User Manual for FAWAV

James S. Walker
Department of Mathematics
University of Wisconsin–Eau Claire
Eau Claire, WI 54702–4004

E-mail: walkerjs@uwec.edu

Web page: http://www.uwec.edu/walkerjs/

Contents

Chapter 1

Introduction

FAWAV is designed as an aid to signal processing of 1D signals (especially audio) and 2D signals (especially images). I originally designed FAWAV as a teaching aid for my courses on Digital Signal Processing and Digital Image Processing, and as a tool for my own research. But, I felt that it was good enough to make available as free software for others to use. I am pleased that other researchers at institutions such as U.C. Berkeley, the University of Barcelona, the University of North Dakota, and Washington University-St. Louis, have made use of FAWAV. If you have any questions about FAWAV please contact me at the email address given on the title page. Or, if you have used FAWAV in your own research I would love to hear about that (I always enjoy receiving preprints and papers).

1.1 Components of FAWAV

FAWAV has several major components. These components (listed on the *File* menu, see Figure 1.1) are

1. *New 1Dim* Choose this item for processing 1D signals. See Chapter 2 on p. 11.
2. *New 2Dim* Choose this item for processing 2D signals. See Chapter 5 on p. 59.
3. *New Audio editor* Choose this item for loading audio signals and selecting portions to process. See Chapter 6 on p. 85.
4. *New Image Processor* Choose this item for loading gray-scale images for various image processing operations, such as compression and denoising. See Chapter 7 on p. 89.

In addition to these main items, there also several choices for opening various types of files. These latter choices will be discussed in the chapters that follow, as they are best understood in the context of the various processing windows that open up to handle these various types of files.

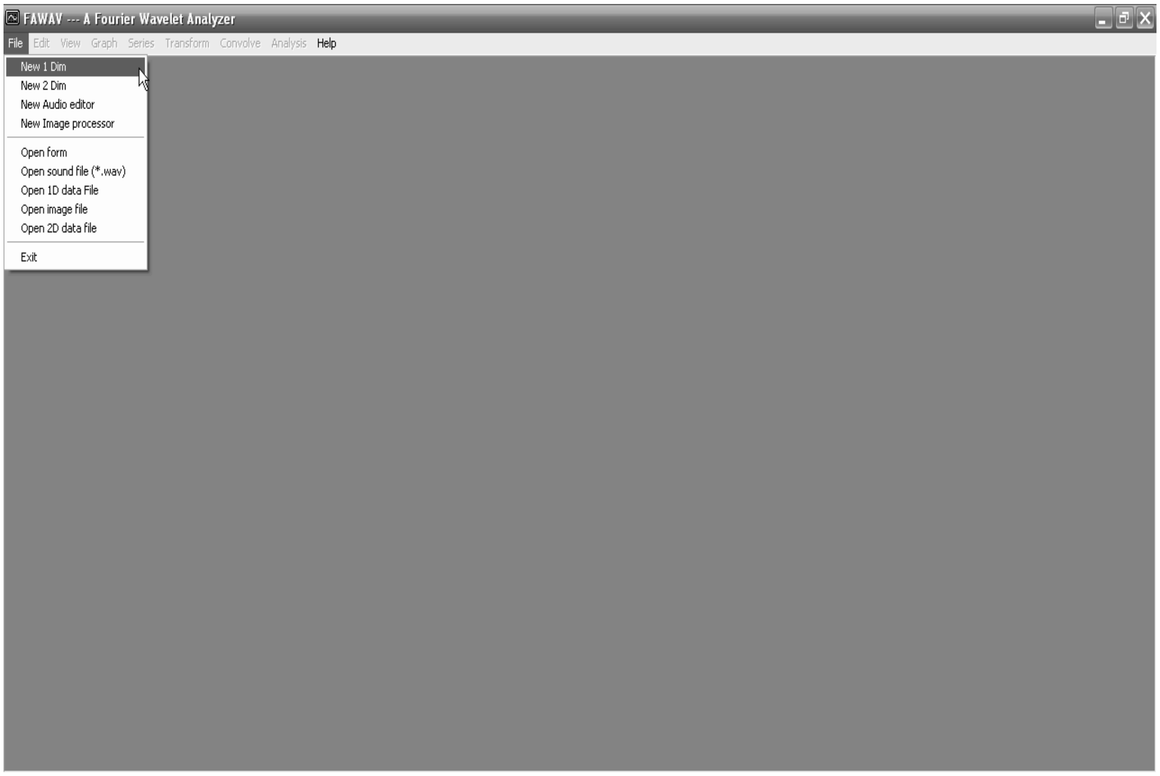


Figure 1.1
Initial menu of FAWAV.

Chapter 2

Processing 1D Signals (1D-form)

To work with 1D signals, click on the *File* menu, and then select *New 1 Dim.* The window shown in Figure 2.1 appears. A description of each of the 9 menu items will be given at the end of this section (see Section 2.11 on p. 33). But first we describe how to use this form for basic signal processing. The procedures that are available for working with 1D signals include the following:

1. *Graphing formulas and external data.* You can plot graphs of elementary functions studied in algebra and calculus (see Section 2.1 on p. 12.) Or, you can load external data (see Section 2.1.12 on p. 19). Or, you can load audio

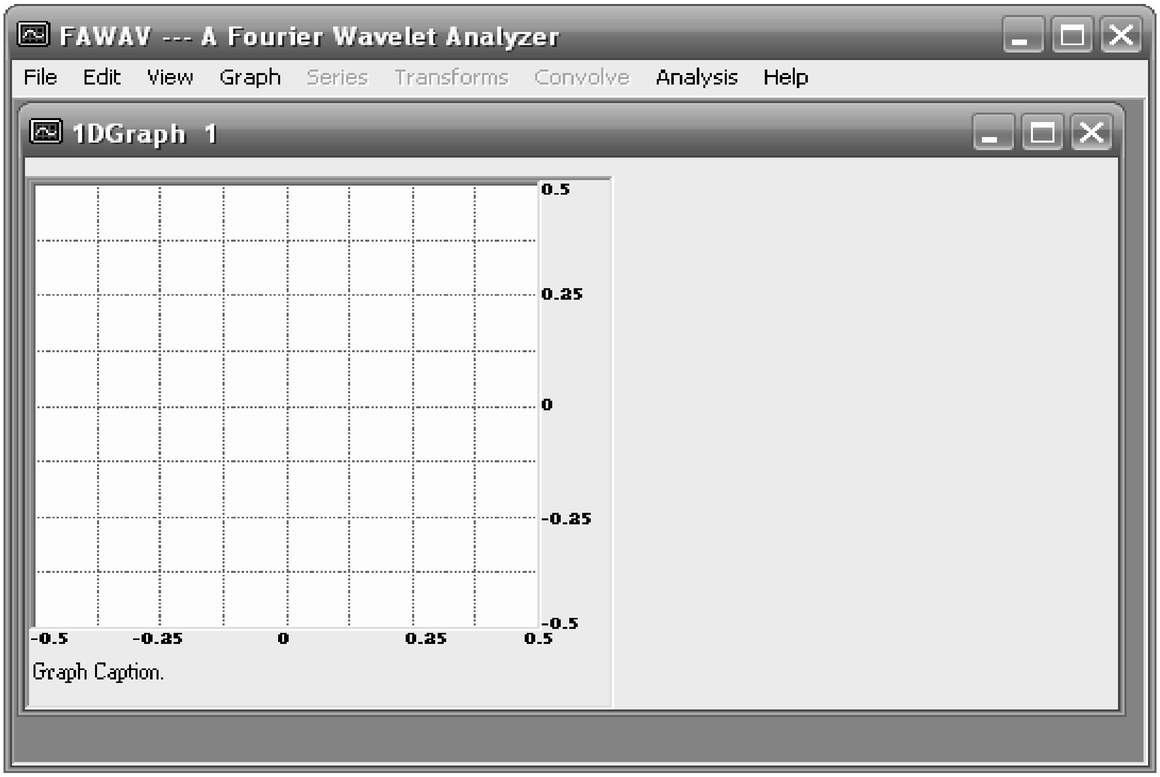


Figure 2.1
The window that opens when *New 1D-form* is selected. (Graph area displayed with white background to save ink).

files (see Section 2.1.11 on p. 19).

2. *Wavelet transforms.* These are the Haar, DaubJ, CoifI transforms, and biorthogonal wavelet transforms, described in *A Primer on Wavelets and their Scientific Applications*. See Section 2.2 on p. 21.

3. *Wavelet series.* These are the Haar, DaubJ, CoifI series, and a selection of biorthogonal wavelets series, described in *A Primer on Wavelets and their Scientific Applications* and its *Examples & Exercises* manual. See Section 2.3 on p. 21.

4. *Fourier series and transforms.* These are the classic Fourier series and transforms implemented using FFTs. See Section 2.6 on p. 23.

5. *Audio.* The 1D signals can be played over your computer's sound system. This is especially helpful for judging the effectiveness of compression and noise removal. See Section 2.8 on p. 31 for a discussion of the simple *Audio Editor* that is part of FAWAV, and Section 2.1.11 on p. 19 for instructions on how to load audio files for signal processing.

6. *Convolutions and Correlations.* There are procedures for calculating the convolution of two signals, for calculating the correlation between two signals, and for calculating an auto-correlation of one signal. See Section 2.9 on p. 31.

7. *Scalograms and spectrograms.* Two advanced methods for creating time-frequency portraits of 1D signals are scalograms and spectrograms. See Section 2.10 on p. 33.

There are also a number of routines for numerical analysis of 1D signals, these are grouped under the *Analysis* choice on the menu (see Section 2.11.8 on p. 36).

2.1 Graphing formulas and external data

Data can be plotted in the 1D-form in one of three ways: (1) plotting an elementary formula from algebra or calculus, (2) loading a sound file, (3) loading an external data file. We describe each of these methods in order.

2.1.1 Plotting a formula

To plot an elementary formula you select *Graph,* followed by *Plot* from the menu. A dialog box opens up on the right of the window, with a text box at the top for entering your formula. To graph a formula over an interval, perform the following steps:

1. Enter a formula in the text box at the top of the dialog box. The simplest way to begin graphing formulas is to use the kind that graphing calculators use. For example, the following two formulas:

   ```
   3x + 7

   4sin(3pi x-2) - 5cos(4pi x-5)
   ```

   are both valid formulas in FAWAV. The second formula uses the expression `pi` for the constant π. For many more examples, see Section 2.1.2 on p. 13 below.

2. Select an *Interval type,* by selecting the option $[-L, L]$ or the option $[0, L]$.

3. Specify the value of L to be used for the interval endpoint(s). For example, if you specify that $L = 2$ and you selected the option $[-L, L]$ then the interval used for graphing will be $[-2, 2]$. Or, if you selected [0,L] then the interval will be [0,2].

4. Click on the command button labeled *Plot.* Assuming all the data are correctly specified, this will produce a graph in the graph area on the left of the window.

Formula command buttons There are five command buttons at the bottom of the formula text box. Here are descriptions of each of these command buttons.

> *Previous* If you have graphed more than one formula, then by clicking on this button you can recall previously entered formulas.
>
> *Next formula* If you have clicked on the *Previous* button one or more times, then you can go forward in the list of formulas by clicking on this button. If you have not clicked the *Previous* button, then clicking this button will clear the text box completely and you can enter a new formula.
>
> *Load* Click on this button in order to retrieve a formula that you previously saved.
>
> *Save* Click on this button in order to save the formula that is presently displayed in the text box.
>
> *Help* Click on this button in order to get help on creating formulas.

2.1.2 Formulas for 1-dimensional graphs

Formulas for graphing are constructed using pre-defined functions, as well as various logical operations, or sums, or products. In Sections 2.1.3 through 2.1.10 we discuss all of the operations that can be employed in graphing formulas.

2.1.3 Functions supported

There are a number of pre-defined functions. We list them here in two categories: (1) the *elementary functions* of the kind encountered in calculus, and (2) certain *special functions* of a miscellaneous nature.

Elementary functions

FAWAV employs 11 elementary functions, which we now list.

1. `abs` The absolute value function, `abs( )`, returns the absolute value of its argument, e.g. $\texttt{abs}(-5) = 5$ and $\texttt{abs}(5) = 5$.

2. `cos` The cosine function is invoked using `cos( )`. For example,

   ```
   cos(3x)

   cos(xˆ2 - 3) + 6x - 3
   ```

   are both valid formulas involving the cosine function.

3. `cosh` The hyperbolic cosine function, `cosh( )`, returns the hyperbolic cosine of its arguments. By definition, `cosh(x)` satisfies

   ```
   cosh(x)  =  [exp(x) + exp(-x)]/2
   ```

   where `exp` is the exponential function.

4. `exp` The exponential function `exp( )` is the standard, base e, exponential function. For example,

   ```
   exp(4x)

   eˆ(4x)
   ```

   are two equivalent expressions using the exponential function.

5. `gri` The greatest integer function `gri( )` is defined as follows:

$$\texttt{gri(x)} = \texttt{largest integer that is} \leq \texttt{x}$$

Or, an equivalent definition is

$$\texttt{gri(x)} = \texttt{closest integer to the left of x on the number line}$$

The greatest integer function can be used to compute remainders in long division (also called Mod functions) and to compute periodic functions.

Mod functions. To compute the remainder of P divided by Q, where P and Q are positive integers and $P \geq Q$, you can use the greatest integer function in the following way: $\mathrm{Remainder}(P/Q) = P - Q\,\mathrm{gri}(P/Q)$. The remainder of P/Q is often denoted $P \mod Q$. Hence $P \mod Q = P - Q\,\mathrm{gri}(P/Q)$.

Periodic functions. The function `gri( )` can be used to create periodic functions. For example, the following formula

```
uˆ2\u=x-wgri(x/w)
```

defines a periodic extension of $\mathtt{x}^2$ having period `w`, where $\mathtt{x}^2$ is periodically extended from the initial interval $[0, \mathtt{w}]$. This formula uses *substitutions* for the variables `v` and `u`; see Section 2.1.10 on p. 18 for further discussion of substitution. As a second example, the following formula

```
cos(pi t)\t=u-w/2\u=v-wgri(v/w)\v=x+w/2
```

defines a periodic extension of $\cos(\pi\mathtt{x})$ having period `w`, where $\cos(\pi\mathtt{x})$ is periodically extended from the initial interval [-$\mathtt{w}/2, \mathtt{w}/2$].

6. `log` The natural logarithm function is defined by `log( )`. You can also use `Ln( )`.

7. `sgn` The sign function, `sgn( )`, is defined by

$$\texttt{sgn(x)} = \begin{cases} 1 & \text{if } \texttt{x > 0} \\ 0 & \text{if } \texttt{x = 0} \\ -1 & \text{if } \texttt{x < 0} \end{cases}$$

You can also use `sign( )` for the sign function.

One important use of the sign function is in defining the odd extension of a function. For example, suppose you graph the function

```
cos(x) + 3
```

over the interval $[0, 4]$. Then graph

```
sign(x)(cos(abs(x)) + 3)
```

over $[-4, 4]$. This second function is the odd extension of the first function.

8. `sin` The sine function is invoked using `sin( )`. For example,

```
sin(3x)

sin(5x - 6) + x - 2
```

are both valid formulas involving the sine function.

9. `sinh` The hyperbolic sine function, `sinh( )`, returns the hyperbolic sine of its arguments. By definition, `sinh(x)` satisfies

   ```
   sinh(x)  =  [exp(x) - exp(-x)]/2.
   ```

10. `sqr` The positive square root function, `sqr( )`, computes the positive square root of its argument. For example, `sqr(4) = 2` and `sqr(9) = 3`. Also `sqr(0) = 0`. The square roots of negative numbers are not real, so in FAWAV applying the `sqr` function to a negative number produces an error message.

11. `tan` The tangent function is invoked using `tan( )`. For example,

    ```
    tan(3x)

    tan(5x - 6) + x - 2
    ```

    are both valid formulas involving the tangent function.

Special functions

FAWAV employs 7 special functions, which we shall now describe.

1. `bell` The function `bell( )` is defined as follows. If $|\mathtt{x}| \geq 0.75$, then `bell(x) = 0`. If $|\mathtt{x}| \leq 0.25$, then `bell(x) = 1`. Finally, if $0.25 < |\mathtt{x}| < 0.75$, then `bell(-x) = bell(x)` (that is, the bell function is even), and we have for $0.25 < \mathtt{x} < 0.75$, that `bell(x)` is defined by

   ```
   bell(x) = sin(aw)sin(aw)\a=0.5pi\w=sin(av)sin(av)
                \v=sin(au)sin(au)\u=1.5 - 2x
   ```

   Here are 4 examples that use the `bell` function.

   (a) Graph the function `bell(x)` over the interval $[-4, 4]$. Use *Analysis/Trace* (see p. 38) to check that `bell(x) = 1`, for $|\mathtt{x}| \leq 0.25$ and `bell(x) = 0`, for $|\mathtt{x}| \geq 0.75$.

   (b) Graph the function `bell(x/4)` over the interval $[-4, 4]$. Use *Analysis/Trace* to check that this function equals 1 over $[-1, 1]$ and equals 0 for $|\mathtt{x}| \geq 3$.

   (c) Graph the function `sumk(bell(x-k))\k=-3,3` over the interval [-4,4]. (This formula uses a sum command, see Section 2.1.6 on p. 17 for further discussion.) Use *Analysis/Trace* to check that this function equals 1 over the interval $[-3.25, 3.25]$ and equals 0 for $|\mathtt{x}| \geq 3.75$.

   (d) Graph the function `sumk(bell(x-k))sin(pi x)\k=-5,5` over the interval $[-8, 8]$. Then graph `sin(pi x)` over the same interval to check that the first function equals `sin(pi x)` over the interval $[-5.25, 5.25]$ and equals 0 for $|\mathtt{x}| \geq 5.75$.

2. `del` The discrete delta function, `del( )`, is defined as follows

   $$\mathtt{del}(\mathrm{x}) = \begin{cases} 0 & \text{if } \mathtt{|x| > 0} \\ 1 & \text{if } \mathtt{x = 0} \end{cases}$$

   In other words, `del(x)` is 0 if `x < 0` or `x > 0`, and `del(0)` equals 1.

3. `fac` The factorial function, `fac( )`, returns the factorial of a non-negative integer between 0 and 70. For example, `fac(5) = 120`. If the argument of `fac( )` is not an integer, then an error message is returned. A formula such as

```
sumk(x^k / fac(k)) \k = 0, 8
```

is allowed, and returns a partial sum of the Taylor series for $e^{\mathtt{x}}$ at $\mathtt{x} = 0$.

4. `ran` The random number function, `ran( )`, gives a value that is a random number between -1 and 1, with a mean value of 0. Random values are assigned no matter what the argument of `ran( )` is.

   For instance, the formula

   ```
   ran(0)
   ```

   will not produce a constant function. It will produce a randomly varying function with values between -1 and 1. The values of these random numbers are approximately uniformly distributed, i.e., the values of `ran( )` are approximately the values of a random variable with uniform probability density.

5. `rang` The random number function, `rang( )`, gives values that are approximately distributed like that of a Gaussian random variable. Random values are assigned no matter what the argument of `rang( )` is.

   For instance, the formula

   ```
   rang(0)
   ```

   will not produce a constant function. It will produce a randomly varying function.

   The function `rang( )` is used to create an approximation of random noise with a Gaussian normal probability. This type of random noise is called Gaussian white noise. The function `rang( )` is produced by adding N repetitions of `ran( )` and dividing by the square root of N. This number N is 144 by default. To change the number N of terms, enter a formula like the following

   ```
   rang(x)\noiseterms=400
   ```

   In this case N is set equal to 400. You must specify a value for N between 1 and 30000.

6. `rect` The rectangle function, `rect( )`, is defined by

   $$\mathtt{rect(x)} = \begin{cases} 1 & \text{if } \mathtt{|x|\ <\ 0.5} \\ 0.5 & \text{if } \mathtt{|x|\ =\ 0.5} \\ 0 & \text{if } \mathtt{|x|\ >\ 0.5} \end{cases}$$

   For example, to create a function equal to 1 over the interval (a, b) and equal to 0 outside that interval, you enter the formula

   ```
   rect((x-c)/w)\c=(a+b)/2\w=b-a
   ```

7. `sinc` The sinc function, `sinc( )`, is defined by

   $$\operatorname{sinc}(x) = \frac{\sin \pi x}{\pi x}$$

   for x not equal to zero. If $x = 0$, then $\operatorname{sinc}(0) = 1$.

2.1.4 Piecewise functions

To illustrate how piecewise functions are graphed, consider the following two examples.

1. To graph the piecewise function

$$f(x) = \begin{cases} x-3 & \text{for } x < -2 \\ 2x+8 & \text{for } x \geq -2 \end{cases}$$

   enter the formula

```
(x - 3)(x < -2) + (2x + 8)(x >= -2)
```

2. To graph the piecewise function

$$f(x) = \begin{cases} 2 & \text{for } x \leq -\pi \\ \cos x & \text{for } -\pi < x < 2\pi \\ -3 & \text{for } x \geq 2\pi \end{cases}$$

   enter the formula

```
2(x <= -pi) + cos(x)(-pi < x < 2pi) - 3(x >= 2pi)
```

Using the *Dots* option for graphing[1] is often the best way to plot piecewise functions.

The formulas in these two examples made use of *Logic* and *Inequalities.* For further discussion of these topics see Section 2.1.8 on p. 18, and Section 2.1.9 on p. 18, respectively.

2.1.5 Graph operations

If you have drawn the graph of at least one function on the screen, then you can assign values from these function(s) using the graph variables `g1( )`, `g2( )`, `g3( )`, `g4( )`, and `g5( )`.

For example, suppose 3 graphs are displayed on screen. Then the following formula:

```
2g1(x) - 4g3(x)
```

assigns twice the values of graph 1 minus four times the values of graph 3 to the plotted function.

Warning: Do NOT use a graphical function with a higher index than the number of displayed graphs. Also, FAWAV does not support graph variables beyond `g5( )`. That is, if there are 6 or more graphs displayed, then `g6( )` or higher graph indices, will not be accepted.

2.1.6 Sums

There are four sum operators, `sumi( )`, `sumj( )`, `sumk( )`, and `sumn( )`. For example, the formula

```
sumi(xˆi)\i=1,10
```

produces the same function as the more complicated formula

```
x + xˆ2 + xˆ3 + xˆ4 + xˆ5 + xˆ6 + xˆ7 + xˆ8 + xˆ9 + xˆ10
```

Or, the following formula

```
1 + 2sumk(cos(2(k+1)pi x))\k=0,19
```

[1] Select *View/Display style* and then select the option *Dots*.

produces the same function as

$$1 + 2\cos(2\pi x) + 2\cos(4\pi x) + \cdots + \cos(40\pi x).$$

Warning: You cannot use a sum operator twice in the same formula. For example, the formula

```
sumk(cos(kx)) + 8e^(2x-3) - sumk(kx-3)\k=3,5
```

will not be accepted by FAWAV. You should use instead

```
sumk(cos(kx)-kx+3) + 8e^(2x-3)\k=3,5
```

2.1.7 Products

There are two operators for performing products, `prodp( )`and `prodq( )`. For example, the following formula

```
prodp(x-p)\p=-4,3
```

produces the same graph as the following more complicated formula

```
(x+4)(x+3)(x+2)(x+1)x(x-1)(x-2)(x-3)
```

Warning: You cannot use a product operator twice in the same formula. For example, the formula

```
prodp(x-p) - prodp(px)\p=3,5
```

will not be accepted by FAWAV. You should use instead

```
prodp(x-p) - prodq(qx)\p=3,5\q=3,5
```

2.1.8 Logic

There are three logical operations, `&` (and), `or`, and `not( )`. These are Boolean operations which are meant to return values of either 1 (for TRUE) or 0 (for FALSE). For example, the following formula

```
(x>3 or x<-2)sin(3x) + ((-2<=x)&(x<=3))cos(x)
```

produces a function that equals $\sin(3\mathtt{x})$ if $\mathtt{x} > 3$ or if $\mathtt{x} < -2$, and equals $\cos(\mathtt{x})$ if $-2 \leq \mathtt{x} \leq 3$.

2.1.9 Inequalities

The inequality operations are `>`, `>=`, `<`, and `<=`. For example, if you enter the formula

```
(2<x<=4)sin(x) + (x>4)cos(x)
```

the function will equal 0 for $\mathtt{x} \leq 2$, will equal $\sin(\mathtt{x})$ for $2 < \mathtt{x} \leq 4$, and will equal $\cos(\mathtt{x})$ for $\mathtt{x} > 4$.

2.1.10 Command strings

Command strings are used at the end of a formula for performing substitutions or other tasks. They are preceded by the backslash character `\`. Here are a few examples of using command strings.

1. *Substitution.* The following formula shows how a command string performs several substitutions:

   ```
   cos(ax) + bsin(cx)
   \a=2\b=x+c\c=3x
   ```

Notice that the parameters can be constant or variable and that they can invoke other parameters as well.

The allowed parameters are `a`, `b`, `c`, `m`, `t`, `u`, `v`, `w`, `z`. Unless a differential equation is specified (see below), the parameter `y` can also be used.

2. *Limits for a sum.* The following formula shows how a command string is used for specifying the limits for a sum:

   ```
   1/2 + sumk[cos(kx)]\k=-4,7
   ```

   Here the sum is from $\texttt{k} = -4$ to $\texttt{k} = 7$.

3. *Differential equation mode.* A command string can be used to tell FAWAV to solve certain types of differential equations. For example, the differential equation

   $$y' + xy = 4\cos(2x), \quad y(0) = -3$$

   can be solved over the interval $[0, 20]$ in the following way. First, select *Graphs/Plot* and then specify an interval of type $[0, L]$ with $L = 20$. Then enter the following formula:

   ```
   -xy + 4cos(2x)\Diff\y=-3
   ```

   The command string `\Diff` tells FAWAV that the formula requires the numerical solution of a differential equation, and the command string `\y=-3` says to use the initial value of $y(0) = -3$. FAWAV uses the classic Runge-Kutta, order 4, method for approximating solutions.

   You can also plot approximations to solutions of higher order differential equations. You use the variable `y1` for the first derivative `y'` and `y2` for `y''` and so on. For example, graphing the following

   ```
   -y \Diff\Ord=2\y=1\y1=0
   ```

   over the interval $[0, 10]$, approximates the solution of $y'' = -y$ with initial conditions $y(0) = 1$ and $y'(0) = 0$. You can approximate a differential equation up to order 10.

2.1.11 Loading an audio file

FAWAV supports the loading and playing of mono `.wav` files. To load a `.wav` audio file, just click on the graph box on the left side of the window with the right mouse button. Then select *Load/Sound file* from the pop-up menu. A standard *Open File* dialog box will appear from which you can select a `.wav` file for loading.

Note: If the `.wav` file that you choose to open is too large (larger than 2^{16} samples), to load directly into the *1D-Form*, the file will be loaded into the *Audio editor* that is part of FAWAV. From this Audio editor you can select a portion of the audio file for analysis. (See Chapter 6 on p. 85 for more details.)

You can also load a sound file from the *File* menu [using *Open sound file (*.wav)*], but the method described above is preferable.

FAWAV only supports mono `.wav` files. Another common sound file format is `.mp3`. To convert a mono `.mp3` file to a `.wav` file use the AUDACITY program. Load an `.mp3` file into AUDACITY and then select *File/Export as WAV* to save to a `.wav` file. If the `.mp3` file is a stereo file, then select one track in AUDACITY and remove the other track, and then export the single track to a `.wav` file.

2.1.12 Loading external (binary) files

There are two ways to load external data. The easiest way (which we recommend) is to enter the data in a spreadsheet (such as EXCEL) or text editor (such as NOTEPAD) in the following format:

```
(number of points)
(left endpoint)
(right endpoint)
(first function value)
(second function value)
(third function value)
      .
      .
      .
(last function value)
```

The (number of points) must be one of the following numbers:

```
64,  128,  256,  1024,  2048,  4096,  8192,  16384,  32768,  65536.
```

The next two numbers are the left and right endpoints of the interval over which the data are to be plotted. There are two acceptable formats for these two numbers: either 0 and L, or $-L$ and L. For example, the endpoints should be 0 and then 10 when the interval is $[0, 10]$; or, they should be -5 and then 5 when the interval is $[-5, 5]$. After the endpoints, the remainder of the data are the values of the signal. There should be one more value than the value of (number of points). For example, if 1024 points are specified, then there should be 1025 values specified.

Here is a sample listing:

```
128
0
4
2.35
1.357
  .
  .
  .
2.578
```

The first three numbers specify 128 points over an interval of $[0, 4]$. The remaining numbers are the signal values.

Note again that in this example there should be 129 values following the right endpoint of 4, not 128. This quirky requirement of the data format is necessary because of the way FAWAV plots functions and computes Fourier series and Fourier transforms. [Note: If you supply fewer function values than specified by the (number of points) value, then FAWAV supplies 0-values for the extra function values that were not supplied.]

After entering your data, you then select it and copy it to the WINDOWS Clipboard, and then right-click on the FAWAV 1D-form and select *Paste from Clipboard.* Your data should then be graphed.

There is a second way to load data (we recommend the first method, but if you are a very experienced programmer you could use this second method). You right-click on the graph box on the left side of the window, and then select *Load/Graph.* From the dialog box that appears you then select a file of binary data to load. If you want to load data of your own (not previously saved from within FAWAV), then the data file must have the following format. It must end with the extension `.fb1` and consist of 8-byte double precision numbers (except for the number of points, which is a 4-byte integer). The first number specifies the number of points. The number of points must be one of the following 10 numbers:

$$64 \quad 128 \quad 256 \quad 1024 \quad 2048 \quad 4096 \quad 8192 \quad 16384 \quad 32768 \quad 65536.$$

The number of points is saved as a 4-byte integer (sometimes referred to as a long integer). The next two numbers are the left and right endpoints of the interval over which the data are to be plotted. There are two acceptable formats for these two numbers: either 0 and L, or $-L$ and L. For example, the endpoints should be 0 and then 10 when the interval is $[0, 10]$; or, they should be -5 and then 5 when the interval is $[-5, 5]$. These two endpoints are specified as 8-byte double precision numbers. After the endpoints, the time-increment is specified as an 8-byte double precision number.

This time-increment is equal to the length of the interval divided by the number of points. For example, if the interval is $[-5, 5]$ and 1024 points are used, then the time-increment would be 0.009765625.

After the time-increment, the remainder of the data are the values of the signal (all 8-byte double precision numbers). There should be one more value than the number of points listed above. For example, if 1024 points are specified, then there should be 1025 values specified. Here is a sample format for an `.fb1` file:

```
128
0
4
0.03125
0.025
1.357
  .
  .
  .
2.578
```

The first four numbers specify 128 points over an interval of $[0, 4]$, with a time increment of 0.03125. The remaining numbers are the signal values. Note again that in this example there should be 129 values following the right endpoint of 4, not 128. This quirky requirement of the `.fb1` format is necessary because of the way FAWAV plots functions and computes Fourier series and Fourier transforms.

2.2 Wavelet transforms

A wavelet transform of a displayed graph can be calculated by performing the following steps.

1. You select *Transforms/Wavelet.*

2. In the text box labeled *Graph,* enter the number of the graph that you want to transform.

3. In the text box labeled *Levels,* enter the number of levels to use for the transform. By resting the mouse pointer on this text box, a message should appear which tells you the allowed number of levels.

4. If you want to compute an inverse transform, then check the checkbox labeled *Inverse.* Otherwise, make sure this checkbox is not checked.

5. Select the type of wavelet you want to use. The choices are Coif, Daub, Haar, and a selection of biorthogonal wavelets. If you select Coif or Daub, then you must also enter the appropriate number at the left of the wavelet type. By resting the mouse pointer on the text box next to the wavelet type, a message should appear which tells you the allowed numbers.

6. Click on the button labeled *Plot.* This will produce the wavelet transform.

2.3 Wavelet series

A wavelet series of a displayed graph can be calculated by performing the following steps.

1. Choose *Series/Wavelet* from the menu.

2. In the text box labeled *Graph,* enter the number of the graph that you want to compute a series for.

3. In the text box labeled *Levels,* enter the number of levels to use for the series. By resting the mouse pointer on this text box, a message should appear which tells you the allowed number of levels.

4. Select the type of wavelet you want to use. The choices are Coif, Daub, Haar, and a selection of biorthogonal wavelets. If you select Coif or Daub, then you must also enter the appropriate number at the left of the wavelet type. By resting the mouse pointer on the text box next to the wavelet type, a message should appear which tells you the allowed numbers.

5. Specify the series type. There are four choices:

   - Threshold
   - Highest mag. coefficients
   - Energy fraction
   - Ascending terms

   We will discuss these choices in Section 2.3.1.

6. Click on the button labeled *Plot.* This will produce the wavelet series.

2.3.1 Series type

There are 4 types of series: (1) Threshold, (2) Highest mag. coefficients, (3) Energy fraction, (4) Ascending terms. We now discuss each of these types.

1. *Threshold.* The threshold method is a commonly used method of handling wavelet coefficients. This is for several reasons. First, there is the fact that for highly correlated data, most of the wavelet coefficients will be near zero. Consequently, a high compression ratio can be obtained by choosing a proper threshold. Second, the method is very useful for removing various types of noise, especially white noise, which is the type of noise that gives the familiar static sound in audio transmissions.

   To specify a threshold make sure that this option is selected by clicking on the label *Threshold.* Then to modify the threshold(s) click on the button labeled *Edit settings.*

   After selecting *Edit settings,* there are two methods for defining a threshold: *Uniform* and *Multiple.* Here are descriptions of each of these two methods.

   a. *Uniform Threshold.* This method involves specifying a threshold by typing in a value in the text box labeled *Value.* You should enter an expression of the form `1/2^n` where `n` is a positive integer. For example, `1/2^2` or `1/2^3`, etc. If the checkbox labeled *Quantize* is checked, then a threshold of the form `1/2^n` will produce an `n+1` bit or `n` bit quantization of all wavelet coefficients (transform values) depending on whether the checkbox labeled *Sign bit* is checked. Note: The tool tip[2] for the text box will provide information on how many bits you are specifying for the threshold value that you have entered.

      If the checkbox labeled *Quantization* is not checked, then a value of the form `1/2^n` produces a threshold of the form `M/2^n`. Here `M` is the maximum magnitude of all wavelet coefficients. All wavelet coefficients whose magnitudes are less than this threshold are set equal to zero, but no quantization is performed on the values that survive this thresholding.

   b. *Multiple Thresholds.* This method involves specifying two thresholds by entering in the text box labeled *Value* an expression of the form `1/2^m:1/2^n` where `m` and `n` are positive integers. With this option, quantization is always performed. The first threshold should have the form `1/2^m`, which produces an `m+1` bit or `m` bit quantization of all trend values, depending on whether the checkbox labeled *Sign bit* is checked. Likewise, the second threshold should have the form `1/2^n`, which produces an `n+1` bit or `n` bit quantization of all fluctuation values, depending on whether the checkbox labeled *Sign bit* is checked. Note: The tool tip for the text box will provide information on how many bits you are specifying for the threshold values that you have entered.

[2] The tool tip is the message that appears when a mouse pointer is rested on a text box, or other item in a window.

2. *Highest mag. coefficients.* This method is used when it is known in advance how many terms will be needed.

   To select this option, just click on the label *Highest mag. coefficients* and then enter the number of coefficients desired. The series will use only that many terms whose coefficients are the specified number of highest magnitude. For example, if you enter 16, then the 16 highest magnitude coefficients will be used, and all other coefficients will be set to 0.

   By resting the mouse pointer on the text box for the number of coefficients, a message should appear which specifies the range of allowed numbers.

3. *Energy fraction.* This method involves specifying a target fraction of the total energy of the signal. A series is then formed from the smallest number of highest magnitude coefficients whose energy (sum of their squares) is at least as large as this target fraction. For example, if you enter 0.99 as the fraction, then the series will use the smallest number of coefficients needed to capture 99% of the energy of the signal.

   To select this option, just click on the label *Energy fraction* and then enter the desired energy fraction.

   By resting the mouse pointer on the text box for the energy fraction, a message should appear which specifies the range of allowed fractions.

4. *Ascending terms.* This method is most useful for generating averaged signals. The series is formed using the first M values of a wavelet transform as coefficients, where M is the number of coefficients specified in the text box at the right of the label *Ascending terms.*

   For example, suppose that the number of points is $1024 = 2^{10}$, then by specifying `1024/2^4` number of ascending terms you will produce the 4^{th} averaged signal. Or, by specifying `1024/2^3` number of ascending terms you will produce the 3^{rd} averaged signal.

   To select this option, just click on the label *Ascending terms* and then enter the number of terms desired.

   By resting the mouse pointer on the text box for the number of coefficients, a message should appear which specifies the range of allowed numbers.

2.4 Wavelet packet transform

To compute a wavelet packet transform of a signal you select *Transform/Wavelet packet.* The rest of the steps are the same as described above for wavelet transforms.

Note: FAWAV only computes the simplest type of wavelet packet transform: the wavelet packet transform where each subsignal is further transformed up to the highest level. More complex transforms are performed by more advanced software such as the ones described in Appendix C of *A Primer on Wavelets and their Scientific Applications*.

2.5 Wavelet packet series

To compute a wavelet packet series for a signal you select *Transform/Wavelet packet.* The rest of the steps are the same as described above for wavelet series.

Note: FAWAV only computes the simplest type of wavelet packet series: where the wavelet packet transforms for each subsignal are further transformed up to the highest level. More complex series are performed by more advanced software such as the ones described in Appendix C of *A Primer on Wavelets and their Scientific Applications*.

2.6 Fourier transforms

There are three types of transforms: the Fourier transform (which utilizes complex exponentials), the Fourier cosine transform, and the Fourier sine transform. We discuss the Fourier transform first (the Fourier cosine transform is discussed in Section 2.6.3 on p. 26, and the Fourier sine transform is discussed in Section 2.6.4 on p. 26).

2.6.1 Fourier transform

A Fourier transform, or power spectrum, of a signal can be calculated by performing the following 9 steps.

1. After plotting your signal data, select *Transforms/Fourier* from the menu.
2. Select one of the two data types, either *Real* or *Complex.*
3. If you selected *Real,* then enter the number of the graph that you want to transform into the text box labeled *Graph.* The values of this signal will constitute the real part of the data, the imaginary part will be assumed to be all zero values. On the other hand, if you selected *Complex,* then enter in the text boxes labeled *Real* and *Imag* the numbers of the graphs whose values you want to use for the real and imaginary parts of the data, respectively.
4. Choose which *Output type* you want. There are three output types to choose from: (a) *Real/Im,* (b) *Amp/Ph,* and (c) *Power sp.* We now describe these three *Output types.*

   (a) *Real/Im* This output type produces the real and imaginary parts of the DFT of the signal.

   (b) *Amp/Ph* This output type produces the amplitude and phase of the DFT of the signal. The amplitude values are the magnitudes of the complex values of the DFT, while the phase values are the angle values (in radians) of these complex numbers in the plane.

   (c) *Power sp* This output type produces a discrete version of the power spectrum of a signal. The values of the power spectrum of a signal are the squares of the magnitudes of the complex values of the Fourier transform of the signal. For a discrete signal the Fourier transform consists of the DFT of the discrete signal.

   Power spectra are used for describing the (global) energy of each frequency in a signal. At least insofar as a DFT can accurately capture the contribution of that frequency to the signal.

5. Choose which *exponent type* to use. The two exponent types are $-$ and $+$. If you select $-$ then (discrete sampled versions of) the exponentials $e^{-i2\pi ux}$ are used for computing transforms. On the other hand, if you select $+$ then (discrete sampled versions of) the exponentials $e^{+i2\pi ux}$ are used for computing transforms.

   These two types are inverse to each other: by Fourier transforming with one exponent type, and then Fourier transforming the result with the other exponent type, you return to the original data.

6. Choose the *Interval type.* There are the following four interval types:

   (i) *Interval type* $[-L, L] \to [-A, A]$ This interval type is used when the signal is defined over an interval of the type $[-L, L]$ and it is desired to produce a Fourier transform over an interval of type $[-A, A]$.

   When an inverse transform is being performed (by choosing the opposite exponent sign), then this same interval type should also be selected for performing the inverse transform.

   (ii) *Interval type* $[0, L] \to [0, A]$ This interval type is used when the signal is defined over an interval of the type [0,L] and it is desired to produce a Fourier transform over an interval of type [0,A].

   When an inverse transform is being performed (by choosing the opposite exponent sign), then this same interval type should also be selected for performing the inverse transform.

   (iii) *Interval type* $[0, L] \to [-A, A]$ This interval type is used when the signal is defined over an interval of the type $[0, L]$ and it is desired to produce a Fourier transform over an interval of type $[-A, A]$.

   When an inverse transform is being performed (by choosing the opposite exponent sign), then the interval type $[-L, L] \to [0, A]$ should be selected for performing the inverse transform.

   (iv) *Interval type* $[-L, L] \to [0, A]$ This interval type is used when the signal is defined over an interval of the type $[-L, L]$ and it is desired to produce a Fourier transform over an interval of type $[0, A]$.

   When an inverse transform is being performed (by choosing the opposite exponent sign), then the interval type $[0, L] \to [-A, A]$ should be selected for performing the inverse transform.

7. Select a *Filter type.* The four choices for *Filter type* are the following:

(i) *Blackman filter* This filter multiplies the signal values $f[k]$ by the factors

$$0.42 + 0.5\cos(2k\pi/N) + 0.08\cos(4k\pi/N)$$

for $k = -N/2, \ldots, 0, \ldots, N/2$ when the domain of the function is of the form $[-L, L]$ and by the factors

$$0.42 + 0.5\cos[2(k - N/2)\pi/N] + 0.08\cos[4(k - N/2)\pi/N]$$

for $k = 0, \ldots, N$ when the domain of the function is of the form $[0, L]$. The symbol N denotes the number of points.

(ii) *Gauss filter* This filter multiplies the signal values $f[k]$ by the factors

$$e^{-4.5(2k/N)^2}$$

for $k = -N/2, \ldots, 0, \ldots, N/2$ when the domain of the function is of the form $[-L, L]$ and by the factors

$$e^{-4.5[2(k-N/2)/N]^2}$$

for $k = 0, \ldots, N$ when the domain of the function is of the form $[0, L]$. The symbol N denotes the number of points.

(iii) *Hamming filter* This filter multiplies the signal values $f[k]$ by the factors

$$0.54 + 0.46\cos(2k\pi/N)$$

for $k = -N/2, \ldots, 0, \ldots, N/2$ when the domain of the function is of the form $[-L, L]$ and by the factors

$$0.54 + 0.46\cos[2(k - N/2)\pi/N]$$

for $k = 0, \ldots, N$ when the domain of the function is of the form $[0, L]$. The symbol N denotes the number of points.

(iv) *Hanning filter* This filter multiplies the signal values $f[k]$ by the factors

$$0.5 + 0.5\cos(2k\pi/N)$$

for $k = -N/2, \ldots, 0, \ldots, N/2$ when the domain of the function is of the form $[-L, L]$ and by the factors

$$0.5 + 0.5\cos[2(k - N/2)\pi/N]$$

for $k = 0, \ldots, N$ when the domain of the function is of the form $[0, L]$. The symbol N denotes the number of points.

There is also a default choice of *None* which performs no filtering of the Fourier transform.

8. Either check, or uncheck, the checkbox labeled *Periodic, endpoint averaged.* If this checkbox is checked, then the first and last values of the signal are averaged to produce a trapezoidal rule approximation of a Fourier transform, thus producing a better approximation to the Fourier transform of an analog signal. If, however, you only want to compute the DFT of a discrete signal, then this checkbox should not be checked.

9. Click the button labeled *Plot*, this will produce the Fourier transform or power spectrum. A separate window will be created for displaying this new data.

2.6.2 Examples

Examples illustrating many of these aspects of computing Fourier transforms[3] can be found in Chapters 5 and 6 of the book

James S. Walker, *Fast Fourier Transforms, 2nd Edition*, CRC Press, 1996

which we shall refer to as *Fast Fourier Transforms, 2nd Edition* from now on.

[3] Using an older command window (DOS) version of FAWAV called FAS.

2.6.3 Fourier cosine transform

A Fourier cosine transform can be calculated by performing the following 4 steps.

1. After plotting your signal, select *Transforms/cosine* from the menu.
2. In the text box labeled *Graph.* enter the number of the graph that you want to transform.
3. Select a *Filter type.* The four choices of *Filter type* are the following:

   (i) *Blackman filter* This filter multiplies the signal values $f[k]$ by
   $$0.42 + 0.5\cos[2(k - N/2)\pi/N] + 0.08\cos[4(k - N/2)\pi/N]$$
   for $k = 0, \ldots, N$. The symbol N denotes the number of points.

   (ii) *Gauss filter* This filter multiplies the signal values $f[k]$ by the factors
   $$e^{-4.5[2(k-N/2)/N]^2}$$
   for $k = 0, \ldots, N$. The symbol N denotes the number of points.

   (iii) *Hamming filter* This filter multiplies the signal values $f[k]$ by the factors
   $$0.54 + 0.46\cos[2(k - N/2)\pi/N]$$
   for $k = 0, \ldots, N$. The symbol N denotes the number of points.

   (iv) *Hanning filter* This filter multiplies the signal values $f[k]$ by the factors
   $$0.5 + 0.5\cos[2(k - N/2)\pi/N]$$
   for $k = 0, \ldots, N$. The symbol N denotes the number of points.

   There is also a default choice of *None* which performs no filtering of the Fourier cosine transform.
4. Click on the button labeled *Plot.* This will produce the Fourier cosine transform.

2.6.4 Fourier sine transform

A Fourier sine transform can be calculated by performing the following 4 steps.

1. After plotting your signal, select *Transforms/sine* from the menu.
2. In the text box labeled *Graph,* enter the number of the graph that you want to transform.
3. Select a *Filter type.* The four choices of *Filter type* are the following:

   (i) *Blackman filter* This filter multiplies the signal values $f[k]$ by
   $$0.42 + 0.5\cos[2(k - N/2)\pi/N] + 0.08\cos[4(k - N/2)\pi/N]$$
   for $k = 0, \ldots, N$. The symbol N denotes the number of points.

   (ii) *Gauss filter* This filter multiplies the signal values $f[k]$ by the factors
   $$e^{-4.5[2(k-N/2)/N]^2}$$
   for $k = 0, \ldots, N$. The symbol N denotes the number of points.

(iii) *Hamming filter* This filter multiplies the signal values $f[k]$ by the factors

$$0.54 + 0.46\cos[2(k - N/2)\pi/N]$$

for $k = 0, \dots, N$. The symbol N denotes the number of points.

(iv) *Hanning filter* This filter multiplies the signal values $f[k]$ by the factors

$$0.5 + 0.5\cos[2(k - N/2)\pi/N]$$

for $k = 0, \dots, N$. The symbol N denotes the number of points.

There is also a default choice of *None* which performs no filtering of the Fourier sine transform.

4. Click on the button labeled *Plot.* This will produce the Fourier sine transform.

2.7 Fourier series, sine series & cosine series

FAWAV allows you to calculate partial sums of Fourier series and Fourier cosine and Fourier sine series. A partial sum of a Fourier series (period Ω) has the form

$$c_0 + \sum_{k=1}^{M} \left(A_k \cos\frac{2k\pi x}{\Omega} + B_k \sin\frac{2k\pi x}{\Omega} \right)$$

where c_0, A_k, and B_k are the Fourier coefficients. We say that M is *the number of harmonics* used in the partial sum.

Similarly, a Fourier cosine series has the form

$$c_0 + \sum_{k=1}^{M} A_k \cos\frac{k\pi x}{\Omega}$$

and a Fourier sine series has the form

$$\sum_{k=1}^{M} B_k \cos\frac{k\pi x}{\Omega}$$

where M is the number of harmonics. (Further discussion of the mathematical details can be found in *Fast Fourier Transforms, 2nd Edition,* Chapter 1.)

These partial sums can all be calculated using FAWAV by the following 6 steps.

1. Graph the signal, and then select either *Series/Fourier,* or *Series/cosine,* or *Series/sine* from the menu.

2. In the text box labeled *Graph,* enter the number of the graph for which you want to compute a Fourier series partial sum.

3. In the text box labeled *Harmonics,* enter the number of harmonics that you want to use in the partial sum.

4. Select the *Filter type.* There are two types, *Unfiltered* and *Filtered.* If you select *Unfiltered,* then the Fourier coefficients are used without any modification when forming the partial sum. If you select *Filtered,* then a list box is activated and you can select a filter. Most of these filters are functions that modify the Fourier coefficients by multiplying them by different functions. These filters are the following sixteen linear filters:

Blackman	Cesaro	Cos	Diffraction
DlVP	FresCos	FresSin	Hamming
Hanning	Heat	Gauss	Kaiser
Poisson	Riesz	Sinc	User

There are also three other non-linear filters, which modify the Fourier coefficients in a non-linear way. They are the following:

Energy percentage Highest magnitude Threshold

All of these 19 filters are described below in Section 2.7.1.

5. Check or uncheck the checkbox labeled *Periodic, endpoint averaged.* If you check it, then the last signal value is set equal to the first signal value when computing DFTs (i.e. a trapezoidal rule is used for approximating Fourier series coefficients). This is done when the discrete Fourier series is computed as an approximation to a Fourier series for an analog signal.
6. Click on the button labeled *Plot.* This will produce the Fourier series partial sum.

2.7.1 Filtering of Fourier series

In this section we describe each of the 19 filtering methods for Fourier series.

1. *Blackman* The Blackman filter modifies a Fourier series partial sum by multiplying the k^{th} and $-k^{\text{th}}$ coefficients by the following convergence factor

   $$0.42 + 0.5\cos(k\pi/N) + 0.08\cos(2k\pi/N)$$

   where N is the number of harmonics in the partial sum.

   For Fourier sine and cosine series, this convergence factor multiplies the k^{th} coefficient.

2. *Cesaro* The Cesaro filter modifies a Fourier series partial sum by multiplying the k^{th} and $-k^{\text{th}}$ coefficients by the following convergence factor

   $$1 - |k|/N$$

   where N is the number of harmonics in the partial sum.

   For Fourier sine and cosine series, this convergence factor multiplies the k^{th} coefficient.

3. *Cos* The Cosine filter modifies a Fourier series partial sum by multiplying the k^{th} and $-k^{\text{th}}$ coefficients by the following factor

   $$\cos(W\,T\,k\pi/L)$$

   where L is the right endpoint of the chosen interval, W and T are the *wave constant* and *time constant* (which you are required to input). These constants must be non-negative.

   For Fourier sine and cosine series, this convergence factor multiplies the k^{th} coefficient.

4. *Diffraction* The Diffraction filter modifies a Fourier series in two ways: First, by multiplying the k^{th} and $-k^{\text{th}}$ coefficients by the factor $\cos[W\,T\,(k\pi/L)^2)]$. Second, by multiplying by the factor $\sin[W\,T\,(k\pi/L)^2)]$. The result is then created by summing the squares of the values of these two modified Fourier series. You are required to supply the values of the *wave constant*, W, and *time constant*, T. The result is the same as the sum of the squares of the graphs produced by the *Frescos* and *FresSin* filters.

   For Fourier sine and cosine series, this convergence factor multiplies the k^{th} coefficient.

   This filter can be used to create a diffraction pattern (for electron diffraction use the wave constant 0.578).

5. *DlVP* The de la Vallee Poussin filter modifies a Fourier series partial sum by multiplying the k^{th} and $-k^{\text{th}}$ coefficients by the following convergence factor

   $$\begin{cases} 1 & \text{if } 2|k| \le N \\ \\ 2(1-|k|/N) & \text{if } N < 2|k| \le 2N \end{cases}$$

where N is the number of harmonics in the partial sum.

For Fourier sine and cosine series, this convergence factor multiplies the $k^{\rm th}$ coefficient.

6. *FresCos* The Fresnel cosine filter modifies a Fourier series partial sum by multiplying the $k^{\rm th}$ and $-k^{\rm th}$ coefficients by the following factor
$$\cos[W\,T\,(k\pi/L)^2]$$
where L is the right endpoint of the chosen interval, W and T are the *wave constant* and *time constant* (which you are required to input). These constants must be non-negative.

   For Fourier sine and cosine series, this convergence factor multiplies the $k^{\rm th}$ coefficient.

7. *FresSin* The Fresnel sine filter modifies a Fourier series partial sum by multiplying the $k^{\rm th}$ and $-k^{\rm th}$ coefficients by the following factor
$$\sin[W\,T\,(k\pi/L)^2]$$
where L is the right endpoint of the chosen interval, W and T are the *wave constant* and *time constant* (which you are required to input). These constants must be non-negative.

   For Fourier sine and cosine series, this convergence factor multiplies the $k^{\rm th}$ coefficient.

8. *Hamming* The Hamming filter modifies a Fourier series partial sum by multiplying the $k^{\rm th}$ and $-k^{\rm th}$ coefficients by the following convergence factor
$$0.54 + 0.46\cos(k\pi/N)$$
where N is the number of harmonics in the partial sum.

   For Fourier sine and cosine series, this convergence factor multiplies the $k^{\rm th}$ coefficient.

9. *Hanning* The Hanning filter modifies a Fourier series partial sum by multiplying the $k^{\rm th}$ and $-k^{\rm th}$ coefficients by the following convergence factor
$$0.5 + 0.5\cos(k\pi/N)$$
where N is the number of harmonics in the partial sum.

   For Fourier sine and cosine series, this convergence factor multiplies the $k^{\rm th}$ coefficient.

10. *Heat* The Heat filter modifies a Fourier series partial sum by multiplying the $k^{\rm th}$ and $-k^{\rm th}$ coefficients by the following factor
$$e^{-W\,T\,(k\pi/L)^2}$$
where L is the right endpoint of the chosen interval, W and T are the *diffusion constant* and *time constant*, which you are required to input. These constants must be non-negative.

    For Fourier sine and cosine series, this convergence factor multiplies the $k^{\rm th}$ coefficient.

11. *Gauss* The Gauss filter modifies a Fourier series partial sum by multiplying the $k^{\rm th}$ and $-k^{\rm th}$ coefficients by the following convergence factor
$$e^{-4.5(k/N)^2}$$

    For Fourier sine and cosine series, this convergence factor multiplies the $k^{\rm th}$ coefficient.

12. *Kaiser* The Kaiser filter modifies a Fourier series partial sum by multiplying the $k^{\rm th}$ and $-k^{\rm th}$ coefficients by the following convergence factor
$$\frac{I[\,T\pi\sqrt{1-(k\pi/N)^2}]}{I(T\pi)}$$
where I is the zero order modified Bessel function, and N is the number of harmonics in the partial sum, and T is a *damping constant* (which you are required to input). The damping constant must be non-negative.

    For Fourier sine and cosine series, this convergence factor multiplies the $k^{\rm th}$ coefficient.

13. *Poisson* The Poisson filter modifies a Fourier series partial sum by multiplying the k^{th} and $-k^{\text{th}}$ coefficients by the following convergence factor
$$r^{|k|}$$
The number r is called the *damping ratio* (which you are required to input). It must be between 0 and 1.

    For Fourier sine and cosine series, this convergence factor multiplies the k^{th} coefficient.

14. *Riesz* The Riesz filter modifies a Fourier series partial sum by multiplying the k^{th} and $-k^{\text{th}}$ coefficients by the following convergence factor
$$(1-(k/N)^2)^p$$
where N is the number of harmonics in the partial sum, and p is called the *damping power* (which you are required to input). The damping power must be non-negative.

    For Fourier sine and cosine series, this convergence factor multiplies the k^{th} coefficient.

15. *Sinc* The Sinc filter modifies a Fourier series partial sum by multiplying the k^{th} and $-k^{\text{th}}$ coefficients by the following factor
$$T\,\mathrm{sinc}(W\,T\,k/L)$$
where L is the right endpoint of the chosen interval, W and T are the *wave constant* and *time constant* (which you are required to input). These constants must be non-negative.

    For Fourier sine and cosine series, this convergence factor multiplies the k^{th} coefficient.

16. *User* The User filter modifies a Fourier series partial sum by multiplying the k^{th} and $-k^{\text{th}}$ coefficients by a factor
$$f(\,|k|/N\,)$$
The function $f(x)$ is created by you, and is assumed to be defined over the interval $[0,1]$. The number N is the number of harmonics in the partial sum.

    For Fourier sine and cosine series, this convergence factor multiplies the k^{th} coefficient.

    For example, the *Cesaro* filter could be implemented as a *User* filter if you entered `x` as the function. Or, the *Hanning* filter could be implemented as a *User* filter if you entered `0.5 + 0.5 cos(pi x)` as the function.

17. *Energy percentage* This filter selects the least number of the highest magnitude Fourier coefficients whose total energy is equal (or larger) than the percentage of the energy of all the coefficients. You will be required to specify this percentage which is a number between 0 and 1.

    For example, if you specify a percentage of 0.99, then the coefficients that are used will have a total energy that is at least 99% of the energy of all the coefficients. These will be the least number of coefficients needed because they are of the highest magnitude.

18. *Highest magnitude* This filter selects a specified number of the highest magnitude coefficients for creating the partial sum. You specify this number in the text box labeled *Harmonics.*

    For example, if you specify 10 harmonics, then the largest 20 or 21 coefficients in magnitude will be used for forming the partial sum. Note: Because Fourier series are formed from sums of complex exponentials, the number of coefficients is twice as many as the number of harmonics (plus one more coefficient, if the constant coefficient is included). [For Fourier cosine series it will be the largest 10 or 11 coefficients, depending on whether the constant term is included; while for Fourier sine series it will be the largest 10 coefficients.]

19. *Threshold* This filter selects those coefficients whose magnitudes are above a threshold that you specify, all other coefficients are set equal to zero. You must specify a non-negative number for the threshold.

2.7.2 Examples

Examples illustrating many of these aspects of computing Fourier series[4] can be found in Chapter 4 of *Fast Fourier Transforms, 2nd Edition*.

2.8 Audio

FAWAV allows one or more graphs to be played as audio files by performing the following 7 steps.

1. Graph the signal(s) that you want to play. Typically this is done by loading an audio file (see Section 2.1.11 on p. 19 for discussion of how to load an audio file)—although FAWAV allows any graph to be played.

2. Select *Graph/Audio* from the menu.

3. In the text box labeled *Graph(s),* enter either one number, or a sequence of numbers separated by commas, that specify which graphs you want to play. For example, suppose there are three graphs displayed. Some possible numbers that you could enter are

   `1`, or `3`, or `1,2`, or `3,2,1,1,2,3`

   which, respectively, would tell FAWAV to play the first graph displayed, or the third graph displayed, or the third graph followed by the second graph followed by the first graph, and then repeating in reverse order.

4. Choose a *bit-rate* of either 8 bits or 16 bits. If you select 8 bits, then volume levels are from -128 to 127 (that is, 2^8 in all). Or, if you select 16 bits, then volume levels are from -32768 to 32767 (that is, 2^{16} in all).

5. Specify a *Sampling rate.* The allowed values are integers from 441 to 44100. The sampling rate specifies how many volume levels are played per second by the computer audio system.

6. Specify a *Volume level.* The volume level is the factor by which the resulting sound is multiplied. For example, if you specify a volume level of 2, then the sound will play two times louder than if you specify a volume level of 1. Warning: Be careful with volume levels if you change from 16 bits to 8 bits. A graph that plays moderately at 16 bits will play exceedingly loudly (most values at highest magnitude volume levels) when played at 8 bits. Conversely, a graph that plays loudly at 8 bits will be hardly audible, if not completely inaudible, at 16 bits.

7. Click on the button labeled *Play.*

2.9 Convolutions and Correlations

FAWAV performs three types of discrete convolution operations. These are

1. convolution

2. pair-correlation

3. auto-correlation

We now describe each of these 3 types.

[4] Using an older command window (DOS) version of FAWAV called FAS.

2.9.1 Convolution

To perform a convolution[5] of two displayed graphs, perform the following 3 steps.

1. Enter the numbers for the displayed graph in the text boxes labeled *Graph 1* and *Graph 2.*

2. If you want to divide the values of the convolution by the length of the interval over which the displayed graphs are plotted, then click on the checkbox labeled *Divide by interval length.* Warning: If the two signals are defined over different intervals, then FAWAV will divide by the length of the interval for the second function. This is confusing, and is avoided by restricting to convolutions of functions defined over a common interval when selecting *Divide by interval length.*

3. Click on the button labeled *Plot.* This will produce the convolution.

2.9.2 Pair-correlation

To perform a pair-correlation (also known simply as correlation) of two displayed graphs, perform the following 4 steps.

1. Enter the numbers for the displayed graph in the text boxes labeled *Graph 1* and *Graph 2.*

2. If you want to divide the values of the correlation by the length of the interval over which the displayed graphs are plotted, then click on the checkbox labeled *Divide by interval length.* Warning: If the two signals are defined over different intervals, then FAWAV will divide by the length of the interval for the second function. This is confusing, and is avoided by restricting to correlations of functions defined over a common interval when selecting *Divide by interval length.*

3. If you want to divide the values of the correlation by the energy of the *Graph 1* signal, then click on the checkbox labeled *Normalize.*

4. Click on the button labeled *Plot.*

2.9.3 Auto-correlation

To perform an autocorrelation on a displayed graph (which is the pair-correlation of a graph with itself), perform the following 4 steps.

1. Enter the number for the displayed graph in the text box labeled *Graph.*

2. If you want to divide the values of the autocorrelation by the length of the interval over which the displayed graph is plotted, then click on the checkbox labeled *Divide by interval length.*

3. If you want to divide the values of the autocorrelation by the energy of the displayed signal, then click on the checkbox labeled *Normalize.*

4. Click on the button labeled *Plot.* This will produce the autocorrelation.

2.9.4 Examples

Several examples of the use of convolution and correlation are given in the two references, *A Primer on Wavelets and their Scientific Applications* and *Fast Fourier Transforms, 2nd Edition*.

[5] To be precise, cyclic convolution. For more of the mathematical theory, see the treatment of convolution in *Fast Fourier Transforms, 2nd Edition*.

2.10 Spectrograms and Scalograms (see Chapters 3 and 4)

On the *Analysis* menu there are two choices for time-frequency analysis of 1D signals: *Spectrogram* and *Scalogram*. These two choices play such important roles in the field of time-frequency analysis that they deserve their own separate chapters. We discuss *Spectrograms* in Chapter 3 beginning on p. 45, and we discuss *Scalograms* in Chapter 4 beginning on p. 53.

2.11 Menu Items for 1D-Form

There are 9 menu items on a 1D-form:

File Edit View Graph Series Transforms Convolve Analysis Help

We now describe the contents of each of these 9 items.

2.11.1 File

The *File* menu contains 13 items, which we now examine.

1. *New 1Dim* Selecting this item will open a new 1D-form. Each 1D-form is numbered when it is opened. You can access a list of all opened windows by selecting *View* from the menu.

2. *New 2Dim* Selecting this item will open a new 2D-form (see Chapter 5 beginning on p. 59). Each 2D-form is numbered when it is opened. You can access a list of all opened windows by selecting *View* from the menu.

3. *New Audio Editor* Selecting this item will open a new Audio Editor window (see Chapter 6 beginning on p. 85). Each Audio Editor is numbered when it is opened. You can access a list of all opened windows by selecting *View* from the menu.

4. *New Image Processor* Selecting this item will open a new Image Processor window (see Chapter 7 beginning on p. 89). Each Image Processor is numbered when it is opened. You can access a list of all opened windows by selecting *View* from the menu.

5. *Open form* You select this item in order to open a previously saved form (either a 1D-form, a 2D-form, a Spectrogram form, or a Scalogram form). A standard WINDOWS File Open dialog box opens up, from which you can select a saved form to load. The file extension `.fw1` is used for 1D forms, the file extension `.fw2` is used for 2D forms, the file extension `.spg` is used for spectrograms, and the file extension `.scg` is used for scalograms.

6. *Open Sound File (*.wav)* This item can be used for loading a `.wav` audio file into the 1D-form. It is recommended, however, that you use the method described in Section 2.1.11 on p. 19 instead. Using that method will help you to become familiar with the Popup menu technique for 1D-forms (see Section 2.12 beginning on p. 41 for a description of this Popup Menu).

7. *Open Data File* This item can be used for loading an `.fb1` data file into the 1D-form. It is recommended, however, that you use the method described in Section 2.1.12 on p. 19 instead. Using that method will help you to become familiar with the Popup menu technique for 1D-forms (see Section 2.12 beginning on p. 41 for a description of this Popup Menu).

8. *Save Form As* Select this item in order to save all of your work in the 1D form to a file with extension `.fw1`. The saved form can be reloaded later by selecting *Open form* from the *File* menu.

9. *Save Form* When you have loaded a previously saved form and then made modifications to that form, then this menu item becomes available for resaving the modified form (using the same file name).

10. *Save Graph* This choice allows you to save one of the graphs in the 1D-form to a file, with extension `.fb1`. Although this method can be used, it is recommended that you right-click on the graph area (near the graph you want to save) and then select *Save graph/Graph* from the Popup menu that appears.

11. *Save Sound File (*.wav)* This choice allows you to save one of the graphs in the 1D-form to a sound file, with extension `.wav`. Although this method can be used, it is recommended that you right-click on the graph area (near the graph you want to save) and then select *Save graph/Sound file* from the Popup menu that appears.

12. *Print* This section can be used to either print an individual graph or to print the entire 1D form (as it is displayed on screen). For further discussion, please go to Section 2.12.9 on p. 41.

13. *Close* Selecting this choice will close the 1D form. *Warning:* FAWAV does not prompt you to save the 1D form if you have made modifications. If you want to save your work, be sure and save the form (using *File/Save Form*) before you close it.

2.11.2 Edit

The *Edit* menu contains 6 items, which we now examine.

1. *Points used* Choose this item in order to change the number of points that FAWAV is using for plotting graphs. The choices are

   $$64 \quad 128 \quad 256 \quad 512 \quad 1024 \quad 2048 \quad 4096 \quad 8192 \quad 16384 \quad 32768 \quad 65536$$

   with the presently used number of points preceded by a checkmark (✓). *Warning:* Choosing a new number of points will erase all graphs from the screen, and from memory, causing you to lose that work (unless you have previously saved it).

2. *Remove graph* If you have one or more graphs displayed, then you can remove one of them (from the screen and from memory) by selecting this item. If there is just one graph displayed, then it will automatically be removed. If there is more than one graph displayed, then a dialog box opens on the right side of the form in which you enter the number of the graph to remove. *Warning:* you are not prompted with a question about whether or not you really want to remove a graph, the graph is simply removed immediately.

3. *Erase screen* Choosing this item will remove all graphs from the screen, and from memory. *Warning:* All graphs will be erased immediately, causing you to lose that work (unless you have previously saved it).

4. *Remarks* When you select this choice, then a text box opens on the right of the form, where you can enter remarks describing the work that you have performed in this form. This is especially useful for keeping a record of your work if you intend to save the 1D-form.

5. *Caption* Choosing this item allows you to edit the caption that appears below the graph area. This caption can also be edited by clicking on it.

6. *Screen capture* This choice will create an image of the entire FAWAV window, as it appears on screen, which you can then paste into word processing documents like WORDPAD documents, or other documents in other programs that accept WINDOWS clipboard input (via a *Paste* command).

2.11.3 View

The menu item *View* contains 6 items, which we now examine.

1. *Display style* This procedure allows you to change the way in which graphs are displayed. The different modes for graphing are organized into four categories: *Grid style, Plot style, Background color,* and *X-Y intervals.*

(a) *Grid style* If you select *Graph paper,* then the background of the graphs will consist of seven dotted horizontal and seven dotted vertical lines (resembling graph paper). If you select *Axes,* then a horizontal axis and vertical axis will be drawn on the screen. If you select *Blank,* then the background of the graphs will be empty.

(b) *Plot style* If you select *Lines,* then the values of graphs will be connected by line segments. On the other hand, if you select *Dots,* then the values of the graphs will be plotted as isolated values.

(c) *Background color* If you select *White,* then the background of the graphs will be white, while if you select *Black,* then the background will be black.

(d) *X-Y intervals* Enter the endpoints (separated by a comma) of the *X*- and *Y*-intervals that you want displayed. For instance, if you enter $-3, 3$ for the *X*-interval, then the *X*-interval for the graphs will be $[-3, 3]$. Note: The graphs will not be recomputed over this interval.

2. *Window* This choice causes the original X- and *Y*-intervals that were first used for graphing to be used for regraphing. (You select this choice after those intervals have been changed in some way and you wish to restore the original intervals.)

3. *Redraw* This choice will redraw the graphs on the screen.

4. *Arrange windows* This choice will perform automatic rearrangement of the positions of multiple windows within the main FAWAV window.

The *View* menu always has at its end a listing of all the windows that are open within the FAWAV workspace, so that you can select which window to be the active one. [The presently active one is marked with a checkmark (✓).]

2.11.4 Graph

The *Graph* menu contains 10 items, which we now describe.

1. *Plot* This item opens up the plotting dialog box on the right of the window. See Section 2.1, beginning on p. 12, for further discussion. There is also online help available by selecting *Help for Graphing* from the *Help* menu when the plotting dialog box is displayed.

2. *Audio* This item opens up the Audio dialog box on the right of the window. See Section 2.8, beginning on p. 31, for further discussion. There is also online help available by selecting *Help for Audio* from the *Help* menu when the Audio dialog box is displayed.

3. *Quantize (8-bit)* This procedure rounds graph values to integers between -128 and 127. It can be used for producing 8-bit sound data.

4. *Quantize (16-bit)* This procedure rounds graph values to integers between -32768 and 32767. It can be used for producing 16-bit sound data.

5. *Denoise (Gabor)* With this choice, you can denoise a signal via Gabor transforms (also known as spectrograms). A Blackman spectrogram (Gabor transform) will be automatically computed and a denoising function will appear as well. If you want to use this default denoising function, then you click on the *Plot* button and then select *Invert* from the spectrogram's menu. For more discussion, see Section 3.4 beginning on p. 47.

6. *Denoise (Wavelet)* This procedure denoises a signal using either VISUSHRINK (when the box labeled *Avg* is not checked), or using CYCLE-SPIN averaging (when the box labeled *Avg* is checked). You select the type of wavelet to use, and the number of levels (4 to 6 works best), and then click the *Plot* button to denoise.

7. *Median denoise* This procedure denoises a signal using a 3-point median filter. It is used when there is "pop noise" (also called "impulsive noise") present.

8. *Adaptive median denoise* This procedure denoises a signal using a 3-point adaptive median filter. It is used when there is “pop noise” (also called “impulsive noise”) present. It usually performs a better denoising than median denoising.

9. *Down sample* This procedure creates a new graph, from the graph that you specify, by using only its even indexed values. A new window appears that contains the graph of this down sampled graph. (Note: this procedure is not available if the graph consists of only 64 points).

10. *Interpolate* This procedure creates a new graph, from the graph that you specify, by inserting a new value at each midpoint between each pair of its values. There are two ways that this can be done.

    (a) *Haar interpolation* Haar interpolation inserts a new value at each midpoint by assigning the immediately preceding function value. Thus producing a function which repeats values two at a time.

    (b) *D-D 4,2 Interpolate* This method of interpolation belongs to a type known as *Deslauriers-Dubuc.* It is described in more detail in *A Wavelet Tour of Signal Processing,* by Stephane Mallat (Academic Press, 1999). With this interpolation, the new function equals the original at the old data points, and has new values at midpoints obtained via an inverse wavelet transform (producing a better approximation to a continuous function than with Haar interpolation).

2.11.5 Series

The series that are available with this menu item fall into two categories: (1) Fourier, Sine, and Cosine; (2) Wavelet, Wavelet Packet. The three choices in the first category are described in detail in Section 2.7, beginning on p. 27. And the two choices in the second category are described in detail in Section 2.3 on p. 21 and Section 2.5 on p. 23.

2.11.6 Transforms

The transforms that are available with this menu fall into two categories: (1) Fourier (see Section 2.6 on p. 23), Sine (see Section 2.6.3 on p. 26), and Cosine (see Section 2.6.4 on p. 26); (2) Wavelet (see Section 2.2 on p. 21) and Wavelet Packet (see Section 2.4 on p. 23).

2.11.7 Convolve

The three choices on the *Convolve* menu are described in detail in Section 2.9 on p. 31.

2.11.8 Analysis

The *Analysis* menu contains 14 items. We describe these items in the 14 subsections that follow.

Statistics

This choice produces a printout on the right of the window of various statistics summarizing the data consisting of a signal’s values. For example, if the graph of `2x + 4` is plotted over the interval $[-0.5, 0.5]$ using 1024 points, and this procedure is invoked, then we get the following printout:

```
Maximum =  5
Minimum =  3
Mean =  4
Variance =  .334310531616211
Std. Dev. =  .578195928398161
Energy =  16742.333984375
Number of samples =  1025
Theoretical Dim =  984.264463088354
```

Most of these values are elementary statistical measures. The `Energy` is the sum of the squares of the values of the signal. The `Number of samples` is 1 more than the number of points, because FAWAV always graphs a right-endpoint in addition to a left-endpoint. The `Theoretical Dim` is a quantity related to the ability to do a wavelet compression of the signal [for more details, see the book, *Wavelet Analysis: From Theory to Software* by M.V. Wickerhauser (AK Peters, 1994)].

Norm difference

This choice allows you to compute various measures of the global difference between functions, typically used to express the amount of error of one graph as approximant to another graph. To compute the norm difference between two displayed graphs, perform the following 4 steps.

1. In the text boxes labeled *Graph1* and *Graph2* enter the numbers for the two graphs whose norm difference you want to compute.
2. Select the *Norm type.* There are two kinds: *Sup norm* and *Power norm.* More details about these choices will be given below.
3. Select whether the norm is to be *Absolute* or *Relative.*
4. Click on the button labeled *Compute.* The norm difference will be computed and the result will be given in the text box below the *Compute* button.

Note: Norm differences are computed over the region shown on screen. This may represent a subinterval of the full intervals over which the two graphs are defined, if you have changed the display interval.

Definitions of various norm differences

We now describe the various norm differences that can be computed. First, assume that *Absolute* norm differences are computed. If a *Sup* norm is chosen, then the maximum absolute value of all differences between the specified signals' values is found, over the interval on which the graph is displayed, and this maximum is the Sup norm difference. If a *Power* norm is selected, then the power to be used must be specified in the text box labeled *Power.* The power, p, must be a number greater than or equal to 1. The power norm difference is computed by forming a Riemann sum approximation to the integral of the power p of the absolute value of the differences between the two graphs, over the interval on which the graph is displayed, and then taking the p^{th} root of the result. Note: If you checked the box labeled *Normalized* then, before the p^{th} root is taken, the approximate integral is divided by the length of the interval on screen.

If *Relative* norm differences are computed, then the absolute norm differences described above are divided by the norm of the graph specified as *Graph1.* That is, the norm difference is relative to the norm of Graph 1. (We discuss norms in the next subsection.)

Norm

A *Norm* of a graph is a numerical measure of its magnitude (we discuss this further below). To compute the norm of a displayed graph, perform the following 4 steps.

1. In the text box labeled *Graph,* enter the number for the graph whose norm you want to compute.
2. Select the *Norm type.* There are two kinds: *Sup* norm and *Power* norm. More details about these choices will be given below.
3. Select whether the norm is to be *Absolute* or *Relative.*
4. Click on the button labeled *Compute.* The norm will be computed and the result will be given in the text box below the *Compute* button.

Note: Norms are computed over the interval shown on screen. This may represent a subinterval of the full interval over which the graph is defined, if you have changed the display interval.

Definitions of various norms

We now describe the various norms that can be computed. First, assume that *Absolute* norms are computed. If a *Sup* norm is chosen, then the maximum absolute value of all the specified graph's values is found over the interval on which the graph is displayed, and this maximum is the *Sup* norm. If a *Power* norm is selected, then the power to be used must be specified in the text box labeled *Power.* The power, p, must be a number greater than or equal to 1. The power norm is computed by forming a Riemann sum approximation to the integral of the power p of the absolute value of the graph, over the interval on which the graph is displayed, and then taking the p^{th} root of the result. Note: If you checked the box labeled *Normalized* then, before the p^{th} root is taken, the approximate integral is divided by the length of the interval on screen.

Trace

This procedure can be used for reading off the values of the displayed graphs. As the function values are displayed on the right, there are *crosshairs* drawn on the graphs on the left to indicate where these function values lie on each graph.

There are a couple ways to change the position of the crosshairs.

1. You can click on the screen at a point on a graph whose coordinates you desire. If you click with the left mouse button, then the crosshairs will move to the position that you have clicked and the values will update on the right side of the screen. If you click with the right mouse button, then you will also have the option of zooming in on the point you have selected.

2. You can also use the button bar on the bottom left of the window, which has buttons labeled with the following symbols:

   `<<        <        >        >>       L       C       R`

   To see how these buttons work it is best to just try them (a succinct description of their properties can also be found by simply resting the mouse on them and waiting for a short message to appear).

Calculator

To perform a calculation with the calculator, proceed as follows.

1. Enter a formula for calculation in the text box at the top. Valid formulas are ones that would be accepted by most graphing calculators. For example, these three formulas

   ```
   cos(pi/8)

   sqr(2)

   g1(a)\a=4sqr(5)
   ```

   are all valid formulas. Notice that in the last formula, we used a command string to substitute the value of `4sqr(5)` in place of `a` in the formula. And notice also that this last formula uses the graph variable `g1` to indicate that the first displayed graph should be used to compute a value for `g1(a)`. (More details on the functions available can be found in Section 2.1 beginning on p. 12.)

2. Click on the button labeled *Calculate* in order to perform the calculation.

Integral

To compute an integral of a displayed graph, perform the following 2 steps.

1. In the text box labeled *Graph,* type in the number of the graph for which you want to compute an integral.

2. Click on the button labeled *Compute.* This will result in the computation of the integral, over the interval on which the graph is displayed. The result, which is a trapezoidal rule approximation of the exact integral, is displayed in the text box below the *Compute* button.

Derivative

To numerically approximate the first derivative of a displayed graph, perform the following steps.

1. In the text box labeled *Graph,* type in the number of the displayed graph that you want to approximate the derivative of.

2. Click on the button labeled *Compute.* The approximate derivative will then be displayed.

Technical note: The approximate derivative is accurate to fourth order. That is, if h is the value of the increment separating successive abscissa values, then the error between this approximate derivative and the true derivative is a multiple of h^4, whenever the formula used to graph the function has at least 5 continuous derivatives.

Scalar product

To compute a scalar product of two displayed graphs, perform the following steps.

1. In the text boxes labeled *Graph1* and *Graph2,* enter the numbers of the graphs for which you want to compute a scalar product.

2. Click on the button labeled *Compute.* This will result in the computation of the scalar product integral over the interval shown on screen. The result, which is a trapezoidal rule approximation of the exact integral, is displayed in the text box below the *Compute* button.

Histogram

To compute a histogram of a displayed graph's values, perform the following steps.

1. In the text box labeled *Graph,* type the number of the graph for which you want to compute a histogram.

2. From the list labeled *Quantization,* select the number of levels (divisions of the range of values) to be used for computing the histogram.

3. If zero levels are to be included—this is known as uniform scalar quantization—then check the box labeled *Include zero values.* If this box is not checked, then the method is known as dead-zone quantization.

   *I*f a sign bit is to be included (i.e., both positive and negative values are quantized), then check the box labeled *Include sign bit.* If this box is not checked, then only absolute values are quantized.

4. To perform the histogram calculation, click on the button labeled *Plot.*

Note: The histogram is performed relative to the largest magnitude of the graph's values (over the interval displayed) and using only displayed values (so if you have zoomed in then just the region where you have displayed values is used for the histogram). The abscissa values of the histogram are the graph maxima divided by 2 raised to the number of bits used. For example, if the histogram is a 6-bit histogram, with sign-bit, then it will be graphed over $[-32, 32]$ so the values ± 32 correspond to $\pm M$ where M is the maximum absolute value for the graph.

Energy graph

This procedure produces a plot of the cumulative energy of the signal over its abscissa values. You need to perform the following 3 steps.

1. In the text box labeled *Graph,* enter the number of the graph for which you want to produce an energy graph.

2. If you want the energy plotted according to percentage of total energy, then you check the box labeled *Percentage.* If this box is not checked, then the energy graph consists of a running total of the squares of the values of the graph that you specified in Step 1. If this box is checked, then this running total is divided in every instance by the sum of all the squares (the total energy), producing a running percentage of total energy.

3. Click on the button labeled *Plot.* This will produce a display of the energy graph.

Sort magnitudes

This procedure rearranges the values of a graph in decreasing order of their magnitudes (ignoring signs). (If there is a tie, then the values are left in their original order.) If there is just one graph displayed, then its values are immediately sorted and the resulting graph is displayed. If there is more than one graph, then you must enter the number for the graph that you want sorted in the text box labeled *Graph* and click on the button labeled *Sort.*

Filtering

Filtering is performed (typically this is done to a Fourier transform of some signal) using the following 3 steps.

1. Select the type of *Data* to be filtered. There are two types: *Real* or *Complex.* If you select *Real* then you must enter the number of the graph for the real data in the text box labeled *Graph.* If you select *Complex* then you must enter the numbers of the graphs for the real and imaginary parts of the complex data in the text boxes labeled *Real* and *Imag.*

2. Select the type of *Filter.* There are two types: *Real* or *Complex.* If you select *Real* then you must enter the number of the graph for the real filter in the text box labeled *Graph.* If you select *Complex* then you must type in the numbers of the graphs for the real and imaginary parts of the complex filter in the text boxes labeled *Real* and *Imag.*

3. To perform the filtering click on the button labeled *Plot.* This will multiply the data function values by the filter function values to produce the filtered function values.

Scalogram

This item allows you to perform a fundamental type of time-frequency analysis of a signal known as a *scalogram.* It is described in Chapter 4 beginning on p. 53.

Spectrogram

This item allows you to perform a fundamental type of time-frequency analysis of a signal known as a *spectrogram.* It is described in Chapter 3 beginning on p. 45.

2.11.9 Help

This item typically has two choices. There is a general choice, *Help for FAWAV (1D),* which you choose to get the general help for all of the features of the 1D-form. When a dialog box is open on the right side of the screen, there will be a second choice specifically designed to access the desired topic in the help file (a form of interactive help).

2.12 The Popup menu for 1D Forms

If you right-click on the graph area on the left of a 1D-form, then a popup menu will appear with 14 items. We now discuss each of these items.

2.12.1 Zoom

This choice will zoom-in to a region centered on the coordinates of the tip of the mouse pointer (at the position where you right-clicked on the graph area).

2.12.2 Expand

This choice will zoom-out to a region centered on the coordinates of the tip of the mouse pointer (at the position where you right-clicked on the graph area).

2.12.3 Window

After zooming-in, this choice can be used to restore the view to the original graph region.

2.12.4 Sort magnitudes

See Section 2.11.8 on p. 40.

2.12.5 Energy graph

See Section 2.11.8 on p. 40.

2.12.6 Histogram

See Section 2.11.8 on p. 39.

2.12.7 Load

This choice can be used for loading a signal. There are two choices: (1) *Graph* (see Section 2.1.12 on p. 19); (2) *Sound* (see Section 2.1.11 on p. 19).

2.12.8 Save graph

This choice can be used for saving a signal in one of two formats: (1) You select *Graph* to save a signal in binary `.fb1` format (for loading using the *Load/Graph* method); (2) You select *Sound file* to save a signal in `.wav` format (for loading using the *Load/Sound* method). **Note:** when saving a signal as a sound signal, a `.wav` file, *make sure that you have played it* (using the *Audio* selection on the *Graph* menu) so that the various sound parameters are properly set.

2.12.9 Print graphs

You can print graphs directly to paper output, or print graphs to a file (for later use in a document preparation system, such as LaTeX).

Printing graphs directly onto paper

To print the graphs on screen directly onto paper, you perform the following 6 steps.

1. From the list of choices, choose a *method of printing*, such as Small, top, left. These choices determine the size of the printout and where it is positioned on the page.
2. If you want the right-side labels (along the vertical) to be printed, then check the box labeled *Right labels.*
3. If you want the lower labels (along the horizontal) to be printed, then check the box labeled *Lower labels.*
4. If you want the caption for the graph to be printed, then check the box labeled *Caption.*
5. Select whether the printout should be Monochrome or Color. If you select *Color* then the printout will work properly if you have a printer that prints in color. Otherwise, it is best to select *Monochrome* which usually prints in black and white.
6. Click on one of the printout buttons, either *Print* or *Print and Eject.* If you click on *Print* then the graph will print, but will not eject from the printer. This allows for several different graphs to be printed on the same page. If you want to eject a printed page, then click on the button labeled *Eject.* If you want to print the graph and eject the page, then click on the button labeled *Print and Eject.*

Printing graphs to a file

To print graphs on screen to a file, you perform the following 2 steps.

1. Click on the button labeled *Save Printout to File.*
2. From the dialog box that appears, specify a file name for saving the printout data. Suppose this file name is `fname`, then the data for printing graphs is saved in two files: `fname.bmp` and `fname_labels.tex`. The first file contains the bit-mapped data needed to create the graph, the second file contains the data needed to label the x- and y-axes for this graph.

For further information on how to include this printout in the LaTeX document preparation system, see the article, *LaTeX Graphics with* FAWAV, which is in the `\LaTeX` subdirectory of the FAWAV directory (as both a LaTeX file and a PostScript file).

2.12.10 Clip

This procedure is used to clip a portion of a graph, thereby creating a smaller size graph (using less points). To clip a portion of a graph, isolate that portion on screen (by changing the horizontal X-interval that is displayed) and then right-click on the graph area and select *Clip graph.* If there is only one graph on screen, then the portion of the graph that is displayed is converted into a separate graph (displayed in a new window). Note: The number of points for the new graph must be a power of two, consequently if the portion of the graph displayed on screen does not consist of a power of two number of points, the clipped graph will have zeros appended to its end values in order to have a power of two number of points.

If there is more than one graph, then a dialog box appears on the right of the window, and you specify which graph you want to clip by entering its number in the text box labeled *Graph* and then clicking on the button labeled *Clip.*

2.12.11 Copy graph

This procedure is used to copy the data from a selected signal. If there is only one graph on screen, then this graph is immediately copied into memory from which it can then be pasted onto a 1D-form.

If there is more than one graph, a dialog box appears on the right of the window, and you specify which graph you want to copy by entering its number in the text box labeled *Graph* and then clicking on the button labeled *Copy.*

2.12.12 Paste graph

After a graph has been copied (as described in the previous item), then this choice is used for pasting it (graphing it) into a 1D-window. The graph can be pasted as many times as you wish as long as another graph is not copied in the meantime.

Note: If you choose to paste a graph on a screen that is already occupied by a graph that uses a different number of points, then a new window is opened and the graph is pasted into the new window.

2.12.13 Remove graph

This choice is used to remove a graph from the screen. If there is only one graph displayed, then it is immediately removed. If there is more than one graph, then a dialog box opens on the right of the screen, and you enter the number of the graph that you want to remove in the text box labeled *Graph*, and then click the button labeled *Remove.*

Warning: Once the graph has been removed, it cannot be restored. All of its data is lost, unless you have saved it.

2.12.14 Copy graph to clipboard

This procedure is used to copy graph data for use in other programs. If there is only one graph on screen, then the data for this graph is immediately copied into memory from which it can then be pasted onto a 1D-form. If there is more than one graph, then you specify which graph you want to copy by typing in its number in the textbox labeled *Graph* and then clicking on the button labeled *Copy.*

This procedure is included for exporting FAWAV function data to other programs, such as EXCEL, using the WINDOWS clipboard. If you are just copying and pasting within FAWAV, then we recommend that you use the *Copy graph* method described above.

2.12.15 Paste graph from clipboard

After a graph has been copied to the clipboard (as described above in *Copy graph to clipboard*), then it can be pasted into any 1D-form by right-clicking on the graphbox and selecting this procedure. The graph can be pasted as many times as you wish as long as other data is not copied to the Clipboard in the meantime.

Note: If you choose to paste a graph on a screen that is already occupied by a graph that uses a different number of points, then a new window is opened and the graph is pasted into the new window.

This procedure is mainly included for the purpose of importing external data (from programs such as EXCEL or NOTEPAD). If you are just copying and pasting within FAWAV, then we recommend that you use the methods *Copy graph* and *Paste graph* described above.

2.12.16 Display cursor coordinates

You select this choice in order to provide a readout of coordinates in the graph area. Its default setting is *off,* and it is listed without a checkmark ✓. When you select it that setting changes to *on* and this choice has a checkmark ✓ attached to it. When the setting is *on* the mouse cursor changes to a *crosshair* appearance within the graph area, and a readout is given of the x- and y-coordinates located at the central point of this crosshair.

Chapter 3

Spectrograms (Gabor transforms)

To compute a spectrogram for a displayed graph, select *Analysis* and *Spectrogram* from the 1D-form menu (as mentioned on p. 33 and p. 40). If only one graph is displayed then the *Spectrogram* window opens with that graph displayed at the bottom. If more than one graph is displayed, then you must enter the number for that graph and press the *Select* button. The *Spectrogram* window will then open up with the selected graph displayed at the bottom.

On the right side of the *Spectrogram* window will be a dialog box for producing the spectrogram. There are several aspects to this, including the following topics:

1. *Computing the spectrogram* See Section 3.1 on p. 45.
2. *Modifying the spectrogram* See Section 3.2 on p. 46.
3. *Inverting the spectrogram* See Section 3.3 on p. 47.
4. *Denoising the spectrogram* See Section 3.4 on p. 47.
5. *Changing the appearance of the spectrogram* See Section 3.5 on p. 47.
6. *Printing the spectrogram* See Section 3.6 on p. 48.
7. *Saving the spectrogram* See Section 3.7 on p. 48.
8. *Recording information* See Section 3.8 on p. 49.
9. *Spectrogram menu items* See Section 3.9 on p. 49.

We now discuss each of these topics.

3.1 Computing the spectrogram

This procedure (which can be invoked by selecting *Graph/Spectrogram* from the *Spectrogram* menu) is used to perform a calculation of a spectrogram of the signal displayed at the bottom of the window. You perform the following 4 steps:

1. In the text box labeled *Window length*, enter the length of the sliding window of data to be transformed. You must enter a power of two (either 16, 32, 64, 128, 256, 512, or 1024).
2. In the text box labeled *Step size*, enter the number of values by which the sliding window is shifted for each successive transform. This value must be at least 1 and no larger than the window length.

   Warning: A bad choice of *Window length* and *Step size* will produce a spectrogram that is incomplete. This usually occurs when either of these values are too small. If this happens, then select *Help/Initialize.*

3. Choose a *Filter* from the given list. The filters that can be selected are *Blackman, Gauss, Hamming, hanning,* and *Boxcar*. Here are descriptions of each of these choices:

   Blackman. This choice, called a *Blackman window,* modifies the data within a window of length N by multiplying the signal values (from $k = 0$ to $k = N$) by the window function

   $$0.42 + 0.5\cos(k\pi/N) + 0.08\cos(2k\pi/N)$$

   Blackman windowing reduces the "leakage" or "smearing" effect that arises from approximating continuous frequencies by discrete ones.

   Gauss. This choice, called a *Gauss window,* modifies the data within a window of length N by multiplying the signal values (from $k = 0$ to $k = N$) by the window function

   $$e^{-4.5(k/N)^2}$$

   Gauss windowing reduces the "leakage" or "smearing" effect that arises from approximating continuous frequencies by discrete ones.

   Hamming. This choice, called a *Hamming window,* modifies the data within a window of length N by multiplying the signal values (from $k = 0$ to $k = N$) by the window function

   $$0.54 + 0.46\cos(k\pi/N)$$

   Hamming windowing reduces the "leakage" or "smearing" effect that arises from approximating continuous frequencies by discrete ones.

   hanning. This choice, called a *Hanning window,* modifies the data within a window of length N by multiplying the signal values (from $k = 0$ to $k = N$) by the window function

   $$0.5 + 0.5\cos(k\pi/N)$$

   Hanning windowing reduces the "leakage" or "smearing" effect that arises from approximating continuous frequencies by discrete ones.

   Boxcar. This choice, called either a *Boxcar window* or a *rectangular window,* modifies the data within a window of length N by multiplying the signal values (from $k = 0$ to $k = N$) by the window function

   $$1$$

   This choice is included mostly in order to illustrate the need for the other window types. If Boxcar windowing is done, then the spectrogram will appeared as if it is "smeared." This "smearing" effect, also known as "leakage," arises from approximating continuous frequencies by discrete ones.

4. Click on the button labeled *Compute.* This will produce the spectrogram of the graph you have specified.

3.2 Modifying the spectrogram

There are two ways to modify the spectrogram. You can perform a *Threshold* operation, or you can perform a *Graphing* operation.

Thresholding

To apply a threshold to a spectrogram select *Graph/Threshold* and then enter a value (no more than the maximum value displayed above the input box) for this threshold, and click the *Apply* button. All values of the spectrogram whose magnitudes are less than this value are set equal to zero, and the new spectrogram is displayed. Thresholding is useful for removing background noise. Usually inversion is performed after thresholding.

Graphing

Graphing is useful for selecting a portion of the spectrogram and performing modifications of the spectrogram. For example, if you enter the formula

```
g1(0.2 <= x <= 0.7)(158 <= y <= 1227)
```

then all values of the spectrogram for points $(\mathtt{x}, \mathtt{y})$ that are outside of the rectangle $[0.2, 0.7] \times [158, 1227]$ will be set equal to zero, and all values for points $(\mathtt{x}, \mathtt{y})$ that are within the rectangle will be left as is. The variable `g1` refers to the magnitude of the spectrogram values only.

The graphing procedure uses methods from the 2D-form graphing procedures. See Chapter 5 on p. 59 for more information. See also Section 3.4 on p. 47 which describes how graphing is used for denoising a spectrogram. There is also online help available (just select *Help / Help for plotting graphs* from the spectrogram menu when the graphing text box is open).

Usually an inversion operation is performed after making a graphing modification of the spectrogram.

3.3 Inverting the spectrogram

This procedure, which is invoked by selecting *Graph/Invert*, will perform an inversion of the spectrogram back to a signal. If you have not modified the spectrogram, then this *Invert* procedure should produce an excellent approximation of the original signal (equal to the original except for tiny floating point roundoff errors).

You can also *Invert* after making modifications. This is useful for removing noise (after thresholding for example), or for capturing a particular frequency band (say from a recorded voice or a piece of recorded music) by using the graphing method.

3.4 Denoising the spectrogram

To perform various types of denoising of a spectrogram, you select *Graph/Denoise* and a denoising function will appear on the left of the window (see Figure 3.1). You can then click the *Plot* button to denoise the spectrogram, followed by *Graph/Invert* to produce a denoised signal.

You can also modify the denoising function that appears in the graphing window to perform more elaborate denoising. Examples of this are given in *A Primer on Wavelets and their Scientific Applications, Second Edition.*

3.5 Changing the appearance of the spectrogram

You can change the appearance of the spectrogram by selecting *Display style* from the *View* menu. The dialog box that appears on the right of the window allows you to change the way the spectrogram is displayed in the following three ways:

1. Select a method by which *Magnitude* of the spectrogram's values are computed. The four choices are

   Log (global) Magnitudes are displayed on a logarithmically graded intensity scale relative to the maximum of all spectrogram magnitudes.

   Log (local) Magnitudes are displayed on a logarithmically graded intensity scale relative to the maximum of all spectrogram magnitudes along each vertical time-slice.

   Linear (global) Magnitudes are displayed on a linearly graded intensity scale relative to the maximum of all spectrogram magnitudes.

   Linear (local) Magnitudes are displayed on a linearly graded intensity scale relative to the maximum of all spectrogram magnitudes along each vertical time-slice.

2. Select a method for how the signal is displayed at the bottom of the window. If you select *Dots*, then the signal values are plotted as isolated dots. If you select *Lines*, then line segments are drawn between signal values, thus producing a more connected graph.

3. Select whether the *Magnitude* of the spectrogram values are plotted, or whether *Phase values* are plotted. When phase values are plotted, the darker values are negative and the whiter values are positive.

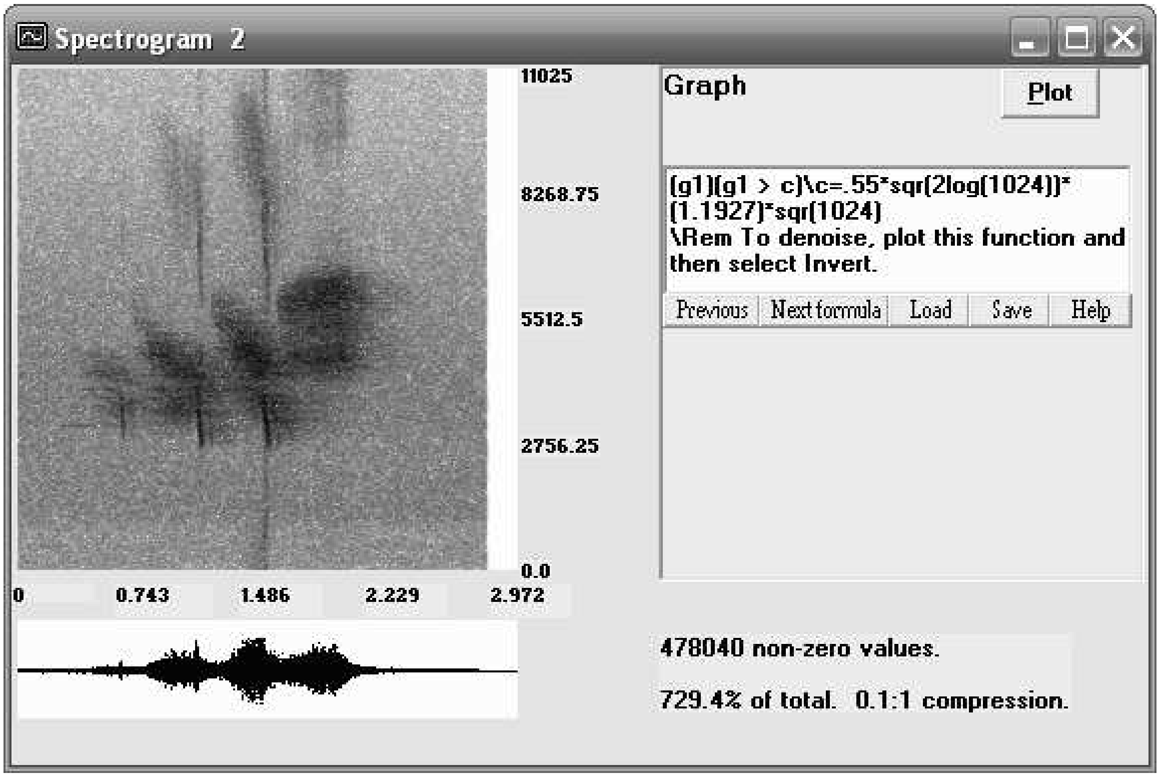
3.6 Printing the spectrogram

To print the spectrogram, select *Print* from the *File* menu. A dialog box appears from which you can choose to either print the spectrogram window, or copy the spectrogram window to the clipboard, or save the spectrogram window to a file (as a `.bmp` image).

Warning: In each of these cases the entire window of the spectrogram form, as it appears on screen, is printed (or copied, or saved). If you do not want the dialog box on the right side of the window to be printed, then you must first slide the right edge of the window over so as to hide this dialog box.

3.7 Saving the spectrogram

To save the spectrogram, select *Save As...* from the *File* menu. A standard dialog box appears from which you can choose where to save the spectrogram and what name to save it under.



```
(g1)(g1 > c)\c=.55*sqr(2log(1024))*
(1.1927)*sqr(1024)
\Rem To denoise, plot this function and
then select Invert.
```

478040 non-zero values.

729.4% of total. 0.1:1 compression.

Figure 3.1
Example of a denoising function.

3.8 Recording information

Selecting *View/Information* allows you to view some basic information about the spectrogram (type, window length, etc.). You can also write your own information into this box, and it will be saved along with the spectrogram (see Saving the spectrogram).

3.9 Menu Items for Spectrogram form

3.9.1 File

The *File* menu has the following 9 items.

1. *New 1D-form* This allows you to open a new 1D-form. See Chapter 2 beginning on p. 11.
2. *New 2D-form* This allows you to open a new 2D-form. See Chapter 5 beginning on p. 59.
3. *New Audio Editor* This allows you to open a new Audio Editor form. See Chapter 6 beginning on p. 85.
4. *New Image Processor* This allows you to open a new Image Processor form. See Chapter 7 beginning on p. 89.
5. *Open* This allows you to open a previously saved spectrogram (a file with extension `.spg`).
6. *Save* This allows you to resave a spectrogram file (extension `.spg`) that you have modified.
7. *Save As* This allows to save a spectrogram form to a file (extension `.spg`).
8. *Print* This allows you to print the spectrogram. See Section 3.6 on p. 48 for further discussion.
9. *Close* This closes the spectrogram form. Warning: You are not prompted as to whether you wish to save your work.

3.9.2 View

The *View* menu contains the following 5 items.

1. *View cursor coordinates* This is a switch on the View menu, switched *on* and *off* by clicking on it. When this choice is *on,* then by moving the mouse cursor over the spectrogram plot you can read off the approximate coordinates (time, frequency) of the point of the mouse arrow. This is especially useful when you want to modify regions of the spectrogram (using the *Graph/Plot* procedure).

2. *Display style* This choice allows you to change the appearance of the spectrogram. It is discussed in detail in Section 3.5 on p. 47.

3. *Information* Selecting *View/Information* allows you to view some basic information about the spectrogram (type, window length, etc.). You can also write your own information into this box, and it will be saved along with the spectrogram (saving the spectrogram is discussed in Section 3.7 on p. 48).

4. *Modification Report* This dialog box, which is activated by selecting *View/Modification Report* records any modifications that have been made on the spectrogram. Such modifications can be done by thresholding or by a graphing operation.

   Warning (bug): Do not modify the statements made in this box. Doing so will corrupt the spectrogram file, when you save it. This is because FAWAV uses these statements to reconstruct the spectrogram when you open the saved file.

5. *Arrange* This choice will perform automatic rearrangement of the positions of multiple windows within the main FAWAV window.

The *View* menu always has at its end a listing of all the windows that are open within the FAWAV workspace, so that you can select which window to be the active one. [The presently active one is marked with a checkmark (✓).]

3.9.3 Graph

The graph menu has the following 8 items:

Spectrogram

This choice opens up the dialog box for computing a spectrogram. See Section 3.1 on p. 45.

Plot

This choice allows you to plot a graph in the spectrogram window. Using the variable `g1` you can modify spectrogram magnitudes. Examples of graphing are given in Section 3.2 on p. 46. More examples are discussed in *A Primer on Wavelets and their Scientific Applications, Second Edition.*

Audio

This choice allows you to play the signal whose spectrogram has been computed. The options are all discussed in Section 2.8 on p. 31.

Threshold

To apply a threshold to a spectrogram you enter a value (no more than the maximum value displayed above the input box) for this threshold and then click the *Apply* button.

All values of the spectrogram whose magnitudes are less than this value are set equal to zero, and the new spectrogram is displayed. If you are not satisfied with the results, you can restore the original spectrogram by selecting *Restore* from the *Graph* menu.

Denoise

This choice allows you to perform denoising. For more details see Section 3.4 on p. 47.

Percussion scalogram

This choice is used for analyzing percussion passages in music. It is described in *A Primer on Wavelets and their Scientific Applications, Second Edition.*

Invert

This selection inverts the spectrogram. See Section 3.3 on p. 47.

Restore

This selection restores the original spectrogram. It is usually invoked after a modification which did not produce the desired results.

3.9.4 Help

This item typically has two choices. There is a general choice, *Help (spectrogram)*, which you choose to get the general help for the spectrogram form. When a dialog box is open on the right side of the screen, there will be a second choice specifically designed to access the desired topic in the help file (a form of interactive help).

3.10 The Popup menu for spectrograms

If you right-click on the graph area on the left of a spectrogram form, then a popup menu will appear with 9 items. We now discuss each of these items.

3.10.1 Print

This selection is used for printing the spectrogram. See Section 3.6 on p. 48.

3.10.2 Plot graph

This selection is the same as *Graph/Plot.* See Section 3.2 on p. 46.

3.10.3 Threshold

This selection is the same as *Graph/Threshold.* See Section 3.2 on p. 46.

3.10.4 Invert

This selection is the same as *Graph/Invert*. See Section 3.3 on p. 47.

3.10.5 Restore

This selection is the same as *Graph/Restore*; it restores the original spectrogram. It is usually invoked after a modification which did not produce the desired results.

3.10.6 Display style

This selection is the same as *View/Display style*. See Section 3.5 on p. 47.

3.10.7 Copy signal

This selection will graph the signal displayed at the bottom of the window in a new 1D-form.

3.10.8 Vertical slice

This selection, which is available when the viewing of cursor coordinates has been activated (via *View/View cursor coordinates*), will graph the real and imaginary parts of the spectrogram in a new 1D-form at the time-coordinate displayed when you right-click on the spectrogram. You can then plot the spectrum of the signal at the time-coordinate by plotting the function `sqr(g1(x)^2+g2(x)^2)`, or you could plot its power spectrum by plotting `g1(x)^2+g2(x)^2`.

3.10.9 Horizontal slice

This selection, which is available when the viewing of cursor coordinates has been activated (via *View/View cursor coordinates*), will graph the real and imaginary parts of the spectrogram in a new 1D-form at the frequency-coordinate displayed when you right-click on the spectrogram.

Chapter 4

Scalograms (Continuous wavelet transforms)

To compute a scalogram for a displayed graph, select *Analysis* and *Scalogram* from the 1D-form menu (as mentioned on p. 33 and p. 40). If only one graph is displayed then the *Scalogram* window opens with that graph displayed at the bottom. If more than one graph is displayed, then you must enter the number for that graph and press the *Select* button. The *Scalogram* window will then open up with the selected graph displayed at the bottom.

On the right side of the *Scalogram* window will be a dialog box for producing the scalogram. There are several aspects to this, including the following topics:

1. *Computing the scalogram* See Section 4.1 on p. 53.
2. *Modifying the scalogram* See Section 4.2 on p. 54.
3. *Inverting the scalogram* See Section 4.3 on p. 54.
4. *Changing the appearance of the scalogram* See Section 4.4 on p. 55.
5. *Printing the scalogram* See Section 4.5 on p. 55.
6. *Saving the scalogram* See Section 4.6 on p. 56.
7. *Recording information* See Section 4.7 on p. 56.
8. *Scalogram menu items* See Section 4.8 on p. 56.

We now discuss each of these topics.

4.1 Computing the scalogram

This procedure (which is normally invoked by selecting *Graph/Scalogram* from the menu) is used to perform a calculation of a scalogram of the signal displayed at the bottom of the window. You perform the following 6 steps:

1. In the text box labeled *Octaves* enter the number of octaves. This should be a positive integer from 1 to 256. Normally this number should be no greater than 8 or 10.
2. In the text box labeled *Voices* enter the number of voices. This is a positive integer. The product of the number of voices times the number of octaves is restricted to be no greater than 256.
3. From the text box labeled *Wavelet* choose the analyzing wavelet to be used.

4. In the text box labeled *Width* enter the width to be used for the wavelet. This should be a positive number (typically no greater than half the length of the interval over which the signal is displayed). The shorter the width, the more the energy of the wavelets are concentrated within smaller regions. The width determines the constant of proportionality, c, that relates the reciprocal scale values (along the vertical axis of the scalogram) to frequency values (the smaller the width, the larger this constant of proportionality c).

5. If you choose one of the Gabor wavelets, then you also must enter a value for frequency in the text box labeled *Freq.* With Gabor wavelets, there is a simple formula that relates width, w, and frequency, v, to the constant of proportionality, c. It is
$$c = v/w.$$

6. To plot the scalogram, click on the button labeled *Compute*.

The general theory of scalograms is described in Chapter 6 of *A Primer on Wavelets and their Scientific Applications*, and also in the article *"Time-frequency Analysis of Musical Instruments,"* by J.S. Walker and J.F. Alm, referenced in *A Primer on Wavelets and their Scientific Applications*. The Mexican hat and Gabor scalograms are discussed in detail in *A Primer on Wavelets and their Scientific Applications*.

4.2 Modifying the scalogram

There are two ways to modify the scalogram. You can perform a *Threshold* operation, or you can perform a *Graphing* operation.

Thresholding

To apply a threshold to a scalogram select *Graph/Threshold* and then enter a value (no more than the maximum value displayed above the input box) for this threshold, and click the *Apply* button. All values of the scalogram whose magnitudes are less than this value are set equal to zero, and the new scalogram is displayed. Thresholding is useful for removing background noise. Usually inversion is performed after thresholding.

Graphing

Graphing is useful for selecting a portion of the scalogram and performing modifications of the scalogram. For example, if you enter the formula

```
g1(0.2 <= x <= 0.7)(158 <= y <= 1227)
```

then all values of the scalogram for points $(\mathtt{x}, \mathtt{y})$ that are outside of the rectangle $[0.2, 0.7] \times [158, 1227]$ will be set equal to zero, and all values for points $(\mathtt{x}, \mathtt{y})$ that are within the rectangle will be left as is.

With a complex-valued scalogram (i.e., the Gabor scalogram), the variable `g1` refers to the magnitude of the scalogram values only. The effect of the formula above on a Gabor scalogram, would be the same, however.

The graphing procedure uses methods from the 2D-form graphing procedures. See Chapter 5 on p. 59 for more information. There is also online help available (just select *Help/Help for plotting graphs* from the scalogram menu when the graphing text box is open).

Usually an inversion operation is performed after making a graphing modification of the scalogram.

4.3 Inverting the scalogram

This procedure, which is invoked by selecting *Graph/Invert*, will perform an inversion of the scalogram back to a signal. If you have not modified the scalogram, and you made a judicious choice of number of octaves and voices so as to capture a sufficient amount of the frequency content of the original signal, then this *Invert* procedure should produce a good/excellent approximation of the original signal.

You can also *Invert* after making modifications. This is useful for removing noise (after thresholding for example) or for capturing a particular frequency band (say from a recorded voice or a piece of recorded music).

There are three methods that you can choose for inverting. These three methods are:

1. *Wavelets.* This method produces an inversion by summing multiples of shifts and scalings of the basic wavelet used to produce the scalogram.

2. *Wavelets and Scaling functions.* This method includes, along with the wavelets, low-frequency data that produces what are called scaling functions. This method will sometimes perform better on signals with a large amount of energy in lower frequencies.

3. *Damped trigonometric functions.* This method consists of canceling out the effects of the correlations with the wavelets by simple division. It will sometimes work more effectively than the above two methods, but it bears no relation to the wavelets used to produce the scalogram.

4.4 Changing the appearance of the scalogram

You can change the appearance of the spectrogram by selecting *Display style* from the *View* menu. The dialog box that appears on the right of the window allows you to change the way the spectrogram is displayed in the following three ways:

1. Select a method by which *Magnitude* of the spectrogram's values are computed. The four choices are

   Log (global) Magnitudes are displayed on a logarithmically graded intensity scale relative to the maximum of all spectrogram magnitudes.

   Log (local) Magnitudes are displayed on a logarithmically graded intensity scale relative to the maximum of all spectrogram magnitudes along each vertical time-slice.

   Linear (global) Magnitudes are displayed on a linearly graded intensity scale relative to the maximum of all spectrogram magnitudes.

   Linear (local) Magnitudes are displayed on a linearly graded intensity scale relative to the maximum of all spectrogram magnitudes along each vertical time-slice.

2. Select a method for how the signal is displayed at the bottom of the window. If you select *Dots*, then the signal values are plotted as isolated dots. If you select *Lines*, then line segments are drawn between signal values, thus producing a more connected graph.

3. If you are plotting a Gabor scalogram, then you can also select whether the *Magnitude* of the scalogram values are plotted, or whether *Phase values* are plotted. When phase values are plotted, the darker values are negative and the whiter values are positive.

4.5 Printing the scalogram

To print the scalogram, select *Print* from the *File* menu. A dialog box appears from which you can choose to either print the scalogram window, or copy the scalogram window to the clipboard, or save the scalogram window to a file (as a `.bmp` image).

Warning: In each of these cases the entire window of the scalogram form, as it appears on screen, is printed (or copied, or saved). If you do not want the dialog box on the right side of the window to be printed, then you must first slide the right edge of the window over so as to hide this dialog box.

4.6 Saving the scalogram

To save the scalogram, select *Save As...* from the *File* menu. A standard dialog box appears from which you can choose where to save the scalogram and what name to save it under.

4.7 Recording information

Selecting *View/Information* allows you to view some basic information about the scalogram (type, window length, etc.). You can also write your own information into this box, and it will be saved along with the scalogram (see Saving the scalogram).

4.8 Scalogram menu items

4.8.1 File

The *File* menu has the following 9 items.

1. *New 1D-form* This allows you to open a new 1D-form. See Chapter 2 beginning on p. 11.
2. *New 2D-form* This allows you to open a new 2D-form. See Chapter 5 beginning on p. 59.
3. *New Audio Editor* This allows you to open a new Audio Editor form. See Chapter 6 beginning on p. 85.
4. *New Image Processor* This allows you to open a new Image Processor form. See Chapter 7 beginning on p. 89.
5. *Open* This allows you to open a previously saved scalogram (a file with extension `.spg`).
6. *Save* This allows you to resave a scalogram file (extension `.spg`) that you have modified.
7. *Save As* This allows to save a scalogram form to a file (extension `.spg`).
8. *Print* This allows you to print the scalogram. See Section 4.5 on p. 55 for further discussion.
9. *Close* This closes the scalogram form. Warning: You are not prompted as to whether you wish to save your work.

4.8.2 View

The *View* menu contains the following 6 items.

1. *View cursor coordinates* This is a switch on the View menu, switched *on* and *off* by clicking on it. When this choice is *on*, then by moving the mouse cursor over the scalogram plot you can read off the approximate coordinates (time, frequency) of the point of the mouse arrow. This is especially useful when you want to modify regions of the scalogram (using the *Graph/Plot* procedure).
2. *Display style* This choice allows you to change the appearance of the scalogram. It is discussed in detail in Section 4.4 on p. 55.
3. *View only scalogram* This choice removes the right side of the form (where dialog boxes appear) so that only the region where the scalogram is displayed appears. After making this choice, it can be undone by clicking on it again.
4. *Information* Selecting *View/Information* allows you to view some basic information about the scalogram (type, window length, etc.). You can also write your own information into this box, and it will be saved along with the scalogram (saving the scalogram is discussed in Section 4.6 on p. 56).

5. *Modification Report* This dialog box, which is activated by selecting *View/Modification Report* records any modifications that have been made on the scalogram. Such modifications can be done by thresholding or by a graphing operation.

   Warning (bug): Do not modify the statements made in this box. Doing so will corrupt the scalogram file, when you save it. This is because FAWAV uses these statements to reconstruct the scalogram when you open the saved file.

6. *Arrange* This choice will perform automatic rearrangement of the positions of multiple windows within the main FAWAV window.

The *View* menu always has at its end a listing of all the windows that are open within the FAWAV workspace, so that you can select which window to be the active one. [The presently active one is marked with a checkmark (✓).]

4.8.3 Graph

The graph menu has the following 6 items:

Scalogram

This choice opens up the dialog box for computing a scalogram. See Section 4.1 on p. 53.

Plot

This choice allows you to plot a graph in the scalogram window. Using the variable `g1` you can modify scalogram values (or magnitudes when a Gabor scalogram is computed). Examples of graphing are given in Section 3.2 on p. 46. More examples are discussed in *A Primer on Wavelets and their Scientific Applications, Second Edition.*

Audio

This choice allows you to play the signal whose scalogram has been computed. The options are all discussed in Section 2.8 on p. 31.

Threshold

To apply a threshold to a scalogram you enter a value (no more than the maximum value displayed above the input box) for this threshold and then click the *Apply* button.

All values of the scalogram whose magnitudes are less than this value are set equal to zero, and the new scalogram is displayed. If you are not satisfied with the results, you can restore the original scalogram by selecting *Restore* from the *Graph* menu.

Invert

This selection inverts the scalogram. See Section 4.3 on p. 54.

Restore

This selection restores the original scalogram. It is usually invoked after a modification which did not produce the desired results.

4.8.4 Help

This item typically has two choices. There is a general choice, *Help (scalogram),* which you choose to get the general help for the scalogram form. When a dialog box is open on the right side of the screen, there will be a second choice specifically designed to access the desired topic in the help file (a form of interactive help).

4.9 The Popup menu for scalograms

If you right-click on the graph area on the left of a scalogram form, then a popup menu will appear with 9 items. We now discuss each of these items.

4.9.1 Print

This selection is used for printing the scalogram. See Section 4.5 on p. 55.

4.9.2 Plot graph

This selection is the same as *Graph/Plot.* See Section 4.2 on p. 54.

4.9.3 Threshold

This selection is the same as *Graph/Threshold.* See Section 4.2 on p. 54.

4.9.4 Invert

This selection is the same as *Graph/Invert*. See Section 4.3 on p. 54.

4.9.5 Restore

This selection is the same as *Graph/Restore*; it restores the original scalogram. It is usually invoked after a modification which did not produce the desired results.

4.9.6 Display style

This selection is the same as *View/Display style*. See Section 4.4 on p. 55.

4.9.7 Copy signal

This selection will graph the signal displayed at the bottom of the window in a new 1D-form.

4.9.8 Vertical slice

This selection, which is available when the viewing of cursor coordinates has been activated (via *View/View cursor coordinates*), will graph the real and imaginary parts of the scalogram in a new 1D-form at the time-coordinate displayed when you right-click on the scalogram.

4.9.9 Horizontal slice

This selection, which is available when the viewing of cursor coordinates has been activated (via *View/View cursor coordinates*), will graph the real and imaginary parts of the scalogram in a new 1D-form at the frequency-coordinate displayed when you right-click on the scalogram.

Chapter 5

Processing 2D Signals (2D-form)

To work with 2D signals, click on the *File* menu, and then select *New 2 Dim.* The window shown in Figure 5.1 appears. A description of each of the 9 menu items will be given at the end of this chapter (see Section 5.11 on p. 76). But first we describe how to use this form for basic signal processing. The procedures that are available for working with 2D signals include the following:

1. *Graphing formulas and external data.* You can plot graphs of elementary functions studied in algebra and calculus (see Section 5.1 on p. 60.) Or, you can load external data (see Section 5.1.12 on p. 66). Or, you can load image

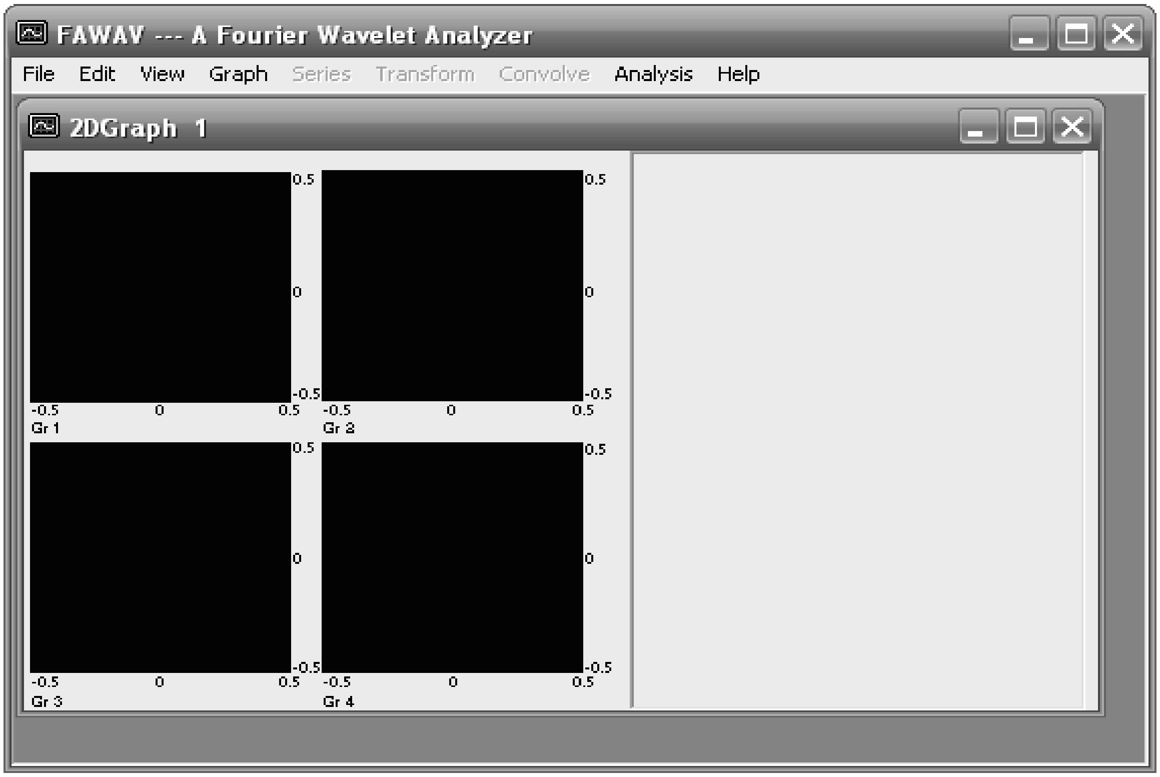


Figure 5.1
The window that opens when *New 2D-form* is selected.

files (see Section 5.1.10 on p. 65).

2. *Wavelet transforms.* These are the Haar, DaubJ, CoifI transforms, and biorthogonal wavelet transforms, described in *A Primer on Wavelets and their Scientific Applications*. See Section 5.2 on p. 66.

3. *Wavelet series.* These are the Haar, DaubJ, CoifI series, and a selection of biorthogonal wavelets series, described in *A Primer on Wavelets and their Scientific Applications*. See Section 5.3 on p. 67.

4. *Fourier series and transforms.* These are the classic Fourier series and transforms implemented using FFTs. See Section 5.6 on p. 68.

5. *Convolutions and Correlations.* There are procedures for calculating the convolution of two signals, for calculating the correlation between two signals, and for calculating an auto-correlation of one signal. See Section 5.8 on p. 71.

6. *Image Compression and Denoising.* Two important methods in image processing are compression (Section 5.9 on p. 72) and denoising (Section 5.10 on p. 74).

There are also a number of routines for numerical analysis of 2D signals, these are grouped under the *Analysis* choice on the menu (see Section 5.11.8 on p. 79).

5.1 Graphing formulas and external data

Data can be plotted in the 2D-form in one of three ways: (1) plotting an elementary formula from algebra or calculus, (2) loading an image file, (3) loading a data file. We describe each of these methods in order.

5.1.1 Plotting a formula

To plot an elementary formula you select *Graph,* followed by *Plot* from the menu. A dialog box opens up on the right of the window, with a text box at the top for entering your formula. To graph a formula over an interval, perform the following steps:

1. Enter a formula in the text box at the top of the dialog box. The simplest way to begin graphing formulas is to use the kind that graphing calculators use. For example, the following two formulas:

   ```
   3x + 7y

   4sin(3pi xy-2) - 5cos(4pi xy-5)
   ```

   are both valid formulas in FAWAV. The second formula uses the expression `pi` for the constant π. For many more examples, see Sections 5.1.2 through 5.1.12.

2. Specify the value of L to be used for the domain $[-L, L] \times [-L, L]$. For example, if you specify that $L = 2$ then the domain used for graphing will be $[-2, 2] \times [-2, 2]$.

3. Click on the command button labeled *Plot.* Assuming all the data are correctly specified, this will produce a graph in the graph area on the left of the window.

Formula command buttons There are five command buttons at the bottom of the formula text box. Here are descriptions of each of these command buttons.

Previous If you have graphed more than one formula, then by clicking on this button you can recall previously entered formulas.

Next formula If you have clicked on the *Previous* button one or more times, then you can go forward in the list of formulas by clicking on this button. If you have not clicked the *Previous* button, then clicking this button will clear the text box completely and you can enter a new formula.

Load Click on this button in order to retrieve a formula that you previously saved.

Save Click on this button in order to save the formula that is presently displayed in the text box.

Help Click on this button in order to get help on creating formulas.

5.1.2 Formulas for 2-dimensional graphs

Formulas for graphing are constructed using pre-defined functions, as well as various logical operations, or sums, or products. In Sections 5.1.3 through 5.1.9 we discuss all of the operations that can be employed in graphing formulas. These operations are mostly the same as for 1D formulas (with the exception of graph variables, and that y is NOT a parameter in 2D), but we list them here again for the convenience of the reader.

5.1.3 Functions supported

There are a number of pre-defined functions. We list them here in two categories: (1) the *elementary functions* of the kind encountered in calculus, and (2) certain *special functions* of a miscellaneous nature.

Elementary functions

FAWAV employs 11 elementary functions, which we now list.

1. `abs` The absolute value function, `abs( )`, returns the absolute value of its argument, e.g. $\texttt{abs}(-5) = 5$ and $\texttt{abs}(5) = 5$.

2. `cos` The cosine function is invoked using `cos( )`. For example,

```
cos(3x+7y)

cos(xˆ2 + yˆ4 - 3) + 6x - 3y + 8
```

   are both valid formulas involving the cosine function.

3. `cosh` The hyperbolic cosine function, `cosh( )`, returns the hyperbolic cosine of its arguments. By definition, `cosh(x)` satisfies

```
cosh(x) = [exp(x) + exp(-x)]/2
```

   where `exp` is the exponential function.

4. `exp` The exponential function `exp( )` is the standard, base e, exponential function. For example,

```
exp(4xy)

eˆ(4xy)
```

   are two equivalent expressions using the exponential function.

5. `gri` The greatest integer function `gri( )` is defined as follows:

   $$\texttt{gri(x)} = \texttt{(largest integer that is} \leq \texttt{x)}$$

   Or, an equivalent definition is

   $$\texttt{gri(x)} = \texttt{(closest integer to the left of x on the number line)}$$

6. `log` The natural logarithm function is defined by `log( )`. You can also use `Ln( )`.

7. `sgn` The sign function, `sgn( )`, is defined by

   $$\texttt{sgn(x)} = \begin{cases} 1 & \text{if } \texttt{x > 0} \\ 0 & \text{if } \texttt{x = 0} \\ -1 & \text{if } \texttt{x < 0} \end{cases}$$

   You can also use `sign( )` for the sign function.

8. `sin` The sine function is invoked using `sin( )`. For example,

   ```
   sin(3x)

   sin(5xy - 6) + xyˆ2 - 2
   ```

   are both valid formulas involving the sine function.

9. `sinh` The hyperbolic sine function, `sinh( )`, returns the hyperbolic sine of its arguments. By definition, `sinh(x)` satisfies

   ```
   sinh(x) = [exp(x) - exp(-x)]/2.
   ```

10. `sqr` The positive square root function, `sqr( )`, computes the positive square root of its argument. For example, `sqr(4) = 2` and `sqr(9) = 3`. Also `sqr(0) = 0`. The square roots of negative numbers are not real, so in FAWAV applying the `sqr` function to a negative number produces an error message.

11. `tan` The tangent function is invoked using `tan( )`. For example,

    ```
    tan(3x+6y)

    tan(5x - 6) + x - 2y + 9
    ```

    are both valid formulas involving the tangent function.

Special functions

FAWAV employs 7 special functions, which we shall now describe.

1. `bell` The function `bell( )` is defined as follows. If $|\mathtt{x}| \geq 0.75$, then `bell(x) = 0`. If $|\mathtt{x}| \leq 0.25$, then `bell(x) = 1`. Finally, if $0.25 < |\mathtt{x}| < 0.75$, then `bell(-x) = bell(x)` (that is, the bell function is even), and we have for $0.25 < \mathtt{x} < 0.75$, that `bell(x)` is defined by

   ```
   bell(x) = sin(aw)sin(aw)\a=0.5pi\w=sin(av)sin(av)
             \v=sin(au)sin(au)\u=1.5 - 2x
   ```

More discussion of the `bell` function is given in the 1D chapter (see Section 2.1.3 on p. 15).

2. `del` The discrete delta function, `del( )`, is defined as follows

   $$\texttt{del}(\mathrm{x}) = \begin{cases} 0 & \text{if } \texttt{|x| > 0} \\ 1 & \text{if } \texttt{x = 0} \end{cases}$$

   In other words, `del(x)` is 0 if `x < 0` or `x > 0`, and `del(0)` equals 1.

3. `fac` The factorial function, `fac( )`, returns the factorial of a non-negative integer between 0 and 70. For example, `fac(5) = 120`. If the argument of `fac( )` is not an integer, then an error message is returned. A formula such as

   ```
   sumk(x^k / fac(k)) \k = 0, 8
   ```

   is allowed, and returns a partial sum of the Taylor series for e^{x} at $\mathtt{x} = 0$.

4. `ran` The random number function, `ran( )`, gives a value that is a random number between -1 and 1, with a mean value of 0. Random values are assigned no matter what the argument of `ran( )` is.

   For instance, the formula

   ```
   ran(0)
   ```

   will not produce a constant function. It will produce a randomly varying function with values between -1 and 1. The values of these random numbers are approximately uniformly distributed, i.e., the values of `ran( )` are approximately the values of a random variable with uniform probability density.

5. `rang` The random number function, `rang( )`, gives values that are approximately distributed like that of a Gaussian random variable. Random values are assigned no matter what the argument of `rang( )` is.

   For instance, the formula

   ```
   rang(0)
   ```

   will not produce a constant function. It will produce a randomly varying function.

   The function `rang( )` is used to create an approximation of random noise with a Gaussian normal probability. This type of random noise is called Gaussian white noise. The function `rang( )` is produced by adding N repetitions of `ran( )` and dividing by the square root of N. This number N is 144 by default. To change the number N of terms, enter a formula like the following

   ```
   rang(x)\noiseterms=400
   ```

   In this case N is set equal to 400. You must specify a value for N between 1 and 30000.

6. `rect` The rectangle function, `rect( )`, is defined by

   $$\texttt{rect(x)} = \begin{cases} 1 & \text{if } \texttt{|x| < 0.5} \\ 0.5 & \text{if } \texttt{|x| = 0.5} \\ 0 & \text{if } \texttt{|x| > 0.5} \end{cases}$$

   For example, to create a function equal to 1 over the rectangle $(a, b) \times (c, d)$ and equal to 0 outside that rectangle, you enter the formula

```
rect((x-u)/w)rect((y-v)/t)
\u=(a+b)/2\w=b-a\v=(c+d)/2\t=z-c
```

Notice that in this formula the parameter `z` is used instead of `d`. That is because the allowed parameters in 2D are `a`, `b`, `c`, `m`, `t`, `u`, `v`, `w`, `z`.

7. `sinc` The sinc function, `sinc( )`, is defined by

$$\operatorname{sinc}(x) = \frac{\sin \pi x}{\pi x}$$

for x not equal to zero. If $x = 0$, then $\operatorname{sinc}(0) = 1$.

5.1.4 Graph operations

If you have drawn the graph of at least one function on the screen, then you can assign values from these function(s) using the graph variables `g1`, `g2`, `g3`, and `g4`.

For example, suppose 3 graphs are displayed on screen. Then the following formula:

```
2g1 - 4g3
```

assigns twice the values of graph 1 minus four times the values of graph 3 to the plotted function.

Warning: Do NOT use a graphical function with a higher index than the number of displayed graphs. Also, FAWAV does not support graph variables beyond `g4`. Notice also that, unlike in 1D, the graph variables do NOT accept arguments in 2D. For instance, if you enter `g1(x)` in 2D that will produce the graph of the values of $g1$ multiplied by `x`. To graph just the values of graph 1, you would enter just `g1`.

5.1.5 Sums

There are four sum operators, `sumi( )`, `sumj( )`, `sumk( )`, and `sumn( )`. For example, the formula

```
sumi(x^i y^(i+1))\i=1,7
```

produces the same function as the more complicated formula

```
xy^2 + x^2y^3 + x^3y^4 + x^4y^5 + x^5y^6 + x^6y^7 + x^7y^8
```

Warning: You cannot use a sum operator twice in the same formula. For example, the formula

```
sumk(cos(kxy)) - sumk(kx-y)\k=3,5
```

will not be accepted by FAWAV. You should use instead

```
sumk(cos(kxy)-kx+y) \k=3,5
```

5.1.6 Products

There are two operators for performing products, `prodp( )`and `prodq( )`. For example, the following formula

```
prodp(xy-p)\p=-4,3
```

produces the same graph as the following more complicated formula

```
(xy+4)(xy+3)(xy+2)(xy+1)xy(xy-1)(xy-2)(xy-3)
```

Warning: You cannot use a product operator twice in the same formula. For example, the formula

```
prodp(x-p) - prodp(py)\p=3,5
```

will not be accepted by FAWAV. You should use instead

```
prodp(x-p) - prodq(qy)\p=3,5\q=3,5
```

5.1.7 Logic

There are three logical operations, `&` (and), `or`, and `not( )`. These are Boolean operations which are meant to handle values of either 1 (for TRUE) or 0 (for FALSE). For example, the following formula

```
(x>3 or y<-2)sin(3x) + ((-2<=x)&(y<=3))cos(x)
```

produces a function that adds $\sin(3\mathtt{x})$, if $\mathtt{x} > 3$ or if $\mathtt{y} < -2$, to $\cos(\mathtt{x})$ when $-2 \leq \mathtt{x}$ and $\mathtt{y} \leq 3$.

5.1.8 Inequalities

The inequality operations are `>`, `>=`, `<`, and `<=`. For example, the formula

```
(2<xy<=4)sin(xy) + (xy>4)cos(xyˆ2)
```

produces a function that equals $\sin(\mathtt{xy})$ for $2 < \mathtt{xy} \leq 4$, and equals $\cos(\mathtt{xy}^2)$ for $\mathtt{xy} > 4$, and equals 0 for all other values of x and y.

5.1.9 Command strings

Command strings are used at the end of a formula for performing substitutions or other tasks. They are preceded by the backslash character `\`. Here are a few examples of using command strings.

1. *Substitution.* The following formula shows how a command string performs several substitutions:

   ```
   cos(ax) + bsin(cˆ2/b)
   \a=2\b=x+y+c\c=3x-y
   ```

   Notice that the parameters can be constant or variable and that they can invoke other parameters as well.

   The allowed parameters are `a`, `b`, `c`, `m`, `t`, `u`, `v`, `w`, `z`. (Note that, unlike in 1D, the variable `y` cannot function as a parameter.)

2. *Limits for a sum.* The following formula shows how a command string is used for specifying the limits for a sum:

   ```
   1/2 + sumk[cos(kxyˆ2)]\k=-4,7
   ```

   Here the sum is from $\mathtt{k} = -4$ to $\mathtt{k} = 7$.

5.1.10 Loading an image file

Gray-scale images can be loaded into the 2D-form. Some standard image file formats can be loaded (e.g., `.bmp`, `.jpg`, etc.), but they must be N by N square images, where N is either 64, 128, 256, 512, or 1024. If a gray-scale image is not square, then it will be loaded into the Image Processor form (see Chapter 7 on p. 89). There are no plans at this time for FAWAV to handle color images, a good free program for color images is IMAGE ANALYZER.

To load an image, you right-click on a graph region and select *Load/Image* from the popup menu. A standard file dialog box will appear from which you can select the file containing the graph's data.

5.1.11 Saving an image file

To save an image in the form, you right-click on it and select *Save/Image* from the popup menu. A dialog box will appear on the screen which will allow you to specify the name of the file and the type of image format to use for saving.

Warnings: (1) If the graph that you want to save is NOT a 256-level gray-scale image, then saving it using this format will usually produce a corrupted file that you cannot later reload. (2) When saving an image file (in any format other than `.pgm`), do NOT include any periods in the file name. For example, the file name

```
noisy barb (mult. noise)
```

will lead to an “Error saving file” message in all formats (except `.pgm` format). If you change the file name to

```
noisy barb (mult noise)
```

it will be saved without error. Either file name will be saved without error in the `.pgm` format.

5.1.12 Loading external (binary) files

To display a graph that was previously saved (using *Save graph/Graph*), you right-click on a graph region and select *Load/Graph* from the popup menu. A standard file dialog box will appear from which you can select the file containing the graph’s data.

It is possible to load external, binary data files (not necessarily created by FAWAV). The data must be in the following format. Except for the initial dimension number (which is a 4-byte integer), the data should consist of 4-byte single precision numbers. The first number specifies the square dimensions of the data (either 64, 128, 256, or 1024) and is saved as a 4-byte integer. The next number specifies the square, centered at the origin, over which the data are to be graphed. For example, if it is 6.5, then the data will be graphed over the square $[-6.5, 6.5] \times [-6.5, 6.5]$. The remaining numbers are the image values, to be listed according to the scheme for an image array described in Section 4.1 of *A Primer on Wavelets and their Scientific Applications*, with $M = N$ and N being one more than the dimension number (either 65, 129, 513, or 1025).

5.1.13 Saving graphs as binary data

This option, which you invoke by right-clicking on the graph you want to save and selecting *Save graph/Graph* from the popup menu, allows you to save its data (in single-precision floating point format). A standard *Save as* dialog box will appear and you can specify the name of the file in which the data will be saved. The file extension `.fb2` will be appended to the file name you specify. The saved data can later be reloaded using *Load/Graph* from the popup menu.

5.2 Wavelet transforms

A wavelet transform of a displayed graph can be calculated by performing the following steps.

1. You select *Transforms/Wavelet.*

2. In the text box labeled *Graph,* enter the number of the graph that you want to transform.

3. In the text box labeled *Levels,* enter the number of levels to use for the transform. By resting the mouse pointer on this text box, a message should appear which tells you the allowed number of levels.

4. If you want to compute an inverse transform, then check the checkbox labeled *Inverse.* Otherwise, make sure this checkbox is not checked.

5. Select the type of wavelet you want to use. The choices are Coif, Daub, Haar, and a selection of biorthogonal wavelets. If you select Coif or Daub, then you must also enter the appropriate number at the left of the wavelet type. By resting the mouse pointer on the text box next to the wavelet type, a message should appear which tells you the allowed numbers.

6. Click on the button labeled *Plot.* This will produce the wavelet transform.

5.3 Wavelet series

A wavelet series of a displayed graph can be calculated by performing the following steps.

1. Choose *Series/Wavelet* from the menu.
2. In the text box labeled *Graph,* enter the number of the graph that you want to compute a series for.
3. In the text box labeled *Levels,* enter the number of levels to use for the series. By resting the mouse pointer on this text box, a message should appear which tells you the allowed number of levels.
4. Select the type of wavelet you want to use. The choices are Coif, Daub, Haar, and a selection of biorthogonal wavelets. If you select Coif or Daub, then you must also enter the appropriate number at the left of the wavelet type. By resting the mouse pointer on the text box next to the wavelet type, a message should appear which tells you the allowed numbers.
5. For 2D wavelet series there is only one type of series, the *threshold* type. To specify the threshold(s) click on the button labeled *Edit settings.* After selecting *Edit settings*, there are two methods for defining a threshold: *Uniform* and *Multiple.* Here are descriptions of each of these two methods.

   a. *Uniform Threshold.* This method involves specifying a threshold by typing in a value in the text box labeled *Value.* You should enter an expression of the form `1/2^n` where `n` is a positive integer. For example, `1/2^2` or `1/2^3`, etc. If the checkbox labeled *Quantize* is checked, then a threshold of the form `1/2^n` will produce an `n+1` bit or `n` bit quantization of all wavelet coefficients (transform values) depending on whether the checkbox labeled *Sign bit* is checked. Note: The tool tip[1] for the text box will provide information on how many bits you are specifying for the threshold value that you have entered.

      If the checkbox labeled *Quantization* is not checked, then a value of the form `1/2^n` produces a threshold of the form `M/2^n`. Here `M` is the maximum magnitude of all wavelet coefficients. All wavelet coefficients whose magnitudes are less than this threshold are set equal to zero, but no quantization is performed on the values that survive this thresholding.

   b. *Multiple Thresholds.* This method involves specifying two thresholds by entering in the text box labeled *Value* an expression of the form `1/2^m:1/2^n` where `m` and `n` are positive integers. With this option, quantization is always performed. The first threshold should have the form `1/2^m`, which produces an `m+1` bit or `m` bit quantization of all trend values, depending on whether the checkbox labeled *Sign bit* is checked. Likewise, the second threshold should have the form `1/2^n`, which produces an `n+1` bit or `n` bit quantization of all fluctuation values, depending on whether the checkbox labeled *Sign bit* is checked. Note: The tool tip for the text box will provide information on how many bits you are specifying for the threshold values that you have entered.

6. Click on the button labeled *Plot.* This will produce the wavelet series.

5.4 Wavelet packet transform

To compute a wavelet packet transform of a signal you select *Transform/Wavelet packet.* The rest of the steps are the same as described above for wavelet transforms.

Note: FAWAV only computes the simplest type of wavelet packet transform: the wavelet packet transform where each subsignal is further transformed up to the highest level. More complex transforms are performed by more advanced software such as the ones described in Appendix C of *A Primer on Wavelets and their Scientific Applications*.

[1] The tool tip is the message that appears when a mouse pointer is rested on a text box, or other item in a window.

5.5 Wavelet packet series

To compute a wavelet packet series for a signal you select *Transform/Wavelet packet.* The rest of the steps are the same as described above for wavelet series.

Note: FAWAV only computes the simplest type of wavelet packet series: where the wavelet packet transforms for each subsignal are further transformed up to the highest level. More complex series are performed by more advanced software such as the ones described in Appendix C of *A Primer on Wavelets and their Scientific Applications*.

5.6 Fourier transforms

A Fourier transform, or power spectrum, of a signal can be calculated by performing the following 9 steps.

1. After plotting your signal data, select *Transforms/Fourier* from the menu.

2. Select one of the two data types, either *Real* or *Complex.*

3. If you selected *Real,* then enter the number of the graph that you want to transform into the text box labeled *Graph.* The values of this signal will constitute the real part of the data, the imaginary part will be assumed to be all zero values. On the other hand, if you selected *Complex,* then enter in the text boxes labeled *Real* and *Imag* the numbers of the graphs whose values you want to use for the real and imaginary parts of the data, respectively.

4. Choose which *Output type* you want. There are three output types to choose from: (a) *Real/Im,* or (b) *Power sp.* We now describe these two *Output types.*

   (a) *Real/Im* This output type produces the real and imaginary parts of the Fourier transform of the signal. For a discrete signal the Fourier transform consists of the DFT of the discrete signal.

   (c) *Power sp* This output type produces a discrete version of the power spectrum of a signal. The values of the power spectrum of a signal are the squares of the magnitudes of the complex values of the Fourier transform of the signal.

   Power spectra are used for describing the (global) energy of each frequency in a signal. At least insofar as a DFT can accurately capture the contribution of that frequency to the signal.

5. Choose which *exponent type* to use. The two exponent types are $-$ and $+$. If you select $-$ then (discrete sampled versions of) the exponentials $e^{-i2\pi ux}$ are used for computing transforms. On the other hand, if you select $+$ then (discrete sampled versions of) the exponentials $e^{+i2\pi ux}$ are used for computing transforms.

   These two types are inverse to each other: by Fourier transforming with one choice, and then Fourier transforming the result with the other exponent type, you return to the original data.

6. Click the button labeled *Plot,* this will produce the Fourier transform or power spectrum. A separate window will be created for displaying this new data.

5.6.1 Examples

Examples illustrating many of these aspects of computing Fourier transforms[2] can be found in Chapters 5 and 6 of the book

James S. Walker, *Fast Fourier Transforms, 2nd Edition*, CRC Press, 1996

which we refer to as *Fast Fourier Transforms, 2nd Edition*.

[2] Using an older command window (DOS) version of FAWAV called FAS.

5.7 Fourier series

FAWAV allows you to calculate partial sums of Fourier series for 2D data. A 2D Fourier series (period Ω in each variable) has the form

$$\sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} c_{m,n} e^{-i2\pi(mx+ny)/\Omega}$$

where $\{c_{m,n}\}$ are the Fourier coefficients.

A Fourier series partial sum can be calculated by performing the following steps.

1. In the textbox labeled *Graph* enter the number of the graph for which you want to compute a Fourier series partial sum.

2. In the box labeled *Sum type* choose which sum type you want to use, either *Spherical* or *rectangular.*

   If you select *Spherical,* then in the textbox labeled *Harmonic limit* enter the number of harmonics you want to use. The tooltip for this textbox will tell you the allowed range of harmonic limits. If you specify, say, 15 as the harmonic limit, then all Fourier series coefficients $c_{m,n}$ with indices m and n satisfying $n^2 + m^2 \leq 15^2$ will be used in forming the Fourier series partial sum.

   If you select *Rectangular,* then in the textboxes labeled *Row* and *Col* type in the number of harmonics you want to use. The tooltip for these textboxes will tell you the allowed range of harmonic limits. If you specify, say, 15 as the row harmonic limit, and 25 as the column harmonic limit, then all Fourier series coefficients $c_{m,n}$ with indices m and n satisfying $|m| \leq 15$ and $|n| \leq 25$ will be used in forming the Fourier series partial sum.

3. Select the *Series type.* There are two types, *Unfiltered* and *Filtered.* If you select *Unfiltered,* then the Fourier coefficients are used without any modification when forming the partial sum. If you select *Filtered,* then a list box is activated and you can select a filter. Most of these filters are functions that modify the Fourier coefficients, by multiplying them by different functions. The 6 filters that are available are the following:

   Blackman
   Cesaro
   Diffraction
   Diffusion
   Hamming
   Hanning

   We will discuss each of these filters below.

4. Click on the button labeled *Plot.* This will produce the Fourier series partial sum.

5.7.1 Filtering of Fourier series

In this section we describe each of the 6 filtering methods for Fourier series.

1. *Blackman* If a spherical Fourier series is being used, then the Blackman filter modifies a Fourier series partial sum by multiplying coefficients $c_{m,n}$ with indices m and n by the following convergence factor

   $$.42 + .5\cos((m^2+n^2)^{1/2}\pi/N) + .08\cos(2(m^2+n^2)^{1/2}\pi/N)$$

   where N is the harmonic limit in the partial sum.

   If a rectangular Fourier series is being used, then the Blackman filter modifies a Fourier series partial sum by multiplying coefficients $c_{m,n}$ with indices m and n by the following convergence factor

   $$[.42 + .5\cos(m\pi/M) + .08\cos(2m\pi/M)] \times [.42 + .5\cos(n\pi/N) + .08\cos(2n\pi/N)]$$

   where M is the row harmonic limit and N is the column harmonic limit.

2. *Cesaro* If a spherical Fourier series is being used, then the Cesaro filter modifies a Fourier series partial sum by multiplying coefficients with indices m and n by the following convergence factor

$$1 - [(m^2 + n^2)^{1/2}]/N$$

where N is the harmonic limit in the partial sum.

If a rectangular Fourier series is being used, then the Cesaro filter modifies a Fourier series partial sum by multiplying coefficients with indices m and n by the following convergence factor

$$[1 - |m|/M][1 - |n|/N]$$

where M is the row harmonic limit and N is the column harmonic limit.

3. *Diffraction* The Diffraction filter modifies a Fourier series in two ways: First, by multiplying coefficients with indices m and n by the factor $\cos[WT(m^2+n^2)(\pi/L)^2)]$, where L is the right endpoint for each dimension of the square $[-L, L] \times [-L, L]$ over which the graph is plotted. Second, by multiplying by the factor $\sin[WT((m^2 + n^2)\pi/L)^2)]$. The result is then created by summing the squares of the values of these two modified Fourier series. You are required to supply the values of the *Wave constant*, W, and *Time constant*, T.

This filter can be used to create a diffraction pattern (for electron diffraction use the wave constant 0.578).

4. *Diffusion* The Diffusion filter modifies a Fourier series partial sum by multiplying coefficients with indices m and n by the following convergence factor

$$e^{-T(m^2+n^2)(\pi/L)^2}$$

where L is the right endpoint for each dimension of the square $[-L, L] \times [-L, L]$ over which the graph is plotted, and T is the damping constant (which you are required to input). The damping constant must be non-negative.

5. *Hamming* If a spherical Fourier series is being used, then the Hamming filter modifies a Fourier series partial sum by multiplying coefficients with indices m and n by the following convergence factor

$$.54 + .46\cos((m^2 + n^2)^{1/2}\pi/N)$$

where N is the harmonic limit in the partial sum.

If a rectangular Fourier series is being used, then the Hamming filter modifies a Fourier series partial sum by multiplying coefficients with indices m and n by the following convergence factor

$$[.54 + .46\cos(m\pi/M)][.54 + .46\cos(n\pi/N)]$$

where M is the row harmonic limit and N is the column harmonic limit.

6. *Hanning* If a spherical Fourier series is being used, then the Hamming filter modifies a Fourier series partial sum by multiplying coefficients with indices m and n by the following convergence factor

$$.5 + .5\cos((m^2 + n^2)^{1/2}\pi/N)$$

where N is the harmonic limit in the partial sum.

If a rectangular Fourier series is being used, then the Hamming filter modifies a Fourier series partial sum by multiplying coefficients with indices m and n by the following convergence factor

$$[.5 + .5\cos(m\pi/M)][.5 + .5\cos(n\pi/N)]$$

where M is the row harmonic limit and N is the column harmonic limit.

5.8 Convolutions and Correlations

FAWAV performs three types of discrete convolution operations. These are

1. convolution
2. pair-correlation
3. auto-correlation

We now describe each of these 3 types.

5.8.1 Convolution

To perform a convolution of two displayed graphs, perform the following 3 steps.

1. Enter the numbers for the displayed graph in the text boxes labeled *Graph 1* and *Graph 2.*

2. If you want to divide the values of the convolution by the area of the square over which the displayed graphs are plotted, then click on the checkbox labeled *Divide by area.* **Warning:** If the two signals are defined over different squares, then FAWAV will divide by the area of the square for the second function. This is confusing, and is avoided by restricting to convolutions of functions defined over a common square when selecting *Divide by area.*

3. Click on the button labeled *Plot.* This will produce the convolution.

5.8.2 Pair-correlation

To perform a pair-correlation (also known simply as correlation) of two displayed graphs, perform the following 4 steps.

1. Enter the numbers for the displayed graph in the text boxes labeled *Graph 1* and *Graph 2.*

2. If you want to divide the values of the convolution by the area of the square over which the displayed graphs are plotted, then click on the checkbox labeled *Divide by area.* **Warning:** If the two signals are defined over different squares, then FAWAV will divide by the area of the square for the second function. This is confusing, and is avoided by restricting to convolutions of functions defined over a common square when selecting *Divide by area.*

3. If you want to divide the values of the correlation by the energy of the *Graph 1* signal, then click on the checkbox labeled *Normalize.*

4. Click on the button labeled *Plot.*

5.8.3 Auto-correlation

To perform an autocorrelation on a displayed graph (which is the pair-correlation of a graph with itself), perform the following 4 steps.

1. Enter the number for the displayed graph in the text box labeled *Graph.*

2. If you want to divide the values of the convolution by the area of the square over which the displayed graphs are plotted, then click on the checkbox labeled *Divide by area.*

3. If you want to divide the values of the autocorrelation by the energy of the displayed signal, then click on the checkbox labeled *Normalize.*

4. Click on the button labeled *Plot.* This will produce the autocorrelation.

5.8.4 Examples

Several examples of the use of convolution and correlation are given in the two references, *A Primer on Wavelets and their Scientific Applications* and *Fast Fourier Transforms, 2nd Edition*.

5.9 Image Compression

There are three ways to perform image compression of gray-scale images. They are the following:

1. Image Compression Workshop (see section 5.9.1).
2. Saving a compressed image in the 2D-form (see Section 5.9.2 on p. 73).
3. Saving a compressed image in the Image Processor form (see Section 7.2 on p. 89).

5.9.1 Image Compression Workshop

This form appears when you select *Image compression workshop* from the *Graph* menu (see Figure 5.2). To compress an image with this method, you should first perform a wavelet (or wavelet packet) transform of the image and then save this transform in the \Compress\Data subdirectory of the FAWAV program directory. Then, by selecting this procedure, you can perform a compression using the following 8-step process.

1. Enter the number of *Levels.* This number should equal the number of levels used for your transform.

2. Select either a *lossy* or *lossless* compression option. With *lossless* your transform should be of the integer-to-integer type.

3. Select a *Rate/Distortion Option.* You can either choose to obtain a target bit rate (bpp) or PSNR value. For example, in the figure, a target bit rate of 0.25 bpp is selected, which achieves a 32:1 compression of an 8-bit gray scale image.

4. Select a compression technique, either *Binary* or *Arithmetic compression.* These two options are explained in section 4.4 of *A Primer on Wavelets and their Scientific Applications*.

5. Choose a *scanning mode.* These modes were various experiments in scanning through wavelet transform values for performing ASWDR (see section 4.5 of *A Primer on Wavelets and their Scientific Applications*). The default option, which is shown, is to use the method described in *A Primer on Wavelets and their Scientific Applications*, which seems to work best on most images.

6. To compress the saved transform, press the *Encode Transform* button. A dialog box will appear from which you can select your saved transform. Information will be written in the textbox at the bottom of the form that describes the progress of the encoding, including a final statement stating the file location of the encoded data.

7. To decompress your compressed file, you press the *Decode transform* button. A dialog box will appear from which you select the file containing your compressed data. When the decoding finishes, a statement will be written in the textbox at the bottom of the form that indicates the file location containing the decompressed transform.

8. To produce an image from the decompressed transform, you return to the 2D-form from which you began this process, and select *Load/Graph* from the popup menu (activated by right-clicking on a graph area). You then load the decompressed transform from the last step and perform an inverse transform (inverse to the transform used on the original image), then perform an 8-bit quantization (select *Graph/Quantize (8-bit image)*) on the resulting image to produce the completed decompressed image.

5.9.2 Saving a compressed image

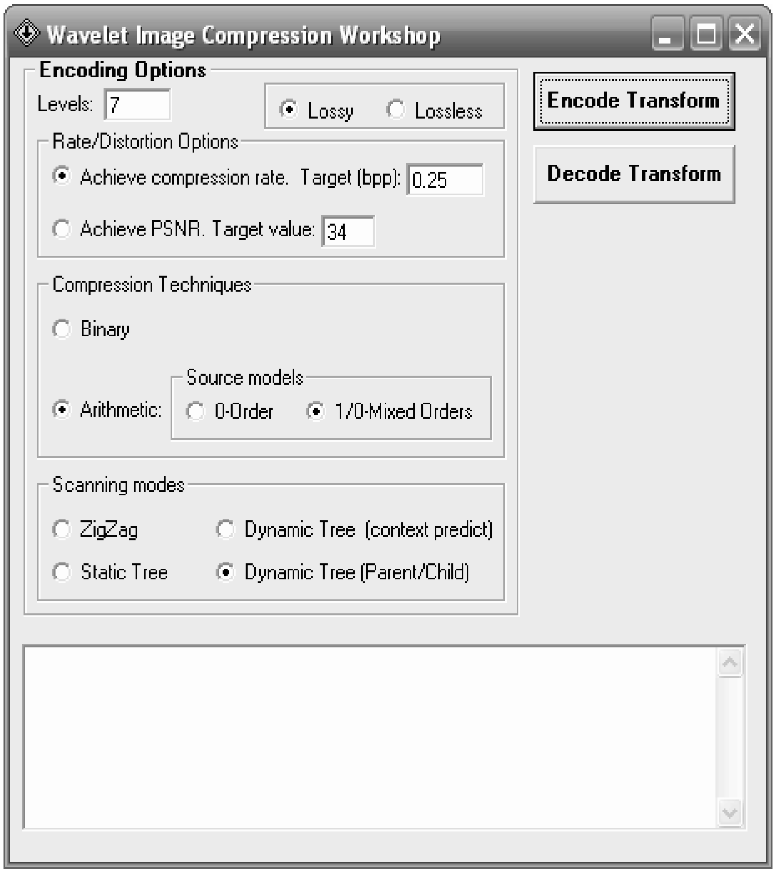
To save a compressed image in a 2D-form, you right-click on the image and select *Save/Image* from the popup menu. You then select either a `*.jpg` format (JPEG compression) or a `*.wc2` format (ASWDR wavelet-based compression). Unless you change the saving options (see below), the default value of 70% quality for JPEG and 32:1 compression ratio for ASWDR will be used.

Changing saving options

To change the options used for saving images you select *Save options (images)* from the *File* menu. The window shown in Figure 5.3 appears. The options for JPEG are mostly self-evident, the quality factor results in more fidelity in the compressed image when a higher quality factor is chosen (there is no simple relation between quality factor and compressed file size).[3] The options for TIFF images are described elsewhere, we shall not discuss them here as TIFF is rarely used for lossy compression.

We now turn to the options for wavelet compression (`.wc2` files). You can specify the number of *levels* to use for the wavelet transform. You can also choose which wavelet to use for transforming. The choices are Daub 9/7 (for lossy compression), and DD (2+2,2) Integer and DD 5/3 Integer (for lossless compression). If you choose to obtain a

[3] A better program for performing JPEG compression is IMAGE ANALYZER, which allows you to specify the compression ratio.



Wavelet Image Compression Workshop

Encoding Options

Levels: 7

● Lossy ○ Lossless

Rate/Distortion Options

● Achieve compression rate. Target (bpp): 0.25

○ Achieve PSNR. Target value: 34

Compression Techniques

○ Binary

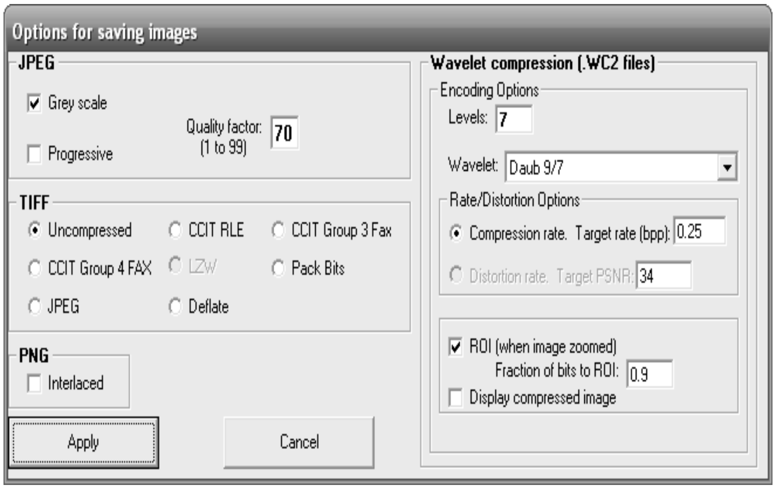
● Arithmetic: Source models ○ 0-Order ● 1/0-Mixed Orders

Scanning modes

○ ZigZag ○ Dynamic Tree (context predict)

○ Static Tree ● Dynamic Tree (Parent/Child)

Figure 5.2
The window that opens when *Image Compression Workshop* is selected.



Options for saving images

JPEG

☑ Grey scale

□ Progressive

Quality factor: (1 to 99) 70

TIFF

● Uncompressed ○ CCIT RLE ○ CCIT Group 3 Fax

○ CCIT Group 4 FAX ○ LZW ○ Pack Bits

○ JPEG ○ Deflate

PNG

□ Interlaced

Wavelet compression (.WC2 files)

Encoding Options

Levels: 7

Wavelet: Daub 9/7

Rate/Distortion Options

● Compression rate. Target rate (bpp): 0.25

○ Distortion rate. Target PSNR: 34

☑ ROI (when image zoomed)

Fraction of bits to ROI: 0.9

□ Display compressed image

Figure 5.3
The window that opens for changing image saving options.

target bit-rate, then you must specify the bits per point (bpp) value. For example, a bpp value of 0.25 specifies 32:1 compression of an 8 bpp gray-scale image. If the selection *ROI* is checked, then you may not select the *Distortion rate* option. By unchecking the *ROI* selection, you can then choose to achieve a target PSNR rate for your compression. If you have zoomed on a portion of an image and you have checked the *ROI* selection, then a specified fraction of the target bit budget (a target bit rate is then your only option) will be allocated to encoding the zoomed area of the image (whichever image you select to save in compressed form). Once you have specified all this information for creating a `.wc2` compressed file, you then click the *Apply* button and return to the 2D-form. At that point, you then right-click on the image you want to save and choose the `wc2` format option in the *Save* dialog box that appears.

5.10 Image Denoising

There are three options for image denoising. The first option is *Denoise (wavelet)* which is used for removing random Gaussian noise from an image (the most common type of noise). The second option is *Denoise (median)* which is used for removing isolated overflow and underflow noise (where the image acquisition device has registered 0 or 255 at various isolated positions in the image), sometimes referred to as “salt and pepper noise.” The third option is *Denoise (adaptive median)*, is a second method for removing salt and pepper noise.

Denoise (wavelet)

This procedure implements the various wavelet-based denoising methods described in the following papers:

1. D. Donoho and I. Johnstone, “Ideal spatial adaptation via wavelet shrinkage.” *Biometrika,* **81,** 425-455, Dec. 1994.
2. R. R. Coifman and D. Donoho, “Translation-invariant denoising.” Technical Report 475, Dept. of Statistics, Stanford University. May 1995.
3. J.S. Walker and Y.-J. Chen, “Image denoising using tree-based wavelet subband correlations and shrinkage.” *Optical Engineering,* **39,** 2900–2908, Dec. 2000.

We first outline the various steps involved in denoising an image, and then describe how to perform the methods described in these three papers.

To denoise an image, you need to go through the following five steps in the denoising dialog box (see Figure 5.4).

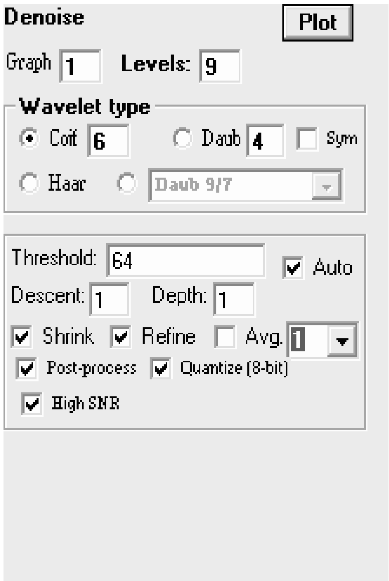
1. You first specify the graph number of the image that you want to denoise, then specify a wavelet to use and a number of levels for the wavelet transform.

2. Specify a *threshold.* This will be done automatically by FAWAV if the option *Auto* is checked. If *Auto* is not checked, then you need to specify the threshold yourself.

3. Check or uncheck the various options labeled *Shrink., Refine,* and *Avg.* If you check the *Avg.* option, then select one of the options 1, 1D, 2, 2D, . . . , 8D. These are organized with the fastest performance first (choice 1) to the slowest performance last (choice 8D), but with increasing fidelity in denoising. Generally, 2D or 4D gives acceptable performance, without unduly increasing computation time (although 8D does not take too long for images smaller than 512 by 512, and produces better denoising). [The notation D indicates that shifts are performed in both $+$ and $-$ directions for performing the averaging. Generally it is best to choose a D option for averaging.]

4. Specify the value of the *Descent* parameter, and specify the value of the *Depth* parameter. The significance of these parameters are described in the third paper cited above.

5. Click on the button labeled *Plot* and the image will be denoised. Note: After the denoised image is displayed, you should round its values to integers. You can do this by selecting *Graph/Quantize (8-bit image)* from the menu. (This quantization step is unnecessary, however, if the item *Quantize* has been checked in the denoising dialog box.)



Graph 1 Levels: 9

Wavelet type

● Coif 6 ○ Daub 4 □ Sym

○ Haar ○ Daub 9/7

Threshold: 64 ☑ Auto

Descent: 1 Depth: 1

☑ Shrink ☑ Refine □ Avg. 1

☑ Post-process ☑ Quantize (8-bit)

☑ High SNR

Figure 5.4
The dialog box that opens for denoising an image.

After the denoised image is displayed, you can use the menu choice *Analysis/Norm difference* in order to calculate error measures between the original and denoised image.

As noted above, this denoising procedure implements the techniques described in papers 1–3 listed above. For instance, to implement the *VisuShrink* method described in paper 1, you should check the *Auto* option of choosing a threshold, check the *Shrink* and *Refine* options and uncheck the *Avg.* option. Finally, set the *Depth* and *Descent* values to 0. Usually a 3-level or 4-level transform works best with this method.

To implement (partially) the cycle-spin threshold method of paper 2, you should check the *Auto* option of choosing a threshold, uncheck the *Shrink* option, check the *Refine* option, and check the *Avg.* option. Finally, set the *Depth* and

Descent values to 0. In order to conserve memory, and save time, FAWAV does not implement the full cycle-spin method (which averages over all possible shifts of the image), but only implements a finite number of shifts (index shifts from $x = -K$ to K, $y = -K$ to K, where $K = 1$, 2, 4, or 8; specified by the options 1D, 2D, 4D, 8D, respectively). You specify a value of K when you check the *Avg.* option. As described above, the values of $K = 2$ or $K = 4$ generally provide acceptable results. Experiments with various images have shown that increasing K to 8 provides only a slight improvement in denoising.

To implement the denoising method described in paper 3 (called the TAWS algorithm) you should check the *Auto* method of choosing a threshold, check the *Shrink* method, check the *Refine* option, and specify non-zero values for the *Depth* and *Descent* parameters. Generally, checking the *Avg.* option produces better denoising (at the cost of increased time). But, the TAWS method allows for the *Avg.* option to be either checked or unchecked. Generally, a value of 1, 2, or 3 for the *Depth* parameter. and an equal or greater value for the *Descent* parameter (no greater than the level number, however), produces the best results.

Denoise (median)

This procedure denoises an image using a 3 by 3 median filter. It is used when there is “salt and pepper noise” (also called “impulsive noise”) present. You simply specify which image to denoise and click on the button labeled Denoise.

Denoise (adaptive median)

This procedure denoises an image using a 3 by 3 adaptive median filter. The pixel value centered on the 3 by 3 region is only replaced by the median if its value differs from the median by more than $\pm 10\%$. It is used when there is “salt and pepper noise” (also called “impulsive noise”) present. It usually performs a better denoising than median denoising. To denoise, you simply specify which image to denoise and click on the button labeled Denoise.

5.11 Menu Items for 2D-Form

There are 9 menu items on a 2D-form:

File Edit View Graph Series Transforms Convolve Analysis Help

We now describe the contents of each of these 9 items.

5.11.1 File

The *File* menu contains 12 items, which we now examine.

1. *New 1Dim* Selecting this item will open a new 1D-form (see Chapter 2 beginning on p. 11). Each 1D-form is numbered when it is opened. You can access a list of all opened windows by selecting *View* from the menu.

2. *New 2Dim* Selecting this item will open a new 2D-form. Each 2D-form is numbered when it is opened. You can access a list of all opened windows by selecting *View* from the menu.

3. *New Audio Editor* Selecting this item will open a new Audio Editor window (see Chapter 6 beginning on p. 85). Each Audio Editor is numbered when it is opened. You can access a list of all opened windows by selecting *View* from the menu.

4. *New Image Processor* Selecting this item will open a new Image Processor window (see Chapter 7 beginning on p. 89). Each Image Processor is numbered when it is opened. You can access a list of all opened windows by selecting *View* from the menu.

5. *Open form* You select this item in order to open a previously saved form (either a 1D-form, a 2D-form, a Spectrogram form, or a Scalogram form). A standard WINDOWS File Open dialog box opens up, from which you can select a saved form to load. The file extension `.fw1` is used for 1D forms, the file extension `.fw2` is used for 2D forms, the file extension `.spg` is used for spectrograms, and the file extension `.scg` is used for scalograms.

6. *Open image file* This item can be used for loading an image file into the 2D-form. It is recommended, however, that you use the method described in Section 5.1.10 on p. 65 instead. Using that method will help you to become familiar with the Popup menu technique for 2D-forms (see Section 5.12 beginning on p. 82 for a description of this Popup Menu).

7. *Open data file* This item can be used for loading an `.fb2` data file into the 2D-form. It is recommended, however, that you use the method described in Section 5.1.12 on p. 66 instead. Using that method will help you to become familiar with the Popup menu technique for 2D-forms (see Section 5.12 beginning on p. 82 for a description of this Popup Menu).

8. *Save Form As* Select this item in order to save all of your work in the 2D form to a file with extension `.fw2`. The saved form can be reloaded later by selecting *Open form* from the *File* menu.

9. *Save Form* When you have loaded a previously saved form and then made modifications to that form, then this menu item becomes available for resaving the modified form (using the same file name).

10. *Print* This section can be used to either print an individual graph or to print the entire 2D form (as it is displayed on screen). For further discussion, please go to Section 5.12.11 on p. 84.

11. *Close* Selecting this choice will close the 2D form. *Warning:* FAWAV does not prompt you to save the 2D form if you have made modifications. If you want to save your work, be sure and save the form (using *File/Save Form*) before you close it.

5.11.2 Edit

The *Edit* menu contains 5 items, which we now examine.

1. *Points used* Choose this item in order to change the number of points that FAWAV is using for plotting graphs. The choices are

   $$64 \quad 128 \quad 256 \quad 512 \quad 1024 \quad 2048$$

   with the presently used number of points preceded by a checkmark (✓). If N points are chosen, then $N \times N$ square images are used. *Warning:* Choosing a new number of points will erase all graphs from the screen, and from memory, causing you to lose that work (unless you have previously saved it).

2. *Erase graphs* Choosing this item will remove all graphs from the screen, and from memory. *Warning:* All graphs will be erased immediately, causing you to lose that work (unless you have previously saved it).

3. *Refresh* Choosing this item will replot all graphs.

4. *Remarks* When you select this choice, then a text box opens on the right of the form, where you can enter remarks describing the work that you have performed in this form. This is especially useful for keeping a record of your work if you intend to save the 2D-form.

5. *Screen capture* This choice will create an image of the entire FAWAV window, as it appears on screen, that you can then paste into word processing documents like WORDPAD documents, or other documents in other programs that accept WINDOWS clipboard input (via a *Paste* command).

5.11.3 View

The menu item *View* contains 6 items, which we now examine.

1. *Size* This choice allows you to toggle between two sizes for displaying images: (1) *Small* where four images can be displayed at once; or (2) *Large* where a single image is displayed (and you select which image to view from option buttons on the lower right of the form).

2. *Zoom* This choice activates the *Zoom* dialog box. For more details, see Section 5.12.1 on p. 83.

3. *Clear sidebox* This choice will remove any dialog box from the right side of the form.

4. *End trace* This choice will remove the *Trace* dialog box from the right of the window and its crosshair symbol from the graph area.

5. *Arrange windows* This choice will perform automatic rearrangement of the positions of multiple windows within the main FAWAV window.

The *View* menu always has at its end a listing of all the windows that are open within the FAWAV workspace, so that you can select which window to be the active one. [The presently active one is marked with a checkmark (✓).]

5.11.4 Graph

The *Graph* menu contains 8 items, which we now describe.

1. *Plot* This item opens up the plotting dialog box on the right of the window. See Section 5.1, beginning on p. 60, for further discussion. There is also online help available by selecting *Help for Graphing* from the *Help* menu when the plotting dialog box is displayed.

2. *Denoise (Wavelet)* This procedure allows for several types of wavelet-based denoisings. See Section 5.10 on p. 74 for more details.

3. *Median denoise* This procedure denoises a signal using a 3 by 3 median filter. See Section 5.10 on p. 76 for more details.

4. *Adaptive median denoise* This procedure denoises a signal using a 3 by 3 adaptive median filter. See Section 5.10 on p. 76 for more details.

5. *Quantize (8-bit)* This procedure rounds graph values to integers between 0 and 255. It can be used for producing 8-bit gray-scale images.

6. *Down sample* This procedure creates a new graph, from the graph that you specify, by using only its even indexed values. A new window appears that contains the graph of this down sampled graph. (Note: this procedure is not available if the graph consists of only 64 by 64 points).

7. *Interpolate* This procedure creates a new graph, from the graph that you specify, by inserting a new values in between each of its values. There are two ways that this can be done.

   (a) *Haar interpolation* Haar interpolation inserts a new value at each midpoint by assigning the immediately preceding function value. Thus producing a function which repeats values two at a time. This is done for both row and columns. (Haar interpolation treats a given image as a trend subimage and sets all fluctuation values to 0, and then performs a 1-level inverse Haar transform.)

   (b) *D-D 4,2 Interpolate* This method of interpolation belongs to a type known as *Deslauriers-Dubuc.* It is described in more detail in *A Wavelet Tour of Signal Processing*, by Stephane Mallat (Academic Press, 1999). With this interpolation, the new function equals the original at the old data points, and has new values at midpoints obtained via an inverse wavelet transform (producing a better approximation to a continuous function than with Haar interpolation).

5.11.5 Series

The series that are available with this menu item fall into two categories: (1) Fourier; (2) Wavelet, Wavelet Packet. The first category is described in detail in Section 5.7, beginning on p. 69. And the two choices in the second category are described in detail in Section 5.3 on p. 67 and Section 5.5 on p. 68.

5.11.6 Transforms

The transforms that are available with this menu fall into two categories: (1) Fourier (see Section 5.6 on p. 68); (2) Wavelet (see Section 5.2 on p. 66) and Wavelet Packet (see Section 5.4 on p. 67).

5.11.7 Convolve

The three choices on the *Convolve* menu are described in detail in Section 5.8 on p. 71.

5.11.8 Analysis

The *Analysis* menu contains 14 items. We describe these items in the 14 subsections that follow.

Statistics

This choice produces a printout on the right of the window of various statistics summarizing the data consisting of a signal's values. For example, if the graph of `(2x + 4)(3y-5)+20` is plotted over the square $[-0.5, 0.5] \times [-0.5, 0.5]$ using 512 points for each dimension, and this procedure is invoked, then we get the following printout:

```
Maximum =  9.5
Minimum = -12.5
Mean =  0
Variance =  20.6647958792355
Std. Dev. =  4.54585480182061
Energy =  5438313.00194664
Samples used =  263169
Theoretical Dim =  141270.692331029
```

Most of these values are elementary statistical measures. The `Energy` is the sum of the squares of the values of the signal. The `Theoretical Dim` is a quantity related to the ability to do a wavelet compression of the signal [for more details, see the book, *Wavelet Analysis: From Theory to Software* by M.V. Wickerhauser (AK Peters, 1994)].

Norm difference

This choice allows you to compute various measures of the global difference between functions, typically used to express the amount of error of one graph as approximant to another graph. To compute the norm difference between two displayed graphs, perform the following 4 steps.

1. In the text boxes labeled *Graph1* and *Graph2* enter the numbers for the two graphs whose norm difference you want to compute.

2. Select the *Norm type.* There are two kinds: *Sup norm* and *Power norm.* More details about these choices will be given below.

3. Select whether the norm is to be *Absolute* or *Relative.*

4. Click on the button labeled *Compute.* The norm difference will be computed and the result will be given in the text box below the *Compute* button.

Note: Norm differences are computed over the region shown on screen. This may represent a subregion of the full region over which the two graphs are defined, if you have changed the display region.

In addition to the norm differences listed above, you can also compute either *SNR* or *PSNR* or *RMS* error measures. We describe each of these below.

Definitions of various norm differences

We now describe the various norm differences that can be computed. First, assume that *Absolute* norm differences are computed. If a *Sup* norm is chosen, then the maximum absolute value of all differences between the specified signals' values is found, over the region over which the graph is displayed, and this maximum is the Sup norm difference. If a *Power* norm is selected, then the power to be used must be specified in the text box labeled *Power.* The power, p, must be a number greater than or equal to 1. The power norm difference is computed by forming a Riemann sum approximation to the integral of the power p of the absolute value of the differences between the two graphs, over the region for which the graph is displayed, and then taking the p^{th} root of the result. Note: If you checked the box labeled *Normalized* then, before the p^{th} root is taken, the approximate integral is divided by the area of the region on screen.

If *Relative* norm differences are computed, then the absolute norm differences described above are divided by the norm of the graph specified as *Graph1*. That is, the norm difference is relative to the norm of Graph 1. (We discuss norms in the next subsection.)

SNR If you choose *SNR* then the following error measure is computed

$$(\mathrm{SNR}) = 10\log_{10}\left[\frac{\displaystyle\sum_{i,j}(f_{i,j}-g_{i,j})^2}{\displaystyle\sum_{i,j}f_{i,j}^2}\right]$$

where f is the graph you have specified as *Graph 1* and g is the graph you have specified as *Graph 2*, and the sums are over all pixels determined by the region displayed for *Graph 1* (this allows you to compare zooms of the two graphs, provided you perform the identical zooms).

PSNR If you choose *PSNR* then the following error measure is computed

$$(\mathrm{PSNR}) = 10\log_{10}\left[\frac{255^2}{\displaystyle\frac{1}{P}\sum_{i,j}(f_{i,j}-g_{i,j})^2}\right]$$

where f is the graph you have specified as *Graph 1* and g is the graph you have specified as *Graph 2*, and the sums are over all pixels determined by the region displayed for *Graph 1* (this allows you to compare zooms of the two graphs, provided you perform the identical zooms). The constant P stands for the number of pixels in the specified region.

RMS If you choose *RMS* then the following error measure is computed

$$(\mathrm{RMS}) = \sqrt{\frac{1}{P}\sum_{i,j}(f_{i,j}-g_{i,j})^2}$$

where f is the graph you have specified as *Graph 1* and g is the graph you have specified as *Graph 2*, and the sums are over all pixels determined by the region displayed for *Graph 1* (this allows you to compare zooms of the two graphs, provided you perform the identical zooms). The constant P stands for the number of pixels in the specified region.

Norm

A *Norm* of a graph is a numerical measure of its magnitude (we discuss this further below). To compute the norm of a displayed graph, perform the following 4 steps.

1. In the text box labeled *Graph,* enter the number for the graph whose norm you want to compute.
2. Select the *Norm type.* There are two kinds: *Sup* norm and *Power* norm. More details about these choices will be given below.
3. Select whether the norm is to be *Absolute* or *Relative.*
4. Click on the button labeled *Compute.* The norm will be computed and the result will be given in the text box below the *Compute* button.

Note: Norms are computed over the region shown on screen. This may represent a subregion of the full region over which the graph is defined, if you have changed the display region.

Definitions of various norms

We now describe the various norms that can be computed. First, assume that *Absolute* norms are computed. If a *Sup* norm is chosen, then the maximum absolute value of all the specified graph's values is found over the region for which the graph is displayed, and this maximum is the *Sup* norm. If a *Power* norm is selected, then the power to be used must be specified in the text box labeled *Power.* The power, p, must be a number greater than or equal to 1. The power norm is computed by forming a Riemann sum approximation to the integral of the power p of the absolute value of the graph, over the region for which the graph is displayed, and then taking the p^{th} root of the result. Note: If you checked the box labeled *Normalized* then, before the p^{th} root is taken, the approximate integral is divided by the area of the region on screen.

Trace

This procedure can be used for reading off the values of the displayed graphs. As the function values are displayed on the right, there are *crosshairs* drawn on the graphs on the left to indicate where these function values lie on each graph.

There are a couple ways to change the position of the crosshairs.

1. You can click on the screen at a point on a graph whose coordinates you desire. If you click with the left mouse button, then the crosshairs will move to the position that you have clicked and the values will update on the right side of the screen. If you click with the right mouse button, then you will also have the option of zooming in on the point you have selected.
2. You can also use the button bar on the bottom left of the window, which has buttons labeled with the following symbols:

   ```
   <<     <     >     >>     L     C     R
   ```

   To see how these buttons work it is best to just try them (a succinct description of their properties can also be found by simply resting the mouse on them and waiting for a short message to appear).

Calculator

To perform a calculation with the calculator, proceed as follows.

1. Enter a formula for calculation in the text box at the top. Valid formulas are ones that would be accepted by most graphing calculators. For example, these two formulas

   ```
   cos(pi/8)

   sqr(2a/b) \a = 32\b = pi
   ```

are valid formulas. Notice that in the last formula, we used command strings to substitute the values of `32` in place of `a` and `pi` in place of `b` in the formula. (More details on the functions available can be found in Section 5.1 beginning on p. 60.)

2. Click on the button labeled *Calculate* in order to perform the calculation.

Integral

To compute an integral of a displayed graph, perform the following 2 steps.

1. In the text box labeled *Graph,* type in the number of the graph for which you want to compute an integral.
2. Click on the button labeled *Compute.* This will result in the computation of the integral, over the region for which the graph is displayed. The result, which is a trapezoidal rule approximation of the exact integral, is displayed in the text box below the *Compute* button.

Scalar product

To compute a scalar product of two displayed graphs, perform the following steps.

1. In the text boxes labeled *Graph1* and *Graph2,* enter the numbers of the graphs for which you want to compute a scalar product.
2. Click on the button labeled *Compute.* This will result in the computation of the scalar product integral over the region shown on screen. The result, which is a trapezoidal rule approximation of the exact integral, is displayed in the text box below the *Compute* button.

Filtering

Filtering is performed (typically this is done to a Fourier transform of some signal) using the following 3 steps.

1. Select the type of *Data* to be filtered. There are two types: *Real* or *Complex.* If you select *Real* then you must enter the number of the graph for the real data in the text box labeled *Graph.* If you select *Complex* then you must enter the numbers of the graphs for the real and imaginary parts of the complex data in the text boxes labeled *Real* and *Imag.*
2. Select the type of *Filter.* There are two types: *Real* or *Complex.* If you select *Real* then you must enter the number of the graph for the real filter in the text box labeled *Graph.* If you select *Complex* then you must type in the numbers of the graphs for the real and imaginary parts of the complex filter in the text boxes labeled *Real* and *Imag.*
3. To perform the filtering click on the button labeled *Plot.* This will multiply the data function values by the filter function values to produce the filtered function values.

5.11.9 Help

This item typically has two choices. There is a general choice, *Help for FAWAV (2D),* which you choose to get the general help for all of the features of the 2D-form. When a dialog box is open on the right side of the screen, there will be a second choice specifically designed to access the desired topic in the help file (a form of interactive help).

5.12 The Popup menu for 2D Forms

If you right-click on the graph area on the left of a 2D-form, then a popup menu will appear with 14 items. We now discuss each of these items.

5.12.1 Zoom

This choice will zoom-in to a region centered on the coordinates of the tip of the mouse pointer (at the time that you right-clicked on the graph area).

5.12.2 Restore full image

This choice will zoom-out to the full image that you clicked on.

5.12.3 View large size or View small graphs

This choice allows you to view either the large version of the graph you clicked on, or view small versions of all the graphs.

5.12.4 Histogram

To compute a histogram of a displayed graph's values, perform the following steps.

1. In the text box labeled *Graph,* type the number of the graph for which you want to compute a histogram.

2. From the list labeled *Quantization,* select the number of levels (divisions of the range of values) to be used for computing the histogram.

3. If zero levels are to be included—this is known as uniform scalar quantization—then check the box labeled *Include zero values.* If this box is not checked, then the method is known as dead-zone quantization.

   *I*f a sign bit is to be included (i.e., both positive and negative values are quantized), then check the box labeled *Include sign bit.* If this box is not checked, then only absolute values are quantized.

4. To perform the histogram calculation, click on the button labeled *Plot.*

Note: The histogram is performed relative to the largest magnitude of the graph's values (over the region displayed) and using only displayed values (so if you have zoomed in then just the region where you have displayed values is used for the histogram). The abscissa values of the histogram are the graph maxima divided by 2 raised to the number of bits used. For example, if the histogram is a 6-bit histogram, with sign-bit, then it will be graphed over $[-32, 32]$ so the values ± 32 correspond to $\pm M$ where M is the maximum absolute value for the graph.

5.12.5 Clip

This procedure is used to clip a portion of a graph, thereby creating a smaller size graph (using less points). To clip a portion of a graph, isolate that portion on screen (by zooming) and then right-click on the graph area and select *Clip graph.*

5.12.6 Copy

This procedure is used to copy the data from the graph that you right-clicked on. You can then use the *Paste* command to copy it into another 2D-form.

5.12.7 Paste

After a graph has been copied (as described in the previous item), then this choice is used for pasting it (graphing it) into a 2D-window. The graph can be pasted as many times as you wish as long as another graph is not copied in the meantime.

5.12.8 Delete

This choice is used to remove the graph that you right-clicked on. **Warning:** Once the graph has been removed, it cannot be restored. All of its data is lost, unless you have saved it.

5.12.9 Load

This choice can be used for loading a signal. There are two choices: (1) *Graph* (see Section 5.1.12 on p. 66); (2) *Image* (see Section 5.1.10 on p. 65).

5.12.10 Save graph

This choice can be used for saving a signal in one of two formats: (1) You select *Graph* to save a signal in binary `.fb2` format (for loading using the *Load/Graph* method); (2) You select *Image* to save a signal in one of the image formats (for loading using the *Load/Image* method).

5.12.11 Print graphs

You can print graphs directly to paper output, or print graphs to a file (for later use in a document preparation system, such as LaTeX). We now describe each of these methods.

Printing graphs directly onto paper

To print the graphs on screen directly onto paper, you perform the following 6 steps.

1. From the list of choices, choose a *method of printing,* such as *Small, top, left.* These choices determine the size of the printout and where it is positioned on the page.
2. If you want the right-side labels (along the vertical) to be printed, then check the box labeled *Right labels.*
3. If you want the lower labels (along the horizontal) to be printed, then check the box labeled *Lower labels.*
4. If you want the caption for the graph to be printed, then check the box labeled *Caption.*
5. Click on one of the printout buttons, either *Print* or *Print and Eject.* If you click on *Print* then the graph will print, but will not eject from the printer. This allows for several different graphs to be printed on the same page. If you want to eject a printed page, then click on the button labeled *Eject.* If you want to print the graph and eject the page, then click on the button labeled *Print and Eject.*

Printing graphs to a file

To print graphs on screen to a file, you perform the following 2 steps.

1. Click on the button labeled *Save Printout to File.*
2. From the dialog box that appears, specify a file name for saving the printout data. Suppose this file name is `fname`, then the data for printing graphs is saved in a file: `fname.bmp`

For further information on how to include this image file in the LaTeX document preparation system, see the article, *LaTeX Graphics with* FAWAV, which is in the `\LaTeX` subdirectory of the FAWAV directory (as both a LaTeX file and a PostScript file).

5.12.12 Display cursor coordinates

You select this choice in order to provide a readout of coordinates in the graph area. Its default setting is *on,* and it is listed with a checkmark ✓. When you select it that setting changes to *off* and the checkmark ✓ is removed. The readout is given of the x- and y-indices, and the image value, located at the tip of the mouse arrow.

Chapter 6

Audio Editor

This form is used for simple processing of sound files: preparing clips of them for analysis by the 1D-form. When it is selected the form shown in Figure 6.1 loads. You can then perform the following tasks:

1. Loading a sound file
2. Clipping a portion of a file
3. Analyzing a sound clip
4. Recording sounds

which we now describe.

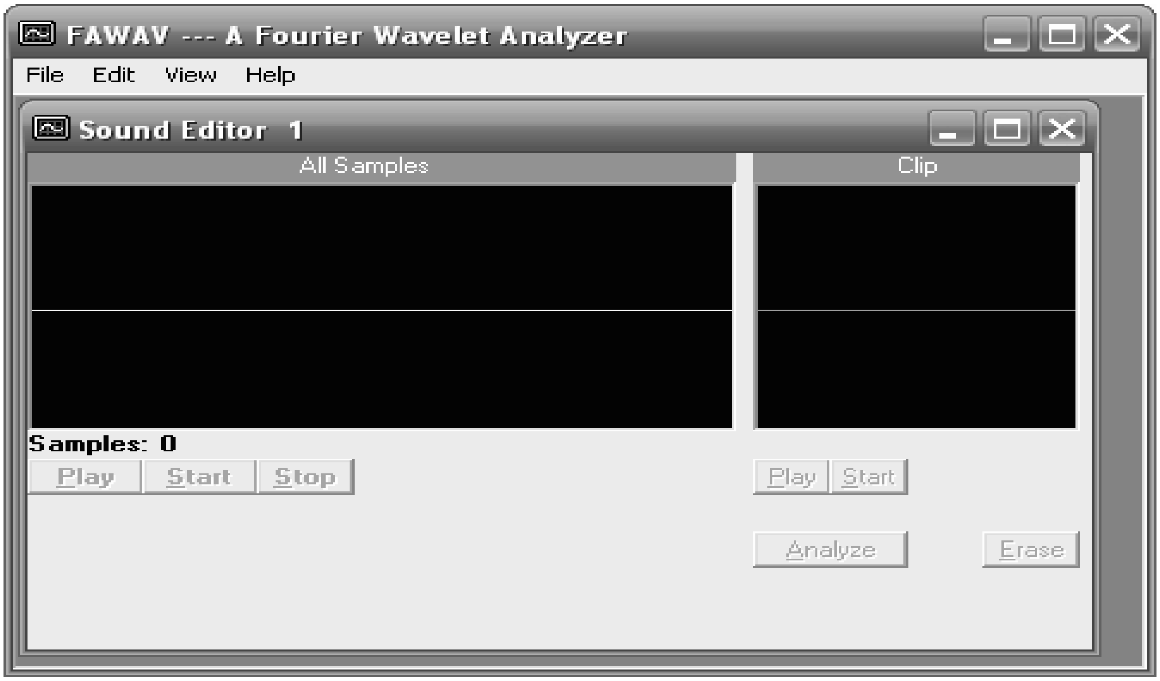


Figure 6.1
The window that opens when *New Audio editor* is selected.

6.1 Loading a sound file

To load a `.wav` file (WINDOWS sound file), choose *File/Open* from the menu. The present version of *Audio Editor* does not handle stereo sound. So make sure you load a mono sound file.

If you are unsure about the nature of the file, load the file with the *Sound Recorder* program, found in *Programs/Accessories/Multimedia* on the *Start* button's menu. You can then select the *Properties* item on the *File* menu of *Sound Recorder.* If the file is classified as mono in its list of properties, then you can safely load it into *Audio Editor.*

You can also use the free sound editor, *Audacity*, to convert a selected stereo track to a mono track, and then save that mono track for loading into *Sound Editor.*

6.2 Clipping a portion of a file

To clip a portion of a sound file for 1D analysis, perform the following 3 steps:

1. Left-click on the sound graph region, on the left side of the *Sound Editor* window, at the place where you want the sound clip to begin. A green line will appear to mark this point on the sound signal.

2. Left-click on the sound graph region, at a place to the right of the line marked in step 1, to mark the endpoint of the sound clip. A red line will appear to mark this point. Note: Because the 1D analysis procedures in FAWAV require a power of two for the number of points, the end line may not appear at precisely the point you have clicked.

3. Right-click on the sound graph region. From the popup menu that appears, select *Clip.* The portion of the sound signal you have marked off will then be graphed in the sound clip box to the right of the *Audio Editor* window. You can then perform analysis on the sound clip by clicking on the *Analysis* button.

6.3 Analyzing a sound clip

Once a sound clipping is displayed in the clip window on the right side of the *Sound Editor* window, then you can perform analysis on this clip by clicking on the Analyze button. The sound clip is then displayed in a 1D-form window, and you can then perform all of the various analysis procedures (wavelet transforms, scalograms, etc.) that are available in the 1D-form window.

6.4 Recording sounds

The present version of *Sound Editor* does not have a procedure for recording sounds. Fortunately, however, there is a good program for recording sound files on your computer. It is called *Sound Recorder*, and it can be found in *Programs/Accessories/Multimedia* from the *Start* button's menu. You can also use the free audio editor, AUDACITY, to do the recording.

You can use *Sound Recorder* to record mono sound files that you can then load into *Audio Editor.* Do not record stereo sound files for loading into *Audio Editor.* Audio Editor only works with mono sound files.

6.5 Menu Items for Audio Editor

There are 4 menu items on the *Audio Editor* form:

File Edit View Help

We now describe the contents of each of these 4 items.

6.5.1 File

The *File* menu contains 9 items, which we now examine.

1. *New 1Dim* Selecting this item will open a new 1D-form (see Chapter 2 beginning on p. 11). Each 1D-form is numbered when it is opened. You can access a list of all opened windows by selecting *View* from the menu.

2. *New 2Dim* Selecting this item will open a new 2D-form (see Chapter 5 beginning on p. 59). Each 2D-form is numbered when it is opened. You can access a list of all opened windows by selecting *View* from the menu.

3. *New Audio Editor* Selecting this item will open a new Audio Editor window. Each Audio Editor is numbered when it is opened. You can access a list of all opened windows by selecting *View* from the menu.

4. *New Image Processor* Selecting this item will open a new Image Processor window (see Chapter 7 beginning on p. 89). Each Image Processor is numbered when it is opened. You can access a list of all opened windows by selecting *View* from the menu.

5. *Open audio file* Selecting this item will allow you to load a `.wav` file. If a `.wav` file is already loaded, then this item is not available (to make it available you would need to select *Clear editor* from the *Edit* menu.

6. *Open form* You select this item in order to open a previously saved form (either a 2D-form, a 2D-form, a Spectrogram form, or a Scalogram form). A standard WINDOWS *File Open* dialog box opens up, from which you can select a saved form to load. The file extension `.fw1` is used for 1D forms, the file extension `.fw2` is used for 2D forms, the file extension `.spg` is used for spectrograms, and the file extension `.scg` is used for scalograms.

7. *Save as* This item allows you to save the audio file presently loaded into the *Sound Editor.* You save it as a `.wav` file.

8. *Save* This item allows you to save the audio file to the file specified by the *Save as* procedure just described.

9. *Close* This item closes the *Sound Editor.* **Warning:** you are not prompted for saving work.

6.5.2 Edit

This item has the following 2 choices.

1. *Erase clip lines* If you select this item, then any clip lines that you have marked on the sound file will be erased.

2. *Clear editor* If you select this item, then the sound file that is presently loaded in the *Audio Editor* will be removed and you can load a different sound file. *Note:* You are not prompted for saving the loaded file.

6.5.3 View

This item contains 2 options. If you select *Arrange windows* then you can perform automatic rearrangement of the positions of multiple windows within the main FAWAV window. The *View* menu always has at its end a listing of all the windows that are open within the FAWAV workspace, so that you can select which window to be the active one. [The presently active one is marked with a checkmark (✓).]

6.5.4 Help

This choice loads a WINDOWS help file for the *Audio Editor* form.

6.6 The Popup menu for the Audio Editor

If you right-click on the graph area on the left of the *Sound Editor* form, then a popup menu will appear with 5 items. We now discuss each of these items.

1. *Move clip region* If a clip region (see Section 6.2 on p, 86) has been specified, then selecting this item will move the left side of the clip region to the horizontal location marked by the tip of the mouse arrow at the point where you right-clicked on the graph area.

2. *Clip* Choosing this item will select a region on the sound file for analysis. (See Section 6.2 on p, 86 for further discussion.)

3. *Cut* Choosing this item will cut out the specified clipping region from the sound file. The cut out region will then be available for pasting at the location specified by the left cursor (colored green) within the present *Audio Editor* form. *Note:* to paste the cut portion into a different sound file, you should select *Clear editor* from the *Edit* menu and then load the new audio file into the present *Sound Editor* form.

4. *Copy* Choosing this item will copy the specified clipping region from the sound file. The copied region will then be available for pasting at the location specified by the left cursor (colored green) within the present *Audio Editor* form. *Note:* to paste the copied portion into a different sound file, you should select *Clear editor* from the *Edit* menu and then load the new audio file into the present *Sound Editor* form.

5. *Paste* If you have cut or copied a portion of a sound file, then choosing this item will insert that portion into the sound file beginning at the location specified by the left cursor (colored green), the sound will be inserted without overwriting any portion of the sound file.

Chapter 7

Image Processor

This form is used for simple processing of image files, such as compression and denoising. When it is selected the form shown in Figure 7.1 loads. You can then perform the following tasks:

1. Loading an Image
2. Compressing an Image
3. Decompressing an Image
4. Denoising an Image
5. Measuring error
6. Saving a processed image

which we shall now describe.

7.1 Loading an Image

To load an image, you perform the following 3 steps:

1. Select *File* from the menu.
2. Select *Get image* from the *File* menu.
3. Use the standard WINDOWS *File Open* dialog box to select an image file.

Image Processor works with various image formats. These must be 8-bit gray-scale images.

7.2 Compressing an Image

Image Processor implements the ASWDR compression algorithm. For details about this algorithm, consult the following references:

1. J.S. Walker, Lossy image codec based on adaptively scanned wavelet difference reduction. *Optical Engineering,* **39,** 1891–1897, July 2000.
2. J.S. Walker and T.Q. Nguyen, Adaptive scanning methods for wavelet difference reduction in lossy image compression. *Proceedings ICIP 2000,* Sept. 2000.

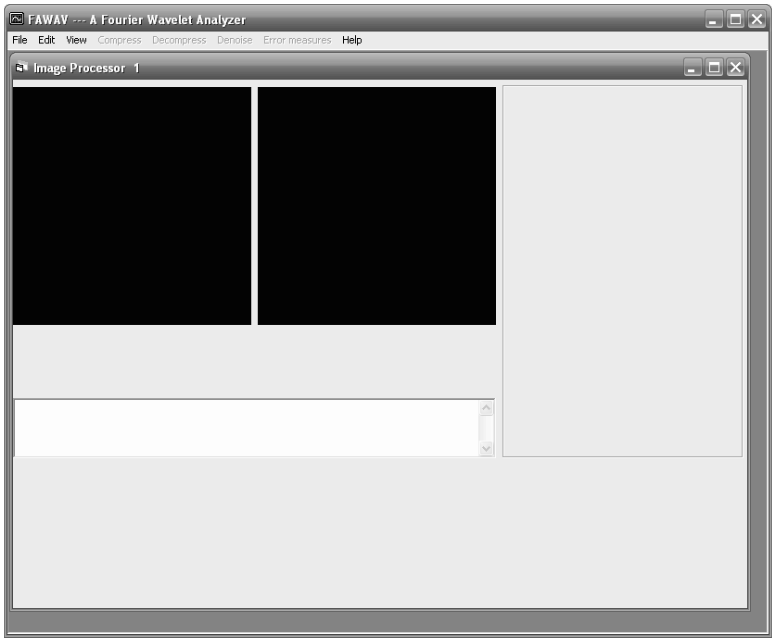


Figure 7.1
The window that opens when *New Image processor* is selected.

3. J.S. Walker and T.Q. Nguyen, Wavelet-based image compression. Chapter 6 in *Handbook of Transforms and Data Compression*, CRC Press, Boca Raton, FL, 2000.

These materials are available from the *Publications* link of the following webpage:

`http://www.uwec.edu/walkerjs/` (7.1)

To compress an image you perform the following steps:

1. Load the image (see Section 7.1 on p. 89 for more details).
2. Specify a wavelet. The choices are (1) DD 5/3 Integer, (2) DD 2+2,2 Integer, (3) Daub 9/7. The first two choices are integer-to-integer systems which allow for lossless compression, and the third is the most popular system for lossy compression. The first two systems are described in the following reference:

   A.R. Calderbank, I. Daubechies, W. Sweldens, B.-L. Yeo, Wavelet transforms that map integers to integers. *Applied and Computational Harmonic Analysis,* **5,** 332–369, 1998.

   and the third system is described in

   A. Cohen, I. Daubechies, J.-C. Feauveau, Biorthogonal bases of compactly supported wavelets. *Comm. Pure Appl. Math.,* **45,** 485–560, 1992.

3. Specify a bit rate (bits per pixel, bpp). Since the original image is 8 bpp, the default choice of 0.25 bpp produces a 32:1 compression. *Note:* To produce lossless compression, with either the 5/3 or 2+2,2 wavelet systems, you should specify 9 bpp.
4. (Optional) If you have selected a region of interest (see Section 7.2.1 on p. 91), then you should specify a percentage of bits to ROI. This percentage, which is a decimal between 0 and 1, specifies the fraction of the Total Bit Budget:

$$\text{(Total Bit Budget)} = \text{(Rows)} \cdot \text{(Columns)} \cdot \text{(8 bpp)}$$

to allocate to the ROI. If p is the Percentage of bits to ROI, then

$$\text{(Total Bit Budget)} \cdot (1 - p)$$

bits are first allocated to compressing the whole image. After this part of the Total Bit Budget is exhausted, then the remainder of bits are allocated to the region of interest.

5. Once these preliminary steps are carried out, you then click on the *Go* button. A dialog box will appear for you to specify the name of the compressed file. Since this is a demonstration program, it supplies information—such as a display of the transformed image and how many symbols are being encoded—as the compression is performed.

7.2.1 Region of Interest

To specify a region of interest, which will be more accurately compressed than the whole image, you click on the image with the right mouse button and (while holding the right mouse button down) draw a rectangle on the screen. When you lift up the right mouse button, the rectangle is displayed, and this rectangle marks off the region of interest.

If you are performing lossy compression, then the region of interest will be preserved with greater fidelity than the rest of the image. On the other hand, if you are performing lossless compression, then the region of interest will be exactly preserved (no errors) while the rest of the image will be compressed in a lossy fashion (how lossy depends on the percentage of bits to ROI value).

Note: To turn off the region of interest, you click on the image with the left mouse button.

7.3 Decompressing an Image

To decompress an image, you select *Decompress* from the menu (this selection is done automatically when the compression process finishes). You then press the *Go* button and select the file you wish to decompress.

If you specify a bpp rate that is less than the bpp rate used for compressing the file, then the file will be decompressed using this new bpp rate. This is called embedding. The compressed file has embedded within it all lower resolution images obtainable from lower bpps. This embedded property holds for both lossy and lossless compressions.

7.4 Denoising an Image

Image processor has four different procedures for image denoising.

TAWS This routine implements the TAWS algorithm as described in the following two references:

1. J.S. Walker and Y.-J. Chen, Image denoising using tree-adapted wavelet subband correlations and shrinkage. *Optical Engineering,* **39,** 2900–2908, Dec. 2000.
2. J.S. Walker, Tree-adapted wavelet shrinkage. *Advances in Imaging and Electron Physics,* 2002.

These materials can be obtained from the *Publications* link of the webpage listed in (7.1)

TAWS is used for denoising images corrupted by random (Gaussian) noise, the most common type of noise. You can choose from options that allow for shrinkage or no shrinkage, as well as averaging or no averaging. Best performance (although it takes longer time) is obtained with averaging.

SureShrink The second routine is the famous method of SureShrink. It is used for denoising images corrupted by random (Gaussian) noise, the most common type of noise.

Median The third method is 3 by 3 median denoising. It is used to denoise images that are corrupted by impulse noise (also called “salt and pepper” noise).

Adaptive median The fourth method denoises an image using a 3 by 3 adaptive median filter. The pixel value centered on the 3 by 3 region is only replaced by the median if its value differs from the median by more than $\pm 10\%$. It is used when there is “salt and pepper noise” (also called “impulsive noise”) present. It usually performs a more sharply resolved denoising than median denoising.

7.5 Measuring error

To measure the error between the original and the decompressed image, you select *Error measures* from the menu. There are three possible error measures to use: (1) Sup, (2) PSNR, (3) RMS. These error measures are computed as follows.

Sup The Sup (supremum) error is the largest magnitude error between the two image values at all pixels. The Sup error is particularly useful for checking on lossless decompression.

PSNR The PSNR (peak signal to noise ratio) is computed by the formula

$$(\mathrm{PSNR}) = 10 \log_{10} \left(\frac{255^2}{(\mathrm{MSE})} \right)$$

where (MSE) is the *Mean Square Error.* The Mean Square Error is the sum of the squares of the differences between image values at all pixels divided by the total number of pixels. PSNR is used most often in assessing the fidelity of lossy compression (its logarithmic character is related to human visual sensitivity).

RMS The RMS (root mean square) error is the square root of the Mean Square Error. The RMS error can be interpreted in terms of grey-scale units.

Note: If a region of interest has been selected, then each of the above error measures can be applied only to the region of interest. This is done by checking the box labeled *Restrict to Region of Interest.* On the other hand, if that box is not checked, then these error measures are applied to the whole images.

7.6 Saving a processed image

To save a processed image (which appears on the right), you select *File* and then select *Save Reconstructed Image.* A dialog box opens and you can save the processed image as a `.bmp` file.

7.7 Menu Items for Image Processor

There are 8 menu items on the *Image Processor* form:

File Edit View Compress Decompress Denoise Error measures Help

We now describe the contents of each of these 8 items.

7.7.1 File

The *File* menu contains 6 items, which we now examine.

1. *New 1Dim* Selecting this item will open a new 1D-form (see Chapter 2 beginning on p. 11). Each 1D-form is numbered when it is opened. You can access a list of all opened windows by selecting *View* from the menu.

2. *New 2Dim* Selecting this item will open a new 2D-form (see Chapter 5 beginning on p. 59). Each 2D-form is numbered when it is opened. You can access a list of all opened windows by selecting *View* from the menu.

3. *New Audio Editor* Selecting this item will open a new Audio Editor window (see Chapter 6 beginning on p. 85). Each Audio Editor is numbered when it is opened. You can access a list of all opened windows by selecting *View* from the menu.

4. *New Image Processor* Selecting this item will open a new *Image Processor* window. Each *Image Processor* form is numbered when it is opened. You can access a list of all opened windows by selecting *View* from the menu.

5. *Get Image* This procedure is used for loading a new image for processing. See Section 7.1 on p. 89 for more details.

6. *Save Reconstructed Image* This procedure is used for saving a processed image. See Section 7.6 on p. 92 for more details.

7.7.2 Edit

This item contains two procedures: (1) Copy left image, and (2) Copy right image. You choose one of these procedures to copy an image (either the left or right image on the *Image Processor* form) to the WINDOWS clipboard. From the clipboard you can paste the copied image into other WINDOWS programs.

7.7.3 View

This item contains 2 options. If you select *Arrange windows* then you can perform automatic rearrangement of the positions of multiple windows within the main FAWAV window. The *View* menu always has at its end a listing of all the windows that are open within the FAWAV workspace, so that you can select which window to be the active one. [The presently active one is marked with a checkmark (✓).]

7.7.4 Compress

You choose this item in order to compress an image that you have loaded. See Section 7.2 on p. 89 for more details.

7.7.5 Decompress

You choose this item in order to decompress an image that you have saved in compressed form. See Section 7.3 on p. 91 for more details.

7.7.6 Denoise

You choose this item in order to perform various kinds of denoising of an image that you have loaded. See Section 7.4 on p. 91 for more details.

7.7.7 Error measures

You choose this item to calculate various error measures between the image on the left of the form and the image on the right of the form. See Section 7.5 on p. 92 for more details.

7.7.8 Help

This choice loads a WINDOWS help file for the *Image Processor* form.