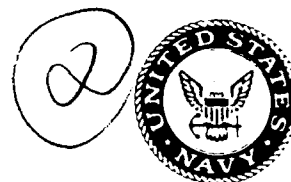


Naval Research Laboratory

Washington, DC 20375-5000



AD-A245 461



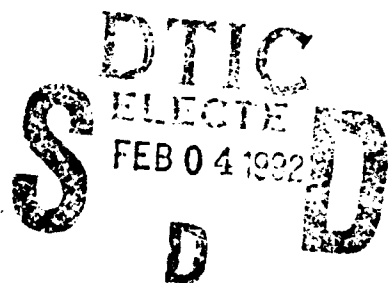
NRL Memorandum Report 6923

A Computer Program for the Calculation of Daubechies Wavelets

DAVID M. DRUMHELLER

*Acoustic Systems Branch
Acoustics Division*

January 22, 1992



92-02460



REPORT DOCUMENTATION PAGE			Form Approved OMB No 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE 1992 January 22	3. REPORT TYPE AND DATES COVERED NRL Report May 1990-October 1990		
4. TITLE AND SUBTITLE A computer Program for the Calculation of Daubechies Wavelets		5. FUNDING NUMBERS PR-RJ35B01 PE-62435N WU-DN480-045		
6. AUTHOR(S) Drumheller, David M.		8. PERFORMING ORGANIZATION REPORT NUMBER NRL Memorandum Report 6923		
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Research Laboratory Washington, DC 20375-5000		10. SPONSORING / MONITORING AGENCY REPORT NUMBER		
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Office of Naval Technology Code 234 800 N. Quincy Street Arlington, VA 22217		11. SUPPLEMENTARY NOTES		
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited		12b. DISTRIBUTION CODE		
13. ABSTRACT (Maximum 200 words) Recently, it has been found that the set of all square summable functions can be represented as a countable sum of time local functions called wavelets. A particular class of functions, Daubechies wavelets, are truly time local. They have compact support on the real line. This report presents and documents a FORTRAN program that calculates Daubechies wavelets.				
14. SUBJECT TERMS Wavelet Orthogonal Basis			15. NUMBER OF PAGES 26	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT UL	

CONTENTS

1. Introduction	1
2. Basic Method of Calculating Wavelets	3
3. Running the Program	3
4. References	7
APPENDIX A - FORTRAN Source Code for Program WAVE	9
APPENDIX B - Sequences for Daubechies wavelets	17

Accession For	
NTIS CRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution/	
Availability Codes	
Dist	Availability
A-1	



A COMPUTER PROGRAM FOR THE CALCULATION OF DAUBECHIES WAVELETS

1. Introduction

Recently, it has been found that there exist other sets of functions that can serve as an orthonormal basis for the set of all square summable functions, $L^2(R)$ [1-6]. In particular, it has been shown that for all $g(t) \in L^2(R)$, and for a certain prescribed function $\psi(t)$,

$$g(t) \stackrel{\text{a.e.}}{=} \sum_{j,k \in \mathbb{Z} \times \mathbb{Z}} g_{j,k} \sqrt{2^j} \psi(2^j(t - 2^{-j}k)), \quad (1)$$

where

$$g_{j,k} = \sqrt{2^j} \int_{-\infty}^{\infty} g(t) \psi^*(2^j(t - 2^{-j}k)) dt, \quad (2)$$

and $\mathbb{Z}$ denotes the integers. Here, the basis set is $\{\sqrt{2^j} \psi(2^j(t - 2^{-j}k))\}_{j,k \in \mathbb{Z} \times \mathbb{Z}}$ and the function $\psi(t)$ is called the *wavelet*. Furthermore, the function $\psi(t)$ is significant only over a small (compact) portion of the real line. Therefore, it is recognized that this representation has a sense of 'time locality.'

For $\psi(t)$ to be admissible as a basis function, its Fourier transform must obey certain properties. In particular, for an admissible $\psi(t)$ it can be shown that

$$\int_{-\infty}^{\infty} \psi(t) e^{-j2\pi ft} dt = \Psi(f) = K\left(\frac{f}{2}\right) \Theta\left(\frac{f}{2}\right), \quad (3)$$

where

$$K(f) = e^{-j2\pi f} H^*(f + 1/2), \quad (4)$$

$$H(f) = \sum_{k=-\infty}^{\infty} h(k) e^{-j2\pi kf}, \quad (5)$$

$$\Theta(f) = \sum_{p=1}^{\infty} H(2^{-p}f), \quad (6)$$

and $\{h(k)\}_{k \in \mathbb{Z}}$ is a sequence such that the following properties hold:

- (i) $|H(0)| = 1$,
- (ii) $h(k) \sim O(k^2)$ as $k \rightarrow \infty$,
- (iii) $|H(f)|^2 + |H(f + 1/2)|^2 = 1$.

(iv) $|H(f)| \neq 0$ for $f \in [0, 1/2)$.

Note that the inverse Fourier transform of $\Theta(f)$ is called the *scaling function*, i.e.,

$$\theta(t) = \int_{-\infty}^{\infty} \Theta(f) e^{j2\pi t} df. \quad (7)$$

The difference between the wavelet and its associated scaling function is rooted in the difference between spanning different subspaces that compose the space $L^2(R)$. It can be shown that one can define a series of subspaces V_j for $j \in Z$ such that $\bigcup_{j \in Z} V_j$ is dense in $L^2(R)$, $V_j \subset V_{j+1}$, $g(t) \in V_j$ if and only if $g(2t) \in V_{j+1}$ for all $j \in Z$, and $g(t) \in V_j$ if and only if $g(t - 2^{-j}k) \in V_j$ for all $j, k \in Z$. The significance of the scaling function is that the set $\{\sqrt{2^j}\theta(2^j(t - k))\}_{k \in Z}$ spans the subspace V_j . On the other hand, one can show that there exists a subspace O_j composed of functions that are orthogonal to those composing V_j such that

$$O_j \oplus V_j = V_{j+1}, \quad (8)$$

where $\oplus$ denotes the Cartesian product. Thus, one can show that

$$\bigcup_{j \in Z} O_j = L^2(R). \quad (9)$$

The significance of the wavelet is that the set $\{\sqrt{2^j}\psi(2^j(t - 2^{-j}k))\}_{k \in Z}$ spans the subspace O_j . More generally, the set $\{\sqrt{2^j}\psi(2^j(t - 2^{-j}k))\}_{j, k \in Z \times Z}$ spans $L^2(R)$.

The results presented above suggests that one can, to some degree, control the shape of the wavelet in the time domain according to how one chooses the sequence $\{h(k)\}_{k \in Z}$. One desirable property is to have a wavelet with compact support in the time domain, i.e., it is time limited in that it is nonzero only over a given interval. Such a wavelet gives a true sense of time locality. A set of orthogonal wavelets with compact support was discovered by Daubechies [2]. They are parameterized by an integer n , are real valued, and are denoted as $\psi_n(t)$ for $n \geq 2$. In fact,

$$\text{supp } \psi_n \subset [(1 - n), n]. \quad (10)$$

These wavelets are derived by choosing the sequence $\{h(k)\}_{k \in Z}$ so that it is of finite length. The result is the set of sequences $\{h_n(k)\}_{k=0}^{(2^n-1)}$ for $n = 1, 2, 3, \dots$. Details of the procedure for finding these sequences can be found in Daubechies' original paper [2].

Daubechies wavelets posses other desirable properties. It can be show that they are bounded, continuous functions for all n , and for $n \geq 4$, they are continuously differentiable. Furthermore, for $n \geq 4$, Daubechies wavelets have a finite spectral spectral variance, i.e., for $\Psi_n(f) \leftrightarrow \psi_n(t)$, then

$$\int_{-\infty}^{\infty} f^2 |\Psi_n(f)|^2 df < \infty. \quad (11)$$

Proof of these properties can be found in [4].

2. Basic Method of Calculating Wavelets

Equations (3) to (6) suggest how one can calculate the Fourier transform of a Daubechies wavelet and its associated scaling function. For $\Theta_n(f) \leftrightarrow \theta_n(t)$, one first calculates (approximates) the Fourier transform of the scaling function via the equation

$$\Theta_n(f) \approx \prod_{p=0}^P H_n(2^{-p}f), \quad (12)$$

where

$$H_n(f) = \sum_{k=0}^{2n-1} h_n(k) e^{-j2\pi k f}. \quad (13)$$

Once $\Theta_n(f)$ is found, one calculates the Fourier transform of the orthogonal wavelet as

$$\Psi_n(f) \approx K_n\left(\frac{f}{2}\right) \Theta_n\left(\frac{f}{2}\right), \quad (14)$$

where

$$K_n(f) = e^{-j2\pi f} H_n^*(f + 1/2). \quad (15)$$

The truncated product in Eq. (12) gives good results for $P = 20$ for low values of n ($n = 3$), to $P = 25$ for high values of n ($n = 13$). This was checked by calculating the normalized cross correlation between two Daubechies wavelets of order n , where one was derived by using $P = N$, and the other with $P = N + 1$. For $P = 25$ (or $P = 20$ for low values of n) the correlation was negligibly different from 1.

Once the Fourier transforms of the Daubechies wavelet and scaling function have been calculated, one can find the associated time domain functions by invoking the inverse Fourier transform. This can be accomplished efficiently through the use of a fast Fourier transform (FFT).

The program listed in Appendix A uses the approach outlined above to calculate Daubechies wavelets and their scaling functions, and is written in VAX extended FORTRAN. The program produces four sequential ASCII files containing sampled versions of the functions $\psi_n(t)$ and $\theta_n(t)$ and the magnitudes of their Fourier transforms. Certainly one can modify the program to produce the complex Fourier transform. An ASCII file is also produced containing the parameters input to the program by the user.

3. Running the Program

Program WAVE is designed to be run interactively, and produces sampled versions of the Daubechies wavelets and their associated scaling functions for $n = 2, 3, \dots, 15$. It is written using double precision, and can take several minutes of wall clock time to run. The

program makes use of a radix 2 FFT as an efficient way of numerically calculating a Fourier integral.

The program needs only one input file, COEFF.DAT, which is listed in Appendix B. This file contains the finite length sequences $\{h_n(k)\}_{k=0}^{2^n-1}$. All other required parameters are input from the terminal. They are:

1. n : The order of the wavelet, an integer from 2 to 15.
2. $npower$: The power of 2 yielding the FFT size. For example, $npower = 5$ implies that the FFTs used in the program are of size 2^5 .
3. f_s : Time domain sampling rate in Hertz of $\psi_n(t)$ and $\theta_n(t)$.
4. $iter$: The number of product iterations used to calculate the Fourier transform of the scaling function as given in Eq. (12), and is the equal to P in that equation. Usually a value of 20 to 25 is a good choice.

These inputs are written to the file WAVE.ECHO. Therefore, the file serves as a record of a single program run.

Among the remaining four ASCII files produced by the program are WAVELET.TIME and SCALE.TIME, which contain the sampled versions of the wavelet and scaling function. Specifically, they contain $\psi_n(i/f_s)$ and $\theta_n(i/f_s)$ where i is an integer and $-2^{npower-1} \leq i \leq 2^{npower-1} - 1$. Similarly, WAVELET.SPEC and SCALE.SPEC contain $\Psi_n(if_s)$ and $\Theta_n(if_s)$ respectively, where $0 \leq i \leq 2^{npower} - 1$. All files contain ordered samples of the functions in column form. For example, WAVELET.TIME contains two columns of numbers in the following form:

$$\begin{array}{rcl}
 1 & & \psi((-2^{npower-1})/f_s) \\
 2 & & \psi((1 - 2^{npower-1})/f_s) \\
 3 & & \psi((2 - 2^{npower-1})/f_s) \\
 4 & & \psi((3 - 2^{npower-1})/f_s) \\
 & \vdots & \\
 2^{npower} & & \psi((2^{npower-1} - 1)/f_s)
 \end{array}$$

Figures 1 through 4 show plots of the contents of the four output files generated by WAVE for $n = 5$, $npower = 9$, $f_s = 30\text{Hz.}$, and $iter = 25$.

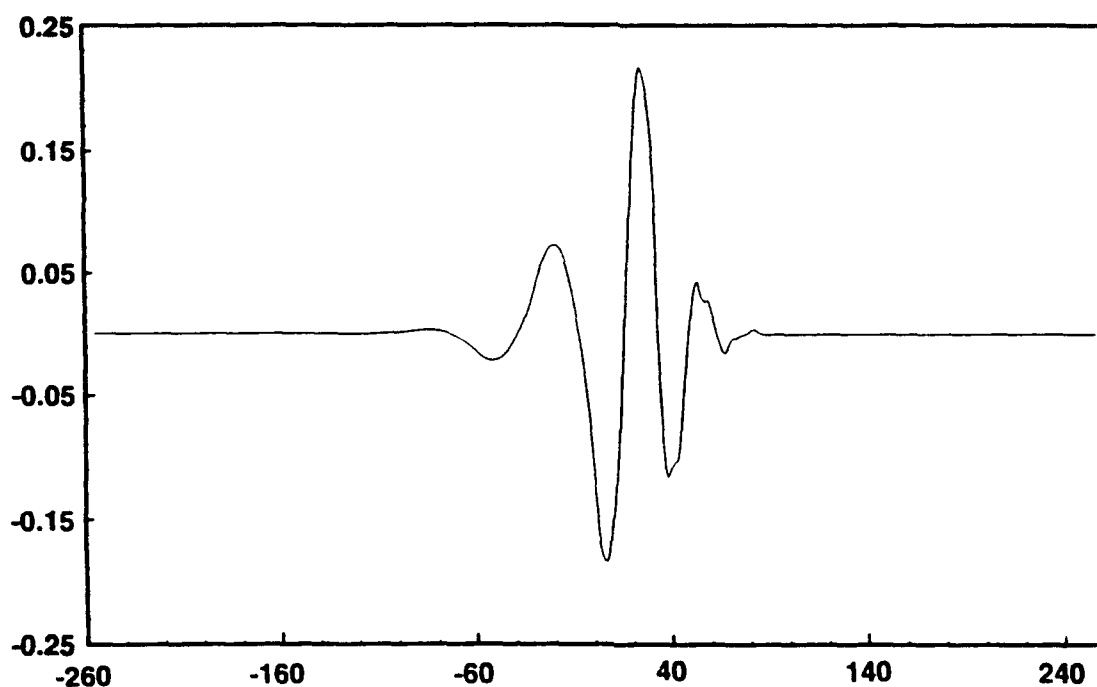


Figure 1. The contents of the file WAVELET.TIME which is a sampled version of the wavelet $\psi_5(t)$.

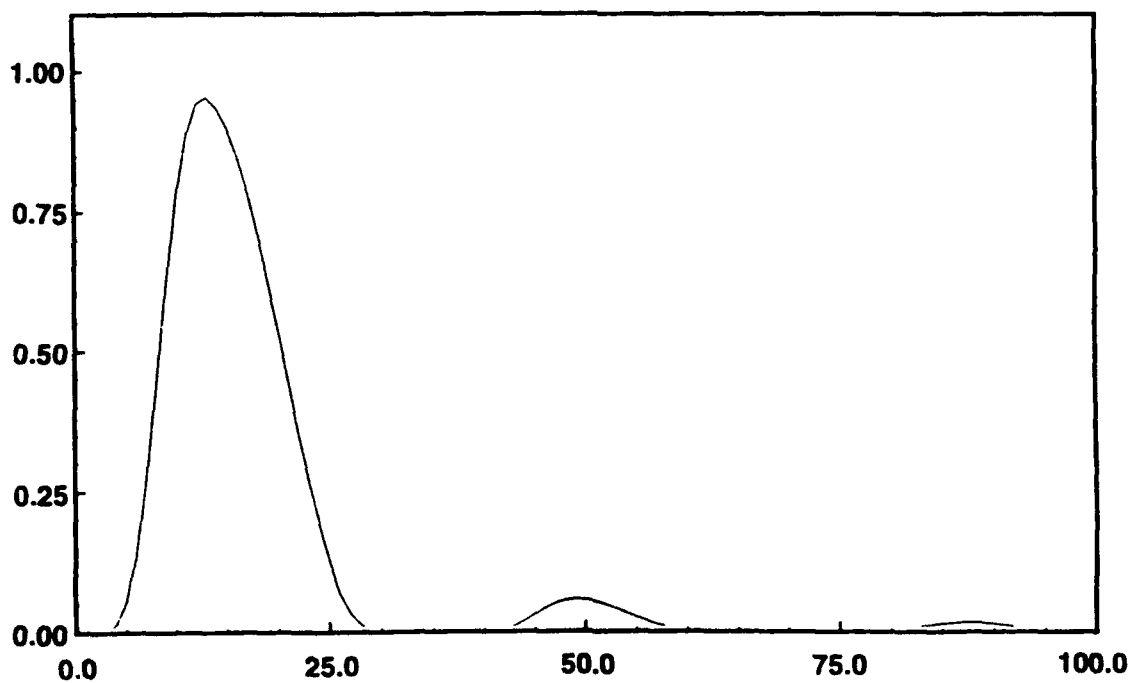


Figure 2. The contents of the file WAVELET.SPEC which is a sampled version of the wavelet spectrum $\Psi_5(f)$. Only the first 100 points are shown.

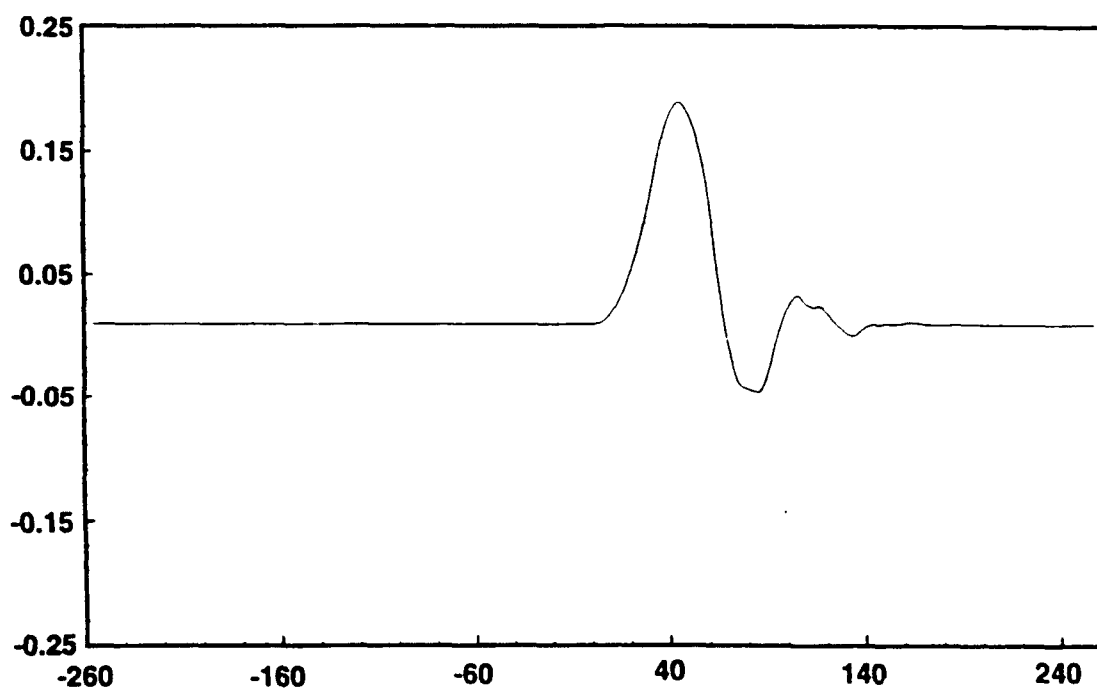


Figure 3. The contents of the file SCALE.TIME which is a sampled version of the scaling function $\theta_5(t)$.

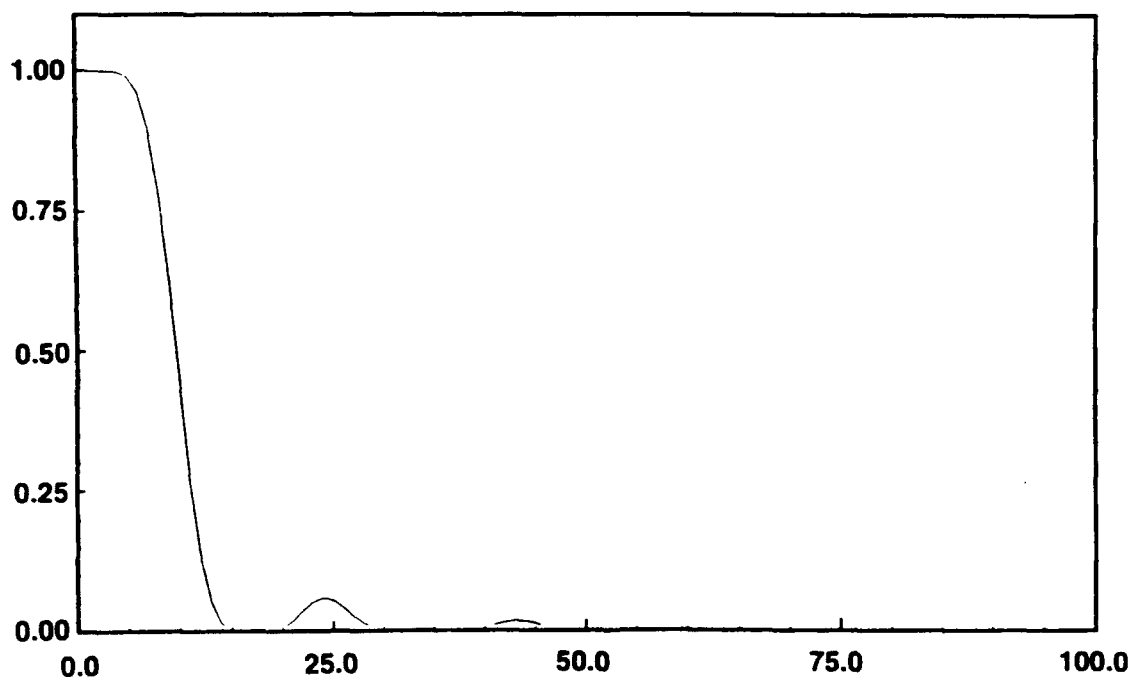


Figure 4. The contents of the file SCALE.SPEC which is a sampled version of the scaling function spectrum $\Theta_5(f)$. Only the first 100 points are shown.

4. References

1. J.M.Combes, A.Grossman, P.Tchamitchian, eds., *Wavelets: Time-Frequency Methods and Phase Space*, Proc. of the Int. Conf., Marseille, France. Dec. 14-18, 1987 (Springer-Verlag, Berlin, Germany, 1989).
2. I.Daubechies, "Orthogonal Bases of Compactly Supported Wavelets," Commun. on Pure and Applied Mathematics **XLI**, 909-996 (1988).
3. I.Daubechies, "Painless nonorthogonal expansions," J. Mathematical Physics **27**(5), 1271-1283 (1986).
4. D.M.Drumheller, "Theory and Application of the Wavelet Transform to Signal Processing," Naval Research Laboratory Report 9316, 31 July 1991.
5. A.Grossmann, J.Morlet, "Decomposition of Hardy Functions into Square Integrable Wavelets of Constant Shape," SIAM J. Mathematical Analysis **15**(4), 723-736 (1984).
6. S.G.Mallat, "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation," IEEE Trans. Pattern Analysis and Machine Intelligence **11**(7), 674-693 (1989).

This page intentionally left blank.

Appendix A

FORTRAN SOURCE CODE FOR PROGRAM WAVE

```
program wave

c This program calculates a wavelet and its spectrum. Note that
c the sign (of the exponential argument) of the Fourier transform
c conforms to the conventional definition found in the engineering
c literature.

implicit none

real*8 h(50),whigh,wlow,deltaw,deltaw2,delta,w,fhigh,ts
real*8 zreal(8192),yreal(8192)
complex*16 htrans,gtrans,x,y(8192),z(8192)
integer n,npower,iter,nsize,i,j

external htrans
external gtrans

c Open output files containing the spectrum and time series of the
c wavelet and scaling functions.

open(unit=20,file='scale.spec',status='new',
&    access='sequential')

open(unit=10,file='wavelet.spec',status='new',
&    access='sequential')

open(unit=31,file='scale.time',status='new',
&    access='sequential')

open(unit=41,file='wavelet.time',status='new',
&    access='sequential')

c Retrieve impulse response.

write(6,1000)
1000 format(/,' enter wavelet order n: ', $)
```

```

        read(5,1010) n
1010  format(bn,i5)

        call read_coeff(h,n)

        write(6,4000)
4000  format(/,' Enter sampling frequency: ', $)
        read(5,4010) fhigh
4010  format(f10.7)
        ts = 1.0/fhigh
        whigh = 2.0*3.141593*fhigh

        write(6,4020)
4020  format(/,' Enter power of 2 for number of samples: ', $)
        read(5,4030) npower
4030  format(bn,i5)

        write(6,4040)
4040  format(/,' Enter number of iterations: ', $)
        read(5,4050) iter

        if (npower .lt. 1 .or. npower .gt. 13) then
            type *, ' power of 2 out of range '
            stop
        endif

c Open output file and echo input data.

        open(unit=50,file='wave.echo',status='new',
&          access='sequential')

        write(50,1900) n,(2**npower),npower,fhigh,whigh,ts,iter
1900  format(' From program wave: ',
& //, ' wavelet order = ',i3,
& /, ' FFT size = ',i5, ' ( power of 2 = ',i3, ' )',
& /, ' Maximum frequency of calculated spectrum = ',e11.5, ' Hz',
& /, '                                     = ',e11.5, ' Rad/S',
& /, ' sampling rate = ',e11.5, ' S',
& /, ' number of iterations for calculating spectrum = ',i4)

        close(unit=50)

```

c Calculate Fourier transform.

```
    nsize = 2**npower
    wlow = 0.0
    deltaw = (whigh-wlow)/float(nsize-1)
    deltaw2 = delta/2.0
    w = wlow
    j = 1

    write(6,5610)
5610  format(/,5x,' << Calculating scaling function spectrum >>')
    do i = 0,(nsize-1)
        w = deltaw*float(i)+wlow
        call phi_spec(x,w,h,n,iter)
        y(i+1) = x*(whigh-wlow)
        write(20,5000) i,cdabs(x)
5000  format(1(2x,i6,2x,e11.5))
    enddo
```

```
    write(6,5611)
5611  format(/,5x,' << Calculating wavelet spectrum >>')
    do i = 0,(nsize-1)
        w = deltaw*float(i)+wlow
        call phi_spec(x,w/2.0,h,n,iter)
        z(i+1) = -1.0*x*gtrans(w/2.0,h,n)
        write(10,5000) i,cdabs(z(i))
    enddo
```

c Note that the use of the IFFT here is only as a numerical
c integration. Furthermore, note that you are integrating
c only over the right half of the spectrum, consequently the resulting
c impulse response is complex (the analytic signal). Note that the
c true impulse response is equal (proportional) to the real part.
c The wavelet and scaling functions are normalized (unit energy)
c prior to writing them to the output files.

```
    do i = 1,nsize
        y(i) = ((-1.0)**(i-1))*y(i)
        z(i) = ((-1.0)**(i-1))*z(i)
    enddo
```

```
    call ifft(y,npower)
    call ifft(z,npower)
```

```

do i = 1, nsize
    yreal(i) = dreal(y(i))
    zreal(i) = dreal(z(i))
enddo
call normalize(yreal, ts, nsize)
call normalize(zreal, ts, nsize)

do i = 1, nsize
    write(31, 5000) (i-1-nsize/2), yreal(i)
    write(41, 5000) (i-1-nsize/2), zreal(i)
enddo

end

subroutine read_coeff(h, n)

c This subroutine reads the impulse response from the
c data file coeff.dat.

    integer i, n
    real*8 h(1)

    open(file='coeff.dat', status='old', access='sequential',
%      unit=99)

    do while (i .ne. n)
        read(99, 1000) h(1), i
1000    format(f14.7, i5)
    enddo

    read(99, 1010) (h(i), i = 2, 2*n)
1010    format(f14.7)

    write(6, 1999)
1999    format(/, ' spectrum coefficients:')
    write(6, 2000) ((i-1), h(i), i = 1, 2*n)
2000    format(2x, 'h(', i2, ') = ', f10.7)

    close(unit=99)

    return
end

```

```

complex*16 function htrans(w,h,n)

integer n
real*8 h(1),w

htrans = (0.0,0.0)
do i = 1,(2*n)
    htrans = htrans+h(i)
%      *dcmplx(cos(float(i-1)*w),-sin(float(i-1)*w))
enddo

htrans = htrans/(1.414214,0.0)

return
end

complex*16 function gtrans(w,h,n)

integer n
real*8 h(1),w
complex*16 htrans

external htrans

gtrans = dcmlpx(cos(w),-sin(w))*conjg(htrans((w+3.141593),h,n))

return
end

subroutine phi_spec(x,w,h,n,iter)

c This subroutine calculates the spectrum of the
c wavelet (as in Daubechies' definition) at the
c frequency w.

integer n,iter,i
complex*16 x,htrans
real*8 h(1),w
external htrans

x = (1.0,0.0)

```



```

do i = 1,iter
  x = x*htrans((w/(2.0**i)),h,n)
enddo

```

```

return
end

```

```

subroutine fft(x,m)

```

c This subroutine calculates an FFT of size 2**m. It is an
c 'in-place' algorithm.

```

complex*16 x(1),u,w,t
n = 2**m
pi = 3.14159265358979
do 20 l = 1,m
  le = 2**(m+1-l)
  le1 = le/2
  u = (1.0,0.0)
  w = dcplx(cos(pi/float(le1)), -sin(pi/float(le1)))
  do 20 j = 1,le1
    do 10 i = j,n,le
      ip = i + le1
      t = x(i) + x(ip)
      x(ip) = (x(i) - x(ip))*u
10    x(i) = t
20    u = u*w
  nv2 = n/2
  nm1 = n-1
  j = 1
  do 30 i = 1,nm1
    if (i .ge. j) go to 25
    t = x(j)
    x(j) = x(i)
    x(i) = t
25    k = nv2
26    if (k .ge. j) go to 30
    j = j-k
    k = k/2
    go to 26
30    j = j+k
  return
end

```

```
subroutine ifft(x,n)
```

c This subroutine calculates the inverse FFT of the array x.

```
complex*16 x(8192)
```

```
integer n,nsiz,i
```

```
real*8 realn
```

```
nsiz = 2**n
```

```
realn = float(nsiz)
```

```
do i = 1,nsiz
```

```
  x(i) = conjg(x(i))
```

```
enddo
```

```
call fft(x,n)
```

```
do i = 1,nsiz
```

```
  x(i) = conjg(x(i)/realn)
```

```
enddo
```

```
return
```

```
end
```

```
subroutine normalize(x,ts,nsiz)
```

c This subroutine energy normalizes a real time series.

```
integer nsiz
```

```
real*8 x(nsiz),ts,sum
```

```
sum = 0.0
```

```
do i = 1,nsiz
```

```
  sum = sum+x(i)**2
```

```
enddo
```

```
sum = sqrt(sum)
```

```
do i = 1,nsiz
```

```
  x(i) = x(i)/sum
```

```
enddo
```

```
return
```

```
end
```

This page intentionally left blank.

Appendix B

SEQUENCES FOR DAUBECHIES WAVELETS

Listed below is the file COEFF.DAT used by program WAVE. It contains the sequences $\{h_n(k)\}_{k=0}^{2^n-1}$ for $n = 2, 3, \dots, 15$. It is formatted according to the statement `format(f14.7,i5)`. The first column lists the values of $h_n(k)$. The second column lists n , and marks the beginning of the sequence. For example, for $n = 3$, we have $h_3(1) = 0.332670552950$, $h_3(2) = 0.806891509311$, $h_3(3) = 0.459877502118$, $h_3(4) = -0.135011020010$, $h_3(5) = -0.085441273882$, and $h_3(6) = 0.035226291882$.

.482962913145	2
.836516303738	
.224143868042	
-.129409522551	
.332670552950	3
.806891509311	
.459877502118	
-.135011020010	
-.085441273882	
.035226291882	
.230377813309	4
.714846570553	
.630880767930	
-.027983769417	
-.187034811719	
.030841381836	
.032883011667	
-.010597401785	
.160102397974	5
.603829269797	
.724308528438	
.138428145901	
-.242294887066	
-.032244869585	
.077571493840	
-.006241490213	
-.012580751999	
.003335725285	
.111540743350	6
.494623890398	

.751133908021	
.315250351709	
- .226264693965	
- .129766867567	
.097501605587	
.027522865530	
- .031582039318	
.000553842201	
.004777257511	
- .001077301085	
.077852054085	7
.396539319482	
.729132090846	
.469782287405	
- .143906003929	
- .224036184994	
.071309219267	
.080612609151	
- .038029936935	
- .016574541631	
.012550998556	
.000429577973	
- .001801640704	
.000353713800	
.054415842243	8
.312871590914	
.675630736297	
.585354683654	
- .015829105256	
- .284015542962	
.000472484574	
.128747426620	
- .017369301002	
- .044088253931	
.013981027917	
.008746094047	
- .004870352993	
- .000391740373	
.000675449406	
- .000117476784	
.038077947364	9
.243834674613	
.604823123690	

.657288078051	
.133197385825	
-.293273783279	
-.096840783223	
.148540749338	
.030725681479	
-.067632829061	
.000250947115	
.022361662124	
-.004723204758	
-.004281503682	
.001847646883	
.000230385764	
-.000251963189	
.000039347320	
.026670057901	10
.188176800078	
.527201188932	
.688459039454	
.281172343661	
-.249846424327	
-.195946274377	
.127369340336	
.093057364604	
-.071394147166	
-.029457536822	
.033212674059	
.003606553567	
-.010733175483	
.001395351747	
.001992405295	
-.000685856695	
-.000116466855	
.000093588670	
-.000013264203	
0.018692339500	11
0.144048360129	
0.449822419238	
0.685506451221	
0.411710892303	
-.162485521339	
-.274320974144	
0.066025638763	

0.149791844607
- .046504355457
- .066445800596
0.031336714900
0.020839548328
- .015365977170
- .003339972936
0.004928945867
- .000308709907
- .000893056839
0.000249184997
0.000054438816
- .000034637754
0.000004494745
0.013114280902 12
0.109587064387
0.377449392844
0.657445006413
0.516294170295
- .044313624533
- .315809615475
- .023471399498
0.182806918672
0.005686977952
- .096186633657
0.010995853244
0.041627451082
- .012180151045
- .012829445168
0.006713258423
0.002249393038
- .002179176553
0.000006459278
0.000388621871
- .000088486615
- .000024241195
0.000012775434
- .000001528836
0.009204916897 13
0.082889405900
0.312115898739
0.611313131287
0.589096065406

0.086639694877
-.316237370186
-.126430468961
0.177816118862
0.071915527849
-.106342427892
-.026758244166
0.056034390582
0.002363616024
-.023833745174
0.003917927648
0.007254616037
-.002760408506
-.001315670455
0.000932006061
0.000049301053
-.000165090932
0.000030664729
0.000010440501
-.000004699171
0.000000521846
0.006547491642
0.063360170581
0.259953209778
0.569486757657
0.659765991407
0.253248224211
-.245883485949
-.207221475070
0.141972692112
0.144030955893
-.083519992219
-.071278880702
0.054864716315
0.027555092282
-.029754599557
-.005754062318
0.012711190182
-.000664409841
-.003831834380
0.001038385046
0.000708200880
-.000381870689

14

- .000042656957
0.000068164760
- .000010124883
- .000004370468
0.000001706613
- .000000176357
0.004129483519 15
0.041776513906
0.179077334434
0.407237141256
0.472918366318
0.110408203455
- .385470269520
- .386018759764
0.027435955232
0.154106559944
- .072306793965
- .118765098585
0.038151205753
0.052121808971
- .030285642429
- .020301136317
0.016290846193
0.004432020222
- .006931177432
0.000000520974
0.002017749394
- .000450763331
- .000363203984
0.000171774677
0.000023620852
- .000029991608
0.000003944981
0.000001876185
- .000000681954
0.000000067409