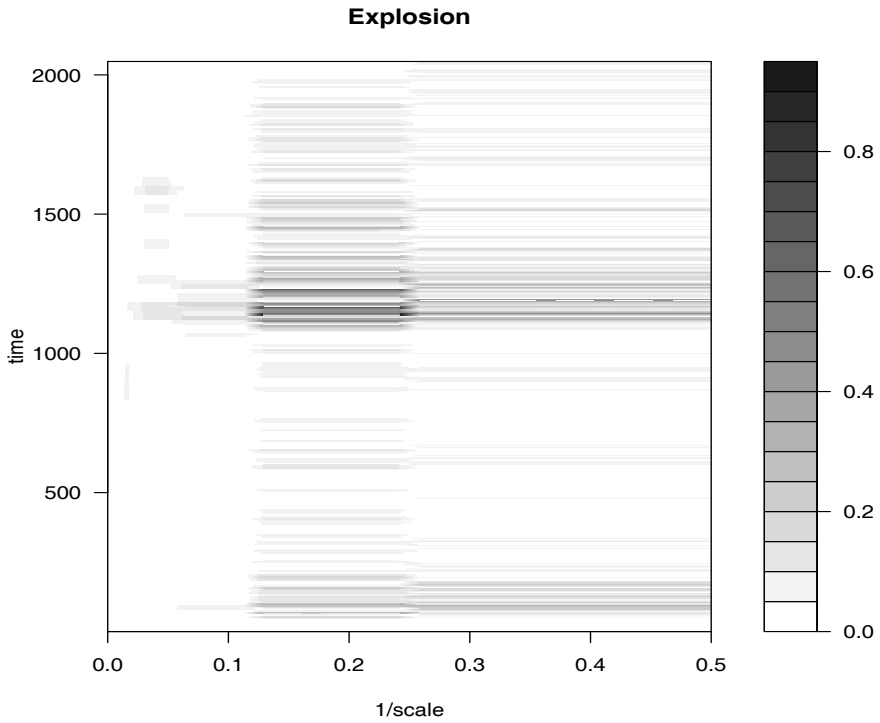

Fig. 4.25. Time-scale image (scalogram) of the explosion series.

Similar analyses may be performed in R using the `wavelets`, `wavethresh`, or `waveslim` packages. We exhibit the analysis for the earthquake series using `wavesthresh`, assuming it has been downloaded and installed.[20]

```
library(wavethresh)
eq = scale(EQ5)  # standardize the series
ex = scale(EXP6)
eq.dwt = wd(eq, filter.number=8)
ex.dwt = wd(ex, filter.number=8)
# plot the wavelet transforms
par(mfrow = c(1,2))
plot(eq.dwt, main="Earthquake")
plot(ex.dwt, main="Explosion")
# total power
TPe = rep(NA,11)  # for the earthquake series
for (i in 0:10){TPe[i+1] = sum(accessD(eq.dwt, level=i)^2)}
TotEq = sum(TPe)  # check with sum(eq^2)
TPx = rep(NA,11)  # for the explosion series
for (i in 0:10){TPx[i+1] = sum(accessD(ex.dwt, level=i)^2)}
TotEx = sum(TPx)  # check with sum(ex^2)
# make a nice table
```

[20] In `wavethresh`, the transforms are denoted by the resolution rather than the scale. If the series is of length $n = 2^p$, then resolution $p - i$ corresponds to level i for $i = 1, \ldots, p$.

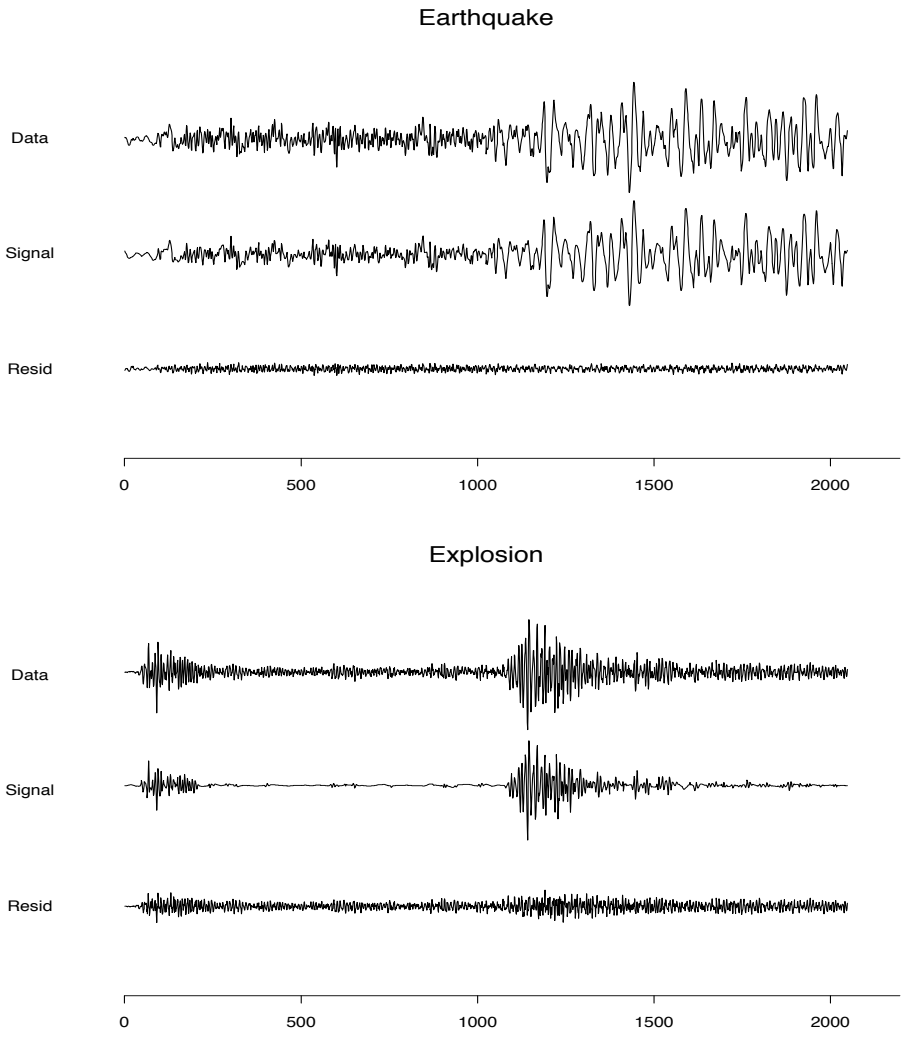

Fig. 4.26. Waveshrink estimates of the earthquake and explosion signals.

```
Power = round(cbind(11:1, 100*TPe/TotEq, 100*TPx/TotEx), digits=3)
colnames(Power) = c("Level", "EQ(%)", "EXP(%)")
Power
```

Wavelets can be used to perform nonparametric smoothing along the lines first discussed in §2.4, but with an emphasis on localized behavior. Although a considerable amount of literature exists on this topic, we will present the basic ideas. For further information, we refer the reader to Donoho and Johnstone (1994, 1995). As in §2.4, we suppose the data x_t can be written in terms of a signal plus noise model as

$$x_t = s_t + \epsilon_t. \tag{4.121}$$

The goal here is to remove the noise from the data, and obtain an estimate of the signal, s_t, without having to specify a parametric form of the signal. The technique based on wavelets is referred to as waveshrink.

The basic idea behind waveshrink is to shrink the wavelet coefficients in the DWT of x_t toward zero in an attempt to denoise the data and then to estimate the signal via (4.116) with the new coefficients. One obvious way to shrink the coefficients toward zero is to simply zero out any coefficient smaller in magnitude than some predetermined value, λ. Such a shrinkage rule is discontinuous and sometimes it is preferable to use a continuous shrinkage function. One such method, termed soft shrinkage, proceeds as follows. If the value of a coefficient is a, we set that coefficient to zero if $|a| \leq \lambda$, and to $\text{sign}(a)(|a| - \lambda)$ if $|a| > \lambda$. The choice of a shrinkage method is based on the goal of the signal extraction. This process entails choosing a value for the shrinkage threshold, λ, and we may wish to use a different threshold value, say, λ_j, for each level of scale $j = 1, \ldots, J$. One particular method that works well if we are interested in a relatively high degree of smoothness in the estimate is to choose $\lambda = \widehat{\sigma}_\epsilon \sqrt{2 \log n}$ for all scale levels, where $\widehat{\sigma}_\epsilon$ is an estimate of the scale of the noise, σ_ϵ. Typically a robust estimate of σ_ϵ is used, e.g., the median of the absolute deviations of the data from the median (MAD). For other thresholding techniques or for a better understanding of waveshrink, see Donoho and Johnstone (1994, 1995), or the S-PLUS wavelets module manual (Bruce and Gao, 1996, Ch 6).

Example 4.23 Waveshrink Analysis of Earthquake and Explosion

Figure 4.26 shows the results of a waveshrink analysis on the earthquake and explosion series. In this example, soft shrinkage was used with a universal threshold of $\lambda = \widehat{\sigma}_\epsilon \sqrt{2 \log n}$ where $\widehat{\sigma}_\epsilon$ is the MAD. Figure 4.26 displays the data x_t, the estimated signal $\widehat{s}_t$, as well as the residuals $x_t - \widehat{s}_t$. According to this analysis, the earthquake is mostly signal and characterized by prolonged energy, whereas the explosion is comprised of short bursts of energy.

Figure 4.26 was generated in S-PLUS using the wavelets module. For example, the analysis of the earthquake series was performed as follows.

```
eq.dwt <- dwt(eq)
eq.shrink <- waveshrink(eq.dwt, shrink.rule="universal",
    shrink.fun="soft")
```

In R, using the `wavethresh` package, use the following commands for the earthquake series.

```
library(wavethresh)
eq = scale(EQ5)
par(mfrow=c(3,1))
eq.dwt = wd(eq, filter.number=8)
eq.smo = wr(threshold(eq.dwt, levels=5:10))
ts.plot(eq, main="Earthquake", ylab="Data")
ts.plot(eq.smo, ylab="Signal")
ts.plot(eq-eq.smo, ylab="Resid")
```