RICE UNIVERSITY

WAVEFORM ANALYSIS
USING DIGITAL TECHNIQUE

by

ELIEZER GOTTLIEB

A THESIS SUBMITTED
IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

Thesis Director's Signature:

Roger M. Goldwyn

Houston, Texas
October 1966
IN MEMORY OF MY FATHER-IN-LAW
ACKNOWLEDGEMENT

The author would like to express his deepest appreciation to Dr. Martin H. Graham and Dr. Roger M. Goldwyn for their constant inspiration, valuable ideas, excellent suggestions and patient guidance of this research. Thanks are also due to Mr. Ernie Sibert of the Rice Computer Project for help on the computational problems.

This research was supported by the Rice Computer Project under AEC Contract No. AT-(40-1)-2572.

In particular I am grateful to my wife, Haya, for her constant encouragement and patience.
# TABLE OF CONTENTS

## CHAPTER 1  SPECTRAL ANALYSIS USING RECURSIVE FILTERS

1.1 Introduction ........................................ 1  
1.2 Spectral Analysis .................................. 3  
1.3 Recursive Filter .................................... 9  
1.4 Computer Results .................................... 15  
1.5 Spectral Analysis Using Recursive Filters ........ 15  
1.6 Computer Solutions .................................. 20  
1.7 Applications of the Recursive Filters ............. 21  

## CHAPTER 2  SPECTRAL ANALYSIS VIA HETERODYNING

2.1 Introduction ......................................... 25  
2.2 Analytical Development ............................. 28  
2.3 Computational Technique ............................ 33  
2.4 Comparison of Techniques ............................ 34  

## CHAPTER 3  FOURIER TRANSFORMATIONS AND INTEGRALS  
WITH OSCILLATORY INTEGRANDS

3.1 Introduction ......................................... 38  
3.2 Taylor Series Fitting ................................ 42  
3.3 Integration .......................................... 43  
3.4 Simultaneous Integration of $S(\omega)$ and $C(\omega)$ 48  
3.5 Comments About $f(x)$ ............................... 50
3.6 Computationaı Technique
3.6.1 $K_n$ and $L_n$ Coefficients
3.6.2 Derivatives of $f(x)$
3.6.3 Discontinuity of $f(x)$
3.6.4 Program Flowchart
3.7 Examples and Computed Results
3.7.1 Example 1 - $f(x) = \cos(\omega t + \phi)$
3.7.2 Example 2 - $f(x) = e^{-t} - 1/ae^{-at}$

CHAPTER 4 NUMERICAL EVALUATION OF DERIVATIVES
4.1 Introduction
4.2 Derivatives in Terms of Difference Ratios
4.3 Numerical Evaluation of Derivatives
4.4 Computer Results

REFERENCES
Chapter 1

Spectral Analysis Using Recursive Filters

1.1. Introduction:

Spectral analysis has long been utilized as a suitable tool for studying signal and noise waveforms, and, in many cases, it has been found that the absolute magnitude squared of the Fourier transform of the waveform contains all the information required for a meaningful determination of specific problems. This squared magnitude as a function of frequency is commonly known as the energy density spectrum [2] and can actually be calculated (as a function of frequency) by integrating the squared result of a convolution of the input signal with the impulse response of a suitable filter. However, there are problems in which this energy density spectrum of a waveform does not always provide sufficient information especially when the signal to be analyzed consists of a sequence of bursts where, in addition to amplitude, a phase relationship is also important. In these cases, the real and imaginary parts of the spectrum should be computed also. The following
discussion in Chapters 1 and 2 will investigate the determination of the energy density spectrum and will develop a fast computational technique. In Chapter 3, a new rapid method for the simultaneous calculation of the real and imaginary parts of the complex spectrum will be shown. For a fundamental discussion of this subject, one should refer to [1] and [2].

Problems in which a spectral analysis of the waveforms is desirable often have the practical requirement that one be able to obtain an efficient, rapidly computable (for economic reasons), solution—perhaps in real time—with high resolution. While many such problems exist, one that served as a particular motivation here is that existing in reproducing a waveform previously recorded on a tape recorder by a frequency modulation system. In this case, the speed of the tape passing under the magnetic head is constantly fluctuating about its designed velocity due to poor stability in the driving system and also because of the elasticity of the magnetic tape. These changes in speed produce a time varying error in the reproduced frequency and therefore the signal itself. A detection of these fluctuations can tell one about the precision of the record-reproduction system and in the frequency modulation case
would enable one to correct the output waveform by using an electronic device that will produce a suitable change in the frequency-output signal characteristic. Such an electronic compensation method certainly promises more precision in the reproduced waveform as well as lower cost in that driving systems need not be as precise. To detect the changes in frequency of a reproduced signal (e.g., \( \cos(\omega t) \)), or generally speaking to determine in real time a waveform spectrum, the following analysis requirements are present:

1°. A very narrow bandwidth analyser is required, and,

2°. A real-time detector is necessary to control the frequency-signal characteristic at the output.

The techniques to be developed enable one to achieve the first requirement and to approach a solution of the second by applying a digital technique.

1.2. Spectral Analysis:

The spectral analysis to be used is based on passing the waveform \( f(t) \) under investigation through a set of linear filters with impulse responses \( h_n(t) \) and Fourier transforms \( H_n(j\omega) \). The center frequency of these filters \( (f_n=\omega_n/2\pi) \)
are the frequencies at which the spectrum is measured. The energy density of the signal at a particular frequency is obtained by evaluating the absolute magnitude squared of the Fourier transform of the output of the appropriate filter. The bandwidth of these filters has an important role in the analysis as it produces a non-zero weighting for the frequencies in the neighborhood of $\omega_n$. Therefore, the bandpass has to be very narrow to consider only $\omega_n$ but should be wide enough to give a non-zero weight to at least the frequencies between two consecutive $\omega_n$. Moreover, because of the finite memory space and computation time, the digital computer can (a) only handle a limited number of samples, (b) only process them through filters of finite length and, (c) only compute a finite number of spectral points. The last limitation requires some a priori knowledge of the signal under investigation.

A promising filter, from a computational point of view, is one with a Gaussian shape about its center frequency because of the advantage of high energy concentration within a small interval in both the time and frequency domains. (Better time and frequency concentration would be obtained with a set of prolate spheroidal functions as discussed in
[3], but the Gaussian shape will be used here for computational convenience.) The monotonic decrease of the Gaussian function about its center enables one to limit the interval of integration to any desired error of truncation by suitably setting the decay rate of the Gaussian shape. This also fixes the bandwidth of the filter.

The Fourier transform of the Gaussian shape used is

$$H(j\omega; \omega_o) = \sqrt{\frac{\pi}{\alpha}} e^{-\left(\omega - \omega_o\right)^2 / 4\alpha}$$  \hspace{1cm} (1.1)$$

where $\omega_o$ is the center frequency and $1/4\alpha$ the decay parameter. $H(j\omega; \omega_o)$ is positive real for all real $\omega$. The bandwidth of these filters (the difference between 3 db points) is $\Delta f = 2(\Delta \omega)/2\pi = .375\sqrt{\alpha}$.

The inverse Fourier transform is

$$h(t; \omega_o) = \frac{1}{2\pi} \int_{-\infty}^{\infty} H(j\omega; \omega_o) e^{-j\omega t} \, d\omega$$

$$= e^{-\alpha t^2} e^{j\omega_o t} = h_1(t; \omega_o) + jh_2(t; \omega_o)$$ \hspace{1cm} (1.2)$$

where

$$h_1(t; \omega_o) = e^{-\alpha t^2} \cos(\omega_o t),$$ \hspace{1cm} (1.3)
\[ h_2(t; \omega_0) = e^{-at^2} \sin(\omega_0 t). \]  

(1.4)

\( h_1 \) and \( h_2 \) are cosine and sine function respectfully in a Gaussian shape and this mathematical fact is one reason for selecting this filter.

When a waveform \( f(t) \), with Fourier transform \( F(j\omega) \), is passed through the filter, the output in the frequency domain is

\[ G(j\omega; \omega_0) = F(j\omega) \times H(j\omega; \omega_0) \]  

(1.5)

and in the time domain is

\[ g(t; \omega_0) = f(t) \ast h(t; \omega_0) \]  

(1.6)

where \((\ast)\) designates the convolution operation. \( h(t; \omega_0) \) is the filter output due to an impulse excitation and \( g(t; \omega_0) \) is the inverse Fourier transform of \( G(j\omega; \omega_0) \).

Dealing with finite signals that start at \( t=0 \) and are zero for all \( t<0 \) and \( t>t_F \) (1.6) becomes

\[ g(t; \omega_0) = \int_{0}^{t} f(\tau) \ h(t-\tau; \omega_0) \ d\tau \]
for $0 < t < t_f$ and

$$g(t; \omega_0) = \int_0^{t_f} f(\tau) h(t-\tau; \omega_0) d\tau$$  \hspace{1cm} (1.7)

for $t \geq t_f$.

The Fourier transform of the energy density spectrum is

$$|F(j\omega)|^2 = F(j\omega) F^*(j\omega)$$  \hspace{1cm} (1.8)

where $F^*(j\omega)$ is the complex conjugate of $F(j\omega)$. The total energy at $\omega_0$ is the integrated output of the filter with center frequency at $\omega_0$, i.e.

$$E(\omega_0) = \int_{-\infty}^{\infty} |g(t; \omega_0)|^2 dt$$  \hspace{1cm} (1.9)

and with Parseval's theorem [2], equation (1.9) can also be expressed as

$$E(\omega_0) = \frac{1}{2\pi} \int_{-\infty}^{\infty} |H(j\omega, \omega_0)|^2 |F(j\omega)|^2 d\omega$$  \hspace{1cm} (1.10)

If the filter bandwidth is relatively narrow and if $|F(j\omega)|^2$ remains fairly constant within the bandpass of
\( H(j\omega;\omega_0) \), then one may write

\[
|F(j\omega_0)|^2 = 2\pi E(\omega_0) \left[ \int_{-\infty}^{\infty} |H(j\omega,\omega_0)|^2 d\omega \right]^{-1} \quad (1.11)
\]

where again \( |F(j\omega)|^2 \) is the energy density spectrum and

\[
\int_{-\infty}^{\infty} |H(j\omega,\omega_0)|^2 d\omega = \pi \sqrt{\frac{2\pi}{\alpha}} \quad (1.12)
\]

Notice that (1.12) is independent of \( \omega_0 \).

With (1.3), (1.4), and (1.7), (1.9) yields the alternate relationship for \( E(\omega_0) \) as

\[
E(\omega_0) = \int_{-\infty}^{\infty} \left[ \int_{-\infty}^{\infty} f(\tau)h_1(t-\tau,\omega_0) d\tau \right]^2 dt
\]

\[
+ \int_{-\infty}^{\infty} \left[ \int_{-\infty}^{\infty} f(\tau)h_2(t-\tau,\omega_0) d\tau \right]^2 dt \quad (1.13)
\]

From (1.11), (1.12), and (1.13), one can fairly well approximate the energy density spectrum as

\[
|F(\omega_0)|^2 = \sqrt{\frac{2\alpha}{\pi}} E(\omega_0) \quad (1.14)
\]

where \( E(\omega_0) \) is obtained from (1.13) for example.
For a full discussion on the above, see [2].

When the signal \( f(t) \) is available at discrete points 
\( t = nT \) where \( n = 0, 1, 2, \ldots, N \) and \( T \) is the interval between 
sampling, then equations (1.14) and (1.13) can be written 
in the summation form

\[
|F(j\omega)|^2 = \frac{2^\alpha}{\pi} \sum_{n=0}^{N} \left[ \sum_{t_k} f(nT)e^{-\alpha(t_k-nT)^2} \cos(\omega_0(t_k-nT)) \right]^2 \\
+ \left[ \sum_{n=0}^{N} f(nT)e^{-\alpha(t_k-nT)^2} \sin(\omega_0(t_k-nT)) \right]^2 \quad (1.15)
\]

where \( t_k \) represent the center of the time "windows" for the 
integration in the time domain.

Equations (1.15) and (1.7) form the basis for the 
spectrum analysis developed in Chapters 2 and 3 respectively. 
This chapter will proceed with the development of a recursive 
filter technique for the analysis of the spectrum of a given 
waveform.

1.3. Recursive Filter:

The sampled data system shown by the simplified block 
diagram in Figure 1.1 has the transfer function
\[ G(s) = \frac{F_{\text{out}}(s)}{F_{\text{in}}(s)} = \frac{1-ae^{-st}}{1-2ae^{-st}e^{-2st}} \]  \tag{1.16}

where \( T \) (constant) is the interval between two consecutive samples and "a" is a constant.

Let the transformation \( z = e^{st} \) be used in (1.16) and substitute \( a = \cos(\omega T) \). Using Z transform theory [4], one can easily find that \( G(z) \) is the Z-transform of \( \cos(n\omega T) \) where \( n = 0,1,2,3,\ldots \) and \( \omega \) is the frequency of the trigonometric function. Moreover, if \( f_{\text{in}}(nT) = 1 \) for \( n = 0 \) and zero for all \( n \neq 0 \), then \( f_{\text{out}}(nT) = \cos(n\omega T) \) which are the values of a cosine wave at points \( nT \).

The following equations as seen from the block diagram present \( f_{\text{out}}(nT) \) in a recursive manner:

\[ f_{\text{out}}(nT) = f'_{\text{in}}(nT) + 2af_{\text{out}}((n-1)T) - f_{\text{out}}((n-2)T) \]

and

\[ f'_{\text{in}}(nT) = f_{\text{in}}(nT) - af_{\text{in}}((n-1)T) \] \tag{1.17}

Another property of \( f_{\text{out}}(nT) \) is a way in which one can evaluate the amplitude of the cosine wave by using only 3
Figure 1.1  BLOCK DIAGRAM OF THE COSINE RECURSIVE FILTER

Figure 1.2  BLOCK DIAGRAM OF THE SINE RECURSIVE FILTER
adjacent points. Let these adjacent points be:

\( f_{\text{out}}(nT) = A \times \cos(n\omega T) \) and \( f_{\text{out}}(nT\pm T) = A \times \cos(n\omega T \pm \omega T) \).

This reduces to

\[
f_{\text{out}}(nT\pm T) = A [\cos(n\omega T) \cos(\omega T) \pm \sin(n\omega T) \sin(\omega T)] \tag{1.19}
\]

Adding and subtracting (1.19) yield

\[
f_{\text{out}}(nT+T) + f_{\text{out}}(nT-T) = 2A \times \cos(n\omega T) \cos(\omega T) \tag{1.20}
\]

\[
- f_{\text{out}}(nT+T) + f_{\text{out}}(nT-T) = 2A \times \sin(n\omega T) \sin(\omega T) \tag{1.21}
\]

or

\[
A \times \cos(n\omega T) = \left[ f_{\text{out}}(nT+T) + f_{\text{out}}(nT-T) \right] / (2a) \tag{1.22}
\]

\[
-A \times \sin(n\omega T) = \left[ f_{\text{out}}(nT+T) - f_{\text{out}}(nT-T) \right] / (2b) \tag{1.23}
\]

where again \( a = \cos(\omega T) \) and \( b = \sin(\omega T) \).

Squaring (1.22) and (1.23) and adding one obtains

\[
A^2 = \left[ f_{\text{out}}(nT+T) + f_{\text{out}}(nT-T) \right]^2 / (2a)^2
\]

\[
+ \left[ f_{\text{out}}(nT+T) - f_{\text{out}}(nT-T) \right]^2 / (2b)^2 \tag{1.24}
\]
where \(A\) is the amplitude of the cosine wave. (Dividing (1.22) by (1.23), the phase angle can also be determined.)

By precisely similar analysis, the recursive technique for the sine wave is determined. The block diagram in Figure 1.2 has the \(Z\)-transformed transfer function

\[
G(Z) = \frac{bz^{-1}}{1-2az^{-1}+z^{-2}} \quad (1.25)
\]

which is the \(Z\)-transform of \(\sin(n\omega T) \quad n=0,1,2,3,\ldots\)

\(f_{out}(nT)\) is given by the following recursive formulae:

\[
f_{out}(nT) = f'_{in}(nT) + 2af_{out}(nT-T) - f_{out}(nT-2T) \quad (1.26)
\]

and

\[
f'_{in}(nT) = bxf_{in}(nT-T). \quad (1.27)
\]

If one excites this filter with \(f_{in}(nT)=1\) when \(n=0\) and zero for all \(n \neq 0\), using (1.26) and (1.27) the output calculated will be \(f_{out}(nT)=\sin(n\omega T)\). The amplitude in this sine recursive filter may be computed as given in equation (1.24) which can be derived following (1.19) through (1.23).

It is clear that to produce the above trigonometric
waveform at discrete points nT one needs to perform 5
additions plus one multiplication per point which saves
about 80% of machine time over a conventional method for
sine or cosine determination.

The range of frequencies that these recursive filters
can effectively span is strongly dependent on T because
a=cos(ω_oT) is limited between ±1 or 0 ≤ ωT ≤ π. The band-
pass, in this case, is from D.C. up to f=1/2T. This cer-
tainly follows from the sampling theorem [5] and is a
necessary and sufficient condition to describe uniquely the
energy density spectrum [2].

If the recursive filter is excited from t=0 with a se-
quence of samples f_in(nT), the output g(nT;ω_o) is the con-
volution

\[ g(nT;ω_o) = \sum_{k=0}^{n} f_{in}(kT)h(nT-kT;ω_o)(k=0,1,2,...,n) \]  \hspace{1cm} (1.28)

which, when multiplied by T, is the summation form of (1.7).

(A zero order hold would be employed.)

A similar recursive technique can be developed for
any function with a finite number of terms in its Z-trans-
form (i.e., e^{-anT}, e^{-anT}cos(ωnT), e^{-anT}sin(nωT), etc.).
1.4. Computer Results:

The recursive sine and cosine filters were examined for accuracy in computing the amplitude according to (1.24). The tests were performed for various numbers of iterations up to one million cycles and for several values of $a = \cos(\omega T)$ ($\pm 0.99, \pm 0.01, \pm 0.5, \pm 0.866$). The process was carried in fixed point starting with $f_{in}(0) = k$ and $f_{in}(nT) = 0$ for all $n \neq 0$. Each run was repeated 5 times with a different $k$ to enable one as well to determine the accuracy as a function of the word length. Results in form of number of decimal digits of accuracy are shown in Figures 1.3 and 1.4. The "maps" in these figures can also be used to determine the number of bits (or decimal digits) one needs to use in order to maintain any desired accuracy when utilizing the recursive filter process.

1.5. Spectrum Analysis Using Recursive Filters:

The recursive filter is a discrete frequency filter, with an impulse response $h(t; \omega_o) = \cos(\omega_o t)$ or $h(t, \omega_o) = \sin(\omega_o t)$ and the Fourier transform

$$H(j\omega; \omega_o) = \frac{1}{2}[\delta(\omega-\omega_o) + \delta(\omega+\omega_o)].$$  (1.29)
FIGURE 1.3  ACCURACY OF AMPLITUDE WHEN COS(NωT) IS
RECURSIVELY EVALUATED
ωT = 30°      N = 0, 1, 2, etc.

UNDERLINED NUMBERS DESIGNATE DIGITS OF
ACCURACY FOR THE ZONE
FIGURE 1.4  ACCURACY OF AMPLITUDE WHEN SIN (NωT) IS RECURSIVELY EVALUATED

\[ ωT = 8° \quad N = 0, 1, 2, \text{ etc.} \]

UNDERLINED NUMBERS DESIGNATE DIGITS OF ACCURACY FOR THE ZONE
This filter may be physically represented by a tuned circuit such as a capacitor and inductor connected in parallel and driven by a voltage or current source. The instantaneous energy stored in this network is

\[ e = \frac{1}{2} [c v^2 + L i^2] \]  

(1.30)

where \( v \) and \( i \) are respectively the voltage and current of the circuit. With \( i = c \frac{dv}{dt} \) and \( \omega_o^2 = 1/Lc \), (1.30) becomes

\[ e = \frac{C}{2} [v^2 + (\frac{dv}{dt})^2] \]  

(1.31)

where \( \tau = \omega_o t \) and \( d\tau = \omega_o dt \).

(1.31) is identical to (1.24) at discrete times \( nT \) with \( n=0,1,2,\ldots \) if

\[ c = \frac{2}{a^2} \]  

(1.32)

\[ \frac{f ((n+1)T) + f ((n-1)T)}{2} = v(nT) \]  

(1.33)

and
\[
\frac{f((n+1)T) - f((n-1)T)}{2\tan(\omega_0 T)} = \frac{dv}{dt} \bigg|_{T=n\omega_0 T}
\]  

\[1.34\]

(1.34) is the derivative expression at \(t=nT\) if \(\omega_0 T\) is small enough to justify the substitution \(\tan(\omega_0 T) = \omega_0 T\).  
The energy stored in the tuned circuit changes when excited by an external source and reaches a steady state after completion of the input signal which is proportional to the amplitude squared \(cv^2/2\).

The voltage \(v\) is

\[
g(t; \omega_0) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(j\omega)H(j\omega; \omega_0) e^{j\omega t} d\omega.
\]

\[1.35\]

with (1.29),

\[
2\pi g(t; \omega_0) = \frac{1}{2} [F(j\omega_0) e^{j\omega_0 t} + F(-j\omega_0) e^{-j\omega_0 t}].
\]

\[1.36\]

Since \(f(t)\) is real, \(F(-j\omega_0) = F^*(j\omega_0)\) and (1.36) becomes

\[
2\pi g(t; \omega_0) = \text{Re}\{F(j\omega_0)\} \cos(\omega_0 t) - \text{Im}\{F(j\omega_0)\} \sin(\omega_0 t)
\]

\[1.37\]

The steady state amplitude squared after the input signal has terminated is therefore
\[ 4\pi^2 |g(t; \omega_0)|^2 = |\mathcal{F}(j\omega_0)|^2 \] (1.38)

where (1.38) is again the energy density spectrum at \( \omega_0 \) and this can be determined by using (1.24) and a set of recursive filters distributed at the frequencies of interest.

The bandpass of these filters are so narrow (equation (1.29)) that the output bandwidth from performing the recursive filtering depends on the input shape and duration and can be derived from the Fourier transform of the signal under investigation. Input waveforms with Gaussian shapes about their center yield bandwidths of \( \Delta f = 0.375\sqrt{a} \) as shown in paragraph 1.2. When a cosine or sine waveform with finite length is investigated, one can expect a spectrum with a sharper bandwidth than with the Gaussian case, but the Gaussian shaped inputs are usually preferred because of the more rapid attenuation of the sidebands which is an important fact in the analysis of short signals or waveforms containing a variety of frequencies.

1.6. Computer Solutions:

The recursive technique of the previous section was investigated using as an input a finite cosine wave in a Gaussian envelope about its center. The Gaussian decay \( \alpha \)
was chosen to give the cosine amplitude a decrease of 97% at its ends. This waveform was sampled every 2 milliseconds. Inputs of .8 second and 10 seconds were examined and the energy density spectrum plots are given in Figure 1.5. The computed bandwidths match the calculated ones. The same waves, without the Gaussian envelopes, were examined and the energy density spectrum are shown in Figure 1.6.

Using the recursive filtering technique for analysis of inputs of 5000 sample lengths (f<397c/s) with a fast computer, one can expect to get about 4 spectral components per second. Equation (1.15) was also used to analyze the same waveforms (the cosine and sine samples were produced recursively) and as a result the bandwidth improved by about 20% but computation time increased about 125% over the recursive filter method. These methods of course do not show frequency multiples of the input. A further improvement concerning machine time will be discussed in Chapter 2.

1.7. Applications of the Recursive Filter:

The recursive technique developed in this Chapter can easily be applied to improve the speed of any numerical integration over a finite interval if the integrands have
Figure 1.5 SPECTRAL PLOT OF A GAUSSIAN SHAPED SIGNAL—RECURSIVE FILTERS TECHNIQUE USED.
Figure 1.6  SPECTRAL PLOT OF A TIME LIMITED COSINE INPUT—RECURSIVE FILTERS TECHNIQUE USED.
finite Z-transforms. These integrands can recursively be derived at discrete points and integrated by Simpson's rule or any other conventional method. Other useful applications are to determine the convolution of two discrete signals and to evaluate the output of a sampled data system in open or closed loop.
Chapter 2

Spectral Analysis via Heterodyning

2.1. Introduction:

The reciprocal relation between the bandwidth of a spectrum and the time duration of the analyzed waveform was discussed in the previous Chapter. This relationship in practical terms requires the processing of a larger amount of data if a better and more accurate resolution of the spectrum is desired. However, analyzing longer input signals for further improvement of the bandwidth certainly requires greater memory facility and increased machine time to affect the analysis and hence has the undesirable effect of making the goal of a real time analysis even more difficult to achieve. To enable one to process more data without increasing the memory requirements or using more machine time, a heterodyning technique will be presented.

This heterodyning technique basically takes the input signal and mixes it twice; (a) with a sine wave and, (b) with a cosine wave. The frequency $f_c$ of the trigonometric function
is the center frequency of the desired spectrum interval. To reduce the number of data points needed for the spectral analysis, these two new waveforms are integrated over each cycle of the sinosoidal wave and, as a result, two final signal are produced which consist of the frequencies \( f \pm f_c \) (\( f \) is an input frequency component near \( f_c \)) with the higher frequency being strongly attenuated by the integration operation. If the spectrum interval is small with respect to the middle frequency, analyzing these low frequency waves can enable one to determine the normalized energy density spectrum of the input signal.

Memory space is saved if the integration operation is simultaneously performed with the reading in of the data to be analyzed. Even more significant is the fact that the low pass filter property of the integration also permits one to replace the reference trigonometric waves by square waves allowing the integration operation process to be carried out by a mere summation. (In this case the signal investigated should not have any significant frequencies of the high odd harmonics of \( f_c \)). Figure 2.1 illustrates the above technique where, for convenience, the input and reference signals are square waves. (Notice the high frequency component introduced in Figure 2.1 (e) and (d)).
Figure 2.1 DERIVATION OF THE LOW FREQUENCY WAVES.

a. INPUT WAVEFORM \( f = 40 \, \text{c/s} \)

b. REFERENCE SINE WAVE, \( f_c = 50 \, \text{c/s} \), WHEN LOW PASSED FILTERED.

c. REFERENCE COSINE WAVE, \( f_c = 50 \, \text{c/s} \), WHEN LOW PASSED FILTERED.

d. INTEGRATION OF THE PRODUCT OF \( a \times b \) OVER EACH CYCLE OF \( b \) WAVE

e. INTEGRATION OF THE PRODUCT OF \( a \times c \) OVER EACH CYCLE OF \( b \) WAVE.
2.2. Analytical Development:

Let \( f(t) \) be a periodic waveform whose energy spectrum one desires to determine within an interval bounded by \( f_1 \) and \( f_2 \) and let \( f_c = \omega_c / 2\pi \) be the center frequency of this interval of interest. Consider a signal

\[
f(t) = A \times \sin(\omega t + \phi) \quad (2.1)
\]

where \( \omega_1 < \omega < \omega_2 \) and \( \phi \) is a constant phase angle, to be analyzed. Multiplying \( f(t) \) by \( \sin(\omega_c t) \) and \( \cos(\omega_c t) \) and integrating over each cycle \( k \) of the sinusoid where \( k = 0, 1, 2, \ldots, N \), one obtains

\[
I_s(kT; \omega) = \int_{2\pi k / \omega_c}^{2\pi (k+1) / \omega_c} A \times \sin(\omega t + \phi) \sin(\omega_c t) \, dt \quad (2.2)
\]

\[
I_c(kT; \omega) = \int_{2\pi k / \omega_c}^{2\pi (k+1) / \omega_c} A \times \sin(\omega t + \phi) \cos(\omega_c t) \, dt \quad (2.3)
\]

where \( T = 1/f_c \) and \( 2\pi k / \omega_c = kT \).

With \( \omega = \omega_c \) (2.2) and 2.3) yield the following solution when low pass filtered given in the matrix form
\[ I_S(kT;\omega_C) = \frac{A}{2} \begin{bmatrix} \sin(\phi) \\ \cos(\phi) \end{bmatrix}^{(k=1)T} = \frac{AT}{2} \begin{bmatrix} \sin(\phi) \\ \cos(\phi) \end{bmatrix} \] (2.4)

and representing the d.c. components.

When \( \omega \neq \omega_C \), (2.2) and (2.3) after performing the integration become

\[ I_S(kT;\omega) = \frac{A}{2} \begin{bmatrix} \sin(\phi) & \cos(\phi) \\ \cos(\phi) & -\sin(\phi) \end{bmatrix} \]

\[ I_C(kT;\omega) = \begin{bmatrix} \sin(\omega-\omega_C) + \frac{\sin(\omega+\omega_C)}{\omega-\omega_C} \\ \cos(\omega-\omega_C) + \frac{\cos(\omega+\omega_C)}{\omega-\omega_C} \end{bmatrix}^{(k+1)T} \]

Passing (2.5) through a low pass filter or equivalently when \( \omega+\omega_C >> |\omega-\omega_C| \), the integration becomes
\[
I_s(kT; \omega_a) = \begin{bmatrix} a & c \\ b & d \end{bmatrix} \begin{bmatrix} \cos(\omega_a(2k+1)T/2) \\ \sin(\omega_a(2k+1)T/2) \end{bmatrix}, \quad I_c(kT; \omega_a) = 
\]

\[ (2.6) \]

where \(|\omega - \omega_c|\) is the actual frequency of \(I_s\) and \(I_c\),

\[
a = A \times \sin(\phi) \sin(\omega_a T/2)/\omega_a, \quad b = A \times \cos(\phi) \sin(\omega_a T/2)/\omega_a
\]

and \(c = -b, \quad d = a\) when \(\omega < \omega_c\)

or \(c = b, \quad d = -a\) when \(\omega > \omega_c\)

For \(\omega_a T/2\) small enough to justify \(\sin \alpha \approx \alpha\) the amplitudes of \(I_s\) and \(I_c\) become independent of \(\omega_a\) and correspond to the d.c. solution of (2.4).

As a result of the mixing operation and the low pass filtering, each of the integrated functions \(I_s(kT; \omega_a)\) and \(I_c(kT; \omega_a)\) contains two frequency components, \(f_c \pm f_\alpha\), of the signal under investigation. The actual amplitudes of the integrated functions for the signal

\[
f(t) = A_+ \sin((\omega_c + \omega_a)t + \phi_+) + A_- \sin((\omega_c - \omega_a)t + \phi_-) \quad (2.7)
\]

therefore, are the sum of \(A_+\) and \(A_-\) of (2.7)
\[
I_s(kT; \omega_a) = \begin{bmatrix}
a_{1} + a_2 & -b_{1} + b_2 \\
b_{1} + b_2 & a_{1} - a_2 \\
\end{bmatrix} \times \begin{bmatrix}
\cos(T(2k+1)\omega_a/2) \\
\sin(T(2k+1)\omega_a/2) \\
\end{bmatrix}
\]

where according to (2.6)

\[a_1 = A_+ \times \sin(\phi_+) \sin(\omega_a T/2)/\omega_a, \quad b_1 = A_+ \times \cos(\phi_+) \sin(\omega_a T/2)/\omega_a\]

and \[a_2 = A_- \times \sin(\phi_-) \sin(\omega_a T/2)/\omega_a, \quad b_2 = A_- \times \cos(\phi_-) \sin(\omega_a T/2)/\omega_a\]

For a meaningful determination of a normalized energy density spectrum one must separate the cases \(\omega_c \neq \omega_a\) with respective amplitude \(A_+\) and \(A_-\). To effect this separation, consider the Gaussian shape filters as described in section 1.2 with equations (1.1) through (1.4). The digital convolution of the impulse response of a particular filter placed at \(\omega_o\) (1.2) with \(I_s\) and \(I_c\) yields, as the steady state solution after the signal terminates,

\[
g_s(kT; \omega_o) = \sum_{n=0}^{N} I_s(kT; \omega_a) e^{- (t_k - nT)^2} \times \left[ \cos(\omega_o (t_k - nT)) + j \sin(\omega_o (t_k - nT)) \right]
\]

\[= K_1(\omega_o) + jK_2(\omega_o)\]  

(2.9)
and in the same manner

$$g_C(kT; \omega_o) = K_3(\omega_o) + jK_4(\omega_o)$$  \hspace{1cm} (2.10)$$

where NT is the last significant term of I_S or I_C and
(K_1; K_3) and (K_2; K_4) respectively correspond to the cosine
and sine terms of I_S and I_C. Hence, comparing (2.9) and
(2.10) to (2.8) yield

$$\begin{bmatrix}
K_1(\omega_o) & K_2(\omega_o) \\
K_3(\omega_o) & K_4(\omega_o)
\end{bmatrix}
\begin{bmatrix}
a_1 + a_2 \\
b_1 + b_2
\end{bmatrix}
= 
\begin{bmatrix}
a_1 + a_2 \\
b_1 + b_2
\end{bmatrix}
\begin{bmatrix}
-a_1 + a_2 \\
b_1 - b_2
\end{bmatrix}$$  \hspace{1cm} (2.11)$$

With the spectral analysis described in section 1.5,
using discrete filters and squaring the steady state amplit-
tude, one finds from the cases of (2.6) and with $\omega a T/2$
small that

$$S(\omega - \omega_a) = (a_1^2 + b_1^2) = A_-(T/2)^2$$  \hspace{1cm} (2.12)$$

$$S(\omega + \omega_a) = (a_2^2 + b_2^2) = A_+(T/2)^2$$  \hspace{1cm} (2.13)$$
where $S$ is a component of the normalized energy density spectrum. Using the relations in (2.11),

$$S(\omega^2 + \omega^2) = [K_1(\omega_o) + K_2(\omega_o)]^2 + [K_3(\omega_o) + K_2(\omega_o)]^2 \quad (2.14)$$

$K_1, K_2, K_3, K_4$ are directly derived from (2.9) and (2.10) and the solution in (2.14) of the spectral energy density at $\omega_o - \omega_a$ and $\omega_o + \omega_a$ has been achieved.

2.3. Computational Technique:

The heterodyning method as described in the previous section can be used to analyze the spectrum of any desired waveform by computing $K_1(\omega_o)$ according to (2.9) and (2.10) for all $\omega_o$ of interest and evaluating the spectral components using (2.14). The bandwidth of the analyzed wave, is again fixed by the Gaussian decay. Additional saving in computation time is achieved if one utilizes several time filters over the signal period instead of one filter. These filters certainly should overlap each other and by choosing $\alpha = 12.6 \times (n+1)^2 / t^2$, where $t$ is the signal duration and $n$ the number of time filters, the total Gaussian envelope has less than 4% ripple. However, this saving in machine time will introduce
a larger bandwidth in the spectrum. Additional machine time can also be saved if the trigonometric functions of (2.9) and (2.10) are calculated recursively.

This technique was tested on the Rice University Computer using various numbers of filter over the signal duration. The bandwidths obtained with this method of spectral analysis were about 5% wider than those achieved using the methods described in Chapter 1. A sideband of about 3% of the computed spectral component appeared at the frequency $2f_c-f$. This undesired sideband always arises due to the round-off errors and the inaccuracy introduced by the numerical integration. Results are shown in Fig. 2.2.

2.4. Comparison of Techniques:

The four different techniques as described in Chapters 1 and 2 to determine the energy density spectrum were compared — each analyzing a cosine wave with a frequency of 50 c/s in a 10 second window sampled every 2 milliseconds. The bandwidths were obtained to within 10% accuracy. The machine time required for each method to calculate 101 spectral points is tabulated in Table 2.1. Special emphasis was given to achieve effective programming and these programs were executed on the Rice University Computer which is a
Figure 2.2  ENERGY DENSITY SPECTRUM USING HETERODYNING TECHNIQUE. ANALYZED WAVE—50.05 c/s COSINE IN A 100 SECOND WINDOW (50000 SAMPLES) (COMPUTING TIME ON RICE MACHINE 10.8 MINUTES)
Table 2.1. Machine time needed to calculate 101 spectral points. Input signal consists of 5000 samples.

<table>
<thead>
<tr>
<th>Method</th>
<th>Machine time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Conventional method using Gaussian shape filters, equation (1.15) and evaluating recursively the sin and cosine.</td>
<td>22.5 min.</td>
</tr>
<tr>
<td>2. Recursive filter and input with the Gaussian shape, equations (1.28), (1.24).</td>
<td>11 min.</td>
</tr>
<tr>
<td>3. Heterodyning technique equations (2.9), (2.10), (2.14)</td>
<td>3.9 min.</td>
</tr>
<tr>
<td>4. Heterodyning technique, evaluating recursively sine and cosine. Equations (2.9), (2.10), (2.14)</td>
<td>1.1 min.</td>
</tr>
</tbody>
</table>

Note: Method 1 with the conventional subroutine to calculate the sine and cosine requires twice the time mentioned.
fairly slow machine with a cycle time of 2 microseconds and an average multiplication time of 0.25 milliseconds. The measured times excluded the time required to produce the signal under investigation.

In conclusion, using the heterodyning technique a reduction of computation time by a factor \( n \) (where \( n \) is the number of samples per cycle of the original signal under investigation) was achieved (Figure 2.2).
Chapter 3
Fourier Transformations and Integrals with Oscillatory Integrands

3.1. Introduction

The solution of many problems in applied mathematics leads to the evaluation of integrals involving oscillatory integrands. These integrals arise in the solution of problems in wave propagation, filtering of data, etc. and, in general, places where Fourier transform technique are applicable. Exact evaluation of the integrals is often impossible and, in the cases where asymptotic analytical evaluation is not suitable, numerical computation must be performed. To be specific, numerical evaluation of integrals of the following type will be examined:

\[ S(\omega) = \int_{a_1}^{a_2} f(x) \sin(\omega x) \, dx \]  \hspace{1cm} (3.1)

and

\[ C(\omega) = \int_{a_1}^{a_2} f(x) \cos(\omega x) \, dx \]  \hspace{1cm} (3.2)

where \( f(x) \) is any arbitrary function of \( x \) slowly varying with respect to \( \sin(\omega x) \) and the endpoints of integration are not necessarily finite. It is often required to determine (3.1) or (3.2) for some entire range of the parameter \( \omega \), and such integrals can rarely be evaluated exactly.
analytically unless $f(x)$ is given by a closed expression. Certainly, when $f(x)$ is only available at discrete points, a numerical integration must be performed and approximate methods then must be applied to obtain numerical values which, after all, are the ultimate aim in these problems.

Conventional integration methods [6], [7] (e.g. Simpson's rule, trapezoidal rule) are not suitable to compute the integrals in question especially when $\omega$ is large because of the following reasons:

1°. The rapid oscillation of the trigonometric function for large $\omega$ requires very small intervals of integration to maintain a desired accuracy with a simple polynomial approximation of the integrand.

2°. The contribution to the integral from each consecutive cycle of $\sin(\omega x)$ tends to increase the error because of the extremely strong cancellation between the positive and negative portions of the cycle.

3°. The enormous number of operations involved for the integration (addition and multiplication) makes these methods inefficient and tedious for a large range of $\omega$.

Improved methods have been suggested by Filon [8] and Luke [9] which avoid the first and third difficulties by
by using polynomial approximations for $f(x)$ rather than for the entire integrand. A new scheme which attempts to alleviate the second difficulty as well was offered by Hurwitz and Zwiefel [10]. They treated all half cycles of the trigonometric function in an identical manner to eliminate the cancellation errors, and employed the Gaussian type integration procedure [11] to minimize the error of the final result, which will depend on the number of points per half cycle used. The actual result is obtained by first an integration over the individual half cycles and then a summation of these partial results.

In this chapter, a new and more promising scheme will be presented. The difficulties mentioned above are eliminated in the following technique by treating all full cycles of the trigonometric function in an identical manner. During each cycle of the trigonometric function, a Taylor series expansion is made of the slowly varying function $f(x)$ about the center of the cycle; and then, the integration with the Taylor series is performed over each individual cycle. This technique certainly promises more accurate results than the parabolic fit used by Filon, and it will later be shown that this approximation depends on the behavior of $f(x)$ at the
center of each cycle of \( \sin(\omega x) \) or \( \cos(\omega x) \). The number of points required depends on the convergence of the odd derivatives of \( f(x) \) to integrate (3.1) and on the even derivatives for (3.2). The summation of the partial results gives the desired integral (3.1) or (3.2) and no cancellation errors arise. A great advantage in this method is that as \( \omega \) increases the better the fit will become and in addition fewer terms of the expansion will be needed.

The accuracy of the forthcoming technique corresponds to the accuracy of the Taylor series coefficients used which can be directly calculated from \( f(x) \) or derived from a sequence of values given at discrete points. To evaluate the coefficients at the points of interest from the values of \( f(x) \) available, an interpolation procedure is required and usually a linear interpolation will be sufficient. In such a case, this method will be found very useful especially when results of the integral are needed for a wide range of \( \omega \).

The number of arithmetic operations involved in computing these integrals is directly dependent on the convergence of the Taylor series expansion and the derivatives (odd or even)
of $f(x)$. Moreover, considerable machine time can be saved without influencing the accuracy of the final result with suitable programming and early stage interpolation.

3.2. Taylor Series Fitting

The representation of the function $f(x)$ as an infinite power series of its variable $x$ about $x = x_0$ where $x_0 = \pi k/\omega$, $k = 1, 3, 5, \ldots$, is the basis of the algorithm to be developed.

Let $f(x)$ be a function with derivatives of all orders at $x = x_0$, and assume that in a neighborhood of $x_0$, $f(x)$ may be represented as the power series

$$f(x) = \sum_{n=0}^{\infty} C_n (x-x_0)^n$$

where $(x_0 - \delta) < x < (x_0 + \delta)$ and $\delta$ represents the length of the interval of convergence.

The coefficients $C_n$ are given by the rule

$$C_n = \frac{f^{(n)}(x_0)}{n!}$$

If the origin is shifted to $x_0$, then

$$f(x+x_0) = \sum_{n=0}^{\infty} \frac{f^{(n)}(x_0)}{n!} (x+x_0)^n$$

(3.3)
The error $R_m$ involved in a truncated series after the $m^{\text{th}}$ term is given by

$$|R_m| \leq \frac{B^{(m)}}{(n+1)!} (x-x_0)^{n+1}$$

where $B^{(m)}$ is the upper bound of $|f^{(m+1)}(x)|$ (because the exact location of $x$ usually is not known).

### 3.3. Integration

The region of integration will be divided into segments of one full cycle each, where the governing factor in this partition is the trigonometric function and every section starts at

$$x = k_1 + k\pi/\omega \quad (3.4)$$

and terminates at

$$x = k_1 + (k+2)\pi/\omega \quad (3.5)$$

where $k_1 = 2\pi \ell/\omega$, $\ell$ is an integer, and $k = 0, 2, 4, \ldots , N$. $k_1$ and $N$ are the constants that shift and keep $x$ within the
region of importance.

$f(x)$ will be fitted about the center of the section at

$$x_{ok} = k_1 + (k+1)\pi/\omega$$  \hspace{1cm} (3.6)

Figure 3.1 shows a detailed partition of $S(\omega)$. $C(\omega)$ is similar but with a 90 degree shift.

Let $f(x)$ now be replaced by the Taylor series expansion with the origin moved to $K_1 + (k+1)\pi/\omega$. The integral $S(\omega)$ of (3.1) for every single cycle after applying (3.3) becomes

$$S_k(\omega) = \sum_{n=0}^{(n)} \frac{f(x_{ok})}{n!} \int_{k_1+k\pi/\omega}^{k_1+(k+2)\pi/\omega} x^n \sin(\omega x) dx$$

Since $k$ is even, it is more convenient to use the following form,

$$S_k(\omega) = \sum_{n=0}^{(n)} \frac{f(x_{ok})}{n!} \int_{-\pi/\omega}^{\pi/\omega} x^n \sin(\omega x) dx$$  \hspace{1cm} (3.7)

as $\sin(\omega x + (k+1)\pi) = -\sin(\omega x)$ for $k$ even.
Introducing a new variable \( u \), where \( u = \omega x \) and \( du = \omega dx \) we get

\[
S_K = -\sum_{n=0}^{\infty} \frac{f(x_{ok})}{n! \omega^{n+1}} L_n
\]

where

\[
L_n = \int_{-\pi}^{\pi} u^n \sin u \, du \tag{3.9}
\]

The integral \( S(\omega) \) is the summation of all even \( S_K(\omega) \)

\[
S(\omega) = -\sum_{k=0}^{N} \sum_{n=0}^{\infty} \frac{f(x_{ok})}{n! \omega^{n+1}} L_n + R_S \tag{3.10}
\]

where \( x_{ok} = k \pi + (k+1)\pi/\omega \), \( k = 0, 2, 4, ..., N \), and \( R_S \) is the integral of the two end pieces which are not full cycles of \( \sin(\omega x) \). \( R_S \) can be integrated by conventional methods and both its intervals should not exceed \( 3\pi \). \( L_n \) of (3.9) can be simplified by integrating by parts

\[
L_n = -u^n \cos(u) \bigg|_{-\pi}^{\pi} + n \int_{-\pi}^{\pi} u^{n-1} \cos(u) \, du
\]

\[
= [\pi^n - (-\pi)^n] + n[u^{n-1} \sin(u)]_{-\pi}^{\pi} - \int_{-\pi}^{\pi} u^{n-2} \sin(u) \, du \tag{3.11}
\]
Since the first term vanishes when \( n \) is even, the second one is always zero and the third is actually \( L_{n-2} \). The following recursive formula is thus obtained from (3.11)

\[
L_{2n-1} = 2\pi \frac{(2n-1)}{(2n-1)!} - \frac{(2n-1)(2n-2)}{(2n-3)!} L_{2n-3}
\]

(3.12)

In (3.10) we use \( L_n/n! \) and it can be very easily shown that this term tends to zero as \( n \) goes to infinity by writing (3.12) in a series form

\[
\frac{L_{2n-1}}{(2n-1)!} = \sum_{i=1}^{\infty} \frac{2\pi(2i-1)}{(2i-1)!} (-1)^{n-i} \frac{1}{(2n-1)!}
\]

For \( n \to \infty \) this series is the series of \( 2\pi \sin(\pi) \) which is 0. With a direct integration we get \( L_{2n} = 0 \) and

\[
L_1 = 2\pi
\]

(3.13)

By precisely similar analysis, the integral \( C(\omega) \) (3.2) can be deduced as

\[
C(\omega) = -\sum_{n=1}^{\infty} \frac{K_{2n}}{(2n)!} \sum_{k=0}^{N} f^{(2n)}(x_{ok}) + R_C
\]

(3.14)
where

\[ K_{2n} = -2n[2\pi^{2n-1} + (2n-1)K_{2n-2}] \quad (3.15) \]

The convergence of \( \frac{K_{2n}}{(2n)!} \) can be proved in the same manner as for \( \frac{L_n}{n!} \), as it turns out to be the same series of \( \sin(\pi) \).

A direct integration leads to \( K_0 = 0, K_{2n-1} = 0 \), and

\[ K_2 = -4\pi \quad (3.16) \]

The segments of integration are as illustrated in Figure 3.1 with a 90 degree shift to fit the cosine wave.

3.4. Simultaneous Integration of \( S(\omega) \) and \( C(\omega) \):

In many applications (3.1) and (3.2) appear in a combined complex form (e.g. complex Fourier transform or its inversion). When such an integral is separately calculated for real and imaginary parts a short cut in the algorithm can be taken because of the strong relationship of the trigonometric functions.

From (3.11) one derives

\[ L_{2n-1} = 2\pi^{2n-1} + (2n-1)K_{2n} \quad (3.17) \]
and in a similar way

\[ K_{2n} = -2nL_{2n-1} \]  \hspace{1cm} (3.18)

The final form from (3.10) is

\[ S(\omega) = - \sum_{n=1}^{\infty} \frac{L_{2n-1}}{(2n-1)!\omega^{2n}} \sum_{k=0}^{N} f(2n-1)(x_{0k}) \]  \hspace{1cm} (3.19)

where \( k \) is an even integer and \( n = 1,2,3,\ldots \).

By applying (3.18) to (3.14)

\[ C(\omega) = \sum_{n=1}^{\infty} \frac{L_{2n-1}}{(2n-1)!\omega^{2n+1}} \sum_{k=0}^{N} f(2n)(x_{0k}) \]  \hspace{1cm} (3.20)

where again \( k \) is even and \( n = 1,2,3,\ldots \). Here the similarity (3.19) and (3.20) will save about half of the computation effort needed, because of the fact that to evaluate a derivative of any order one must compute all the lower order derivatives. Moreover to get \( C(\omega) \) one may integrate 3/4 of the first cycle by any conventional method and proceed with the \( S(\omega) \) technique.
3.5. Comments About $f(x)$:

A Taylor series expansion is fitted to $f(x)$ for each full cycle of the trigonometric function. An increase in frequency causes a reciprocal decrease in the intervals and a further improvement in the fit is achieved. In this case, when the frequency increases, one is able to maintain a desired degree of accuracy with even fewer terms of the Taylor series. Moreover, if within the boundaries of integration $f(x)$ has a fast change or discontinuity, the integration process can be split into two regions excluding the ill-behaved section of $f(x)$. The missing part can be integrated by any other method and added to the result. It is obvious that there are no restrictions on the number of ill-behaved sections of $f(x)$, but if this number is large and these sections occupy a high percentage of the interval to be integrated then a conventional method should be employed.

When $f(x)$ and its derivatives are fairly smooth within the interval of integration (or part of it) or $\omega$ is very large, equation (3.19) can be modified to an algorithm for even faster computation. Let equation (3.19) be multiplied and divided by the factor $2\pi/\omega$

$$S(\omega) = -\sum_{n=1}^{\infty} \frac{L_{2n-1}}{(2n-1)!} \omega^{2n} \left( \frac{N}{\sum_{k=0}^{N} f(x_{ok})} \frac{(2n-1)\omega}{2\pi} \right) \frac{\omega}{2\pi}$$
now the summation with the above assumptions can be changed to an integral

\[
S(\omega) = - \frac{2}{\omega} \sum_{n=1}^{\infty} \frac{L_{2n-1}}{2\pi (2n-1)! \omega^{2n-1}} \int_{k_1}^{k_1 + (N+2) \pi / \omega} f(2n-1)(x) \, dx
\]

and performing the integration,

\[
S(\omega) = \frac{2}{\omega} \sum_{n=1}^{\infty} \frac{L_{2n-1}}{2\pi (2n-1)! \omega^{2n-1}} \left[ f^{(2n-2)}(k_1) \right.
\]
\[
- f^{(2n-2)}(k_1 + (N+2) \pi / \omega) \left. \right]
\]

(3.21)

Applying a similar analysis to (3.20) gives,

\[
C(\omega) = - \frac{2}{\omega} \sum_{n=1}^{\infty} \frac{L_{2n-1}}{2\pi (2n-1)! \omega^{2n}} \left[ f^{(2n-1)}(k_1) \right.
\]
\[
- f^{(2n-1)}(k_1 + (N+2) \pi / \omega) \left. \right]
\]

(3.22)

Equation (3.21) has the even derivatives only while (3.22) has the odd ones. If \( f(x) \) may be replaced with sufficient accuracy by the first 3 terms to calculate \( S(\omega) \) and with 4 terms in evaluating \( C(\omega) \) it is clear from (3.21) and (3.22) that this fast procedure has an advantage over Filon's method which is based on a parabolic fit.
Similar results are obtained when integrating (3.1) or 
(3.2) by parts,

\[ C(\omega) = \frac{f(x) \sin(\omega x)}{\omega} \bigg|_{a_1}^{a_2} - \frac{1}{\omega} \int_{a_1}^{a_2} f'(x) \sin(\omega x) \, dx \quad \text{etc.} \]

which turns to be the exact solution as in (3.22) if the 
\( f^{(n)}(x) \) series converges.

The following examples will show how this fast method 
works. Let

\[ C(\omega) = \int_{a_1}^{a_2} x^3 \cos(\omega x) \, dx \]

where \( a_1 = 2\pi k_1/\omega \), and \( a_2 = 2\pi k_2/\omega \), \( k_1 \) and \( k_2 \) are integers. 
Using (3.22), we get

\[ C(\omega) = -\frac{1}{\omega} \frac{2}{2} (3a_1-3a_2) - \left( \frac{\pi}{6} - 1 \right) \frac{0}{4} \]

which is the exact solution.

By using (3.21) for the same \( f(x) \) and boundaries,

\[ S(\omega) = \frac{1}{\omega} (a_1^3-a_2^3) + \left( \frac{\pi}{6} - 1 \right) \frac{6a_1-6a_2}{\omega} + 0 \]

which is correct if \( \omega \) is very large so the second term is 
negligible.
3.6. Computational Technique:

The crucial factors controlling the application of this method in evaluating (3.1) and (3.2) are the convergence of the coefficients $K_n$ and $L_n$ and the smoothness of $f(x)$ and all of its derivatives (excluding all abrupt changes or discontinuities that must be computed by conventional methods).

3.6.1. $K_n$ and $L_n$ Coefficients: A look at Table 3.1 shows how $L_n/n!$ or $K_n/n!$ converge (recall that $K_{2n}/(2n)! = -L_{2n-1}/(2n-1)!$).

Table 3.1: Values of $L_{(2n-1)/(2n-1)!}$

<table>
<thead>
<tr>
<th>$n$</th>
<th>$L_{2n-1}/(2n-1)!$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6.28318530</td>
</tr>
<tr>
<td>2</td>
<td>4.05224024</td>
</tr>
<tr>
<td>3</td>
<td>1.04808782</td>
</tr>
<tr>
<td>4</td>
<td>0.15044123</td>
</tr>
<tr>
<td>5</td>
<td>0.01385054</td>
</tr>
<tr>
<td>6</td>
<td>0.00089031</td>
</tr>
<tr>
<td>7</td>
<td>0.00004228</td>
</tr>
<tr>
<td>8</td>
<td>0.00000154</td>
</tr>
<tr>
<td>9</td>
<td>0.00000009</td>
</tr>
<tr>
<td>$(2n-1)\text{even}$</td>
<td>0</td>
</tr>
<tr>
<td>$n + \infty$</td>
<td>0</td>
</tr>
</tbody>
</table>
Approximately one additional term is needed for each decimal digit of accuracy desired, but as the element \(1/\omega^{n+1}\) also appear in the summation ((3.19) or (3.20)), the number of terms required is much less. For \(\omega = 100\), one finds that only the first two coefficients of \(L\) or \(K\) are sufficient, and when \(\omega\) is greater then 1000 only the first component will have a significant contribution to the integral.

3.6.2. Derivatives of \(f(x)\): The method suggested requires the evaluation of the derivatives of \(f(x)\) at the points where the angle of the trigonometric function is \((K+1)\pi\) \((k = \text{even integer})\) and the summation of these values and the fast algorithm requires only the boundary derivatives. When \(f(x)\) is given in an analytic form, the derivatives should be directly computed at the points needed. A round-off error arises which depends on the type of function and number of digits utilized. If \(f(x)\) is given only at discrete points, the derivatives can be calculated at these points causing an error that depends on the interval between samples, the order of the derivative, the formula utilized and the number of digits used. A detailed discussion of this subject and a description of a new method for a successive solution is
given in Chapter 4. The errors that exist in the derivatives probably will accumulate during the summation and cause a loss in accuracy. The accumulated error must be reconsidered in every computation because it has a direct dependence on the frequency, the smoothness of $f(x)$ and the convergence of the derivatives. When (3.1) and (3.2) are required for a wide band of $\omega$, it is preferable to evaluate the derivatives at points separated by such an interval that a linear interpolation can determine any derivative at points between these values with sufficient accuracy.

3.6.3. Discontinuity of $f(x)$: Rapid changes in $f(x)$ within one cycle interval certainly will introduce an error in the integration. As mentioned in paragraph 3.5, these segments should be calculated in a different way and must be located before using the technique discussed. Two ways are suggested; a pre-study of the given $f(x)$, or, investigating the computed derivatives. The second way is less accurate, as the method of differences to calculate derivatives tends to smooth discontinuities or rapid changes. For example, integrating \[ \int_{0}^{\infty} \frac{\sin(\omega x)}{x} \, dx \] for $\omega$ large is impossible by the
discussed technique because of the discontinuity of $f(x)$ at zero, so it is advisable to start the integration by this method from the second cycle (or more) and use Simpson's rule to integrate the remaining section.

3.6.4. Program Flowchart: Table 3.2 gives the flowchart for simultaneous derivation of $S(\omega)$ and $C(\omega)$. (Symbols used in this flowchart appear in Chapter 4.)

Table 3.2: Program Flowchart

```
Start

Read $N,D,\omega$

Initialize -- constants and $M(D+1,N+3)$

Read $f(kT)$ store in $M(1,k)$

A
```
Subroutine: evaluate derivatives and store in $M(n+1,k)$

Calculate points of $X_{ok}$ 
$k = 1, 3, 5, ..., N+1.$

Calculate $-$ 
$S_1 = \sum_{k=1}^{N+1} f(X_{ok}), S_2 = \sum_{k=1}^{N+1} f(X_{ok})$ 
use interpolation

Calculate $-$ 
$A_{2n-1} = L_{2n-1}/(2n-1)$ 
$B_{2n-1} = A_{2n-1}/\omega^{2n}$

Calculate $-$ 
$S_{2n-1}(\omega) = B_{2n-1} \times S_1$ 
$C_{2n}(\omega) = B_{2n-1} \times S_2/\omega$
calculate →
\[ S(\omega) = S(\omega) + S_{2n-1}(\omega) \]
\[ C(\omega) = C(\omega) + C_{2n}(\omega) \]

is \( n = D \) ?

Print Results

new \( \omega \) ?

Stop
3.7. Examples and Computed Results

Two different functions of \( f(x) \) that have analytic solutions for both \( S(\omega) \) and \( C(\omega) \) were investigated and examined using the proposed method.

3.7.1. Example 1 \( f(x) = \cos(\omega t + \phi) \):

The integral

\[
S(\omega) = \int_0^\pi \cos(\omega t + \phi) \sin(m \omega t) \, dt \tag{3.23}
\]
yields the solution

\[
S(\omega) = \frac{2m \cos(\phi)}{\omega(m^2 - 1)} - \frac{2\cos(\phi)}{m\omega} \quad (m > 1)
\]

The odd derivatives of \( f(x) \) that contribute to the summation are,

\[
f^{(2n-1)}(t) = (-1)^n \omega^{2n-1} \sin(\omega t + \phi) \tag{3.24}
\]

Applying (3.24) to (3.19),

\[
S(\omega) = -\sum_{n+1}^{\infty} \frac{(-1)^n L_{2n-1}}{\omega m^{2n} (2n-1)!} \sum_{k=0}^{N} f(x_{ok}) \tag{3.25}
\]
(3.25) was calculated and results from the computer are shown in Table 3.3. If \( \omega > 1 \), \( f(x) \) can be considered as a slowly varying function and may be successfully treated with the fast technique as described in section 3.5 using equation (3.21).

As \( \pi / \omega \) is a very small angle \( \cos(\pi / \omega + \phi) \approx \cos(\phi) \) and \( \cos(\pi - \pi / \omega + \phi) \approx \cos(\pi + \phi) \)

Using this approximation (3.21) becomes

\[
S(\omega) = \frac{2\pi}{2\pi 1! \omega m} [\cos(\phi) - \cos(\phi + \pi)] \\
+ \frac{2\pi^3 - 12\pi}{2\pi 3! \omega m} [\cos(\phi + \pi) - \cos(\phi)] + ... 
\]

and in a condensed form

\[
S(\omega) = \frac{2 \cos(\phi)}{\omega m} [1 - \frac{\pi^2 - 6\pi}{6\omega^2} + ...] 
\]

For \( m \gg 1 \) this solution is an excellent approximation.

3.7.2. Example 2 — \( f(x) = e^{-t} - \frac{1}{\alpha} e^{-at} \): In this case a simultaneous solution for \( S(\omega) \) and \( C(\omega) \) will be shown.
Table 3.3: Computed Results of Equation (3.25)

\[ \omega = 1 \quad D = 10 \]

<table>
<thead>
<tr>
<th>m</th>
<th>( \phi ) degree</th>
<th>Analytic solution computer results</th>
<th>Proposed technique computer results</th>
<th>Digits of accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0</td>
<td>2.02020202020\times10^{-1}</td>
<td>2.02020202021\times10^{-1}</td>
<td>11</td>
</tr>
<tr>
<td>100</td>
<td>0</td>
<td>2.00020002000\times10^{-2}</td>
<td>2.00020002001\times10^{-2}</td>
<td>11</td>
</tr>
<tr>
<td>1000</td>
<td>0</td>
<td>2.00000200000\times10^{-3}</td>
<td>2.00000200000\times10^{-3}</td>
<td>11</td>
</tr>
<tr>
<td>100</td>
<td>30</td>
<td>1.73222402997\times10^{-2}</td>
<td>1.73222403027\times10^{-2}</td>
<td>8</td>
</tr>
<tr>
<td>1000</td>
<td>30</td>
<td>1.73205253962\times10^{-3}</td>
<td>1.73205254038\times10^{-3}</td>
<td>8</td>
</tr>
<tr>
<td>100</td>
<td>60</td>
<td>1.00010001000\times10^{-2}</td>
<td>1.00010001052\times10^{-2}</td>
<td>10</td>
</tr>
<tr>
<td>1000</td>
<td>60</td>
<td>1.00000100000\times10^{-3}</td>
<td>1.00000100128\times10^{-3}</td>
<td>9</td>
</tr>
<tr>
<td>100</td>
<td>90</td>
<td>(-5.7 \times 10^{-14})</td>
<td>5.9 \times 10^{-12}</td>
<td>too small to compare</td>
</tr>
<tr>
<td>1000</td>
<td>90</td>
<td>(-5.7 \times 10^{-15})</td>
<td>1.2 \times 10^{-12}</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>180</td>
<td>(-2.00020002000\times10^{-2})</td>
<td>(-2.00020002000\times10^{-2})</td>
<td>12</td>
</tr>
<tr>
<td>1000</td>
<td>180</td>
<td>(-2.00000200000\times10^{-3})</td>
<td>(-2.00000200021\times10^{-3})</td>
<td>10</td>
</tr>
</tbody>
</table>
\[ S(\omega) = \int_0^\infty \left( e^{-t} - \frac{1}{\alpha} e^{-\alpha t} \right) \sin(\omega t) \, dt \]

yields the solution

\[ S(\omega) = \frac{\omega}{1 + \omega^2} - \frac{1}{\alpha} \times \frac{\omega}{\alpha^2 + \omega^2} \]

and

\[ C(\omega) = \frac{1}{1 + \omega^2} - \frac{1}{\alpha^2 + \omega^2} \]

using (3.19) we get,

\[ S(\omega) = -\frac{D/2}{\xi} \sum_{n=1}^{\infty} \frac{L_{2n-1}}{(2n-1)! \omega^{2n}} \sum_{k=0}^{\infty} (-1)^{2n-1} \]

\[ \times \left[ e^{-(2k+1)\pi/\omega} - \alpha^{2n-2} e^{-(2k+1)\alpha\pi/\omega} \right] \]

(3.26)

and applying (3.20)

\[ C(\omega) = \frac{D/2}{\xi} \sum_{n=1}^{\infty} \frac{L_{2n-1}}{(2n-1)! \omega^{2n+1}} \]

\[ \times \sum_{k=0}^{\infty} \left[ e^{-(2k+1)\pi/\omega} - \alpha^{2n-1} e^{-(2k+1)\alpha\pi/\omega} \right] \]

(3.27)
The terms in the summation over \( k \) must be checked to determine how many of them are needed reasonably to assure convergence. (3.26) and (3.27) were reduced to,

\[
S(\omega) = \frac{D}{2} \sum_{n=1}^{\infty} A_{2n-1} (B-C_n D)
\]

and

\[
C(\omega) = \frac{D}{2} \sum_{n=1}^{\infty} \frac{A_{2n-1}}{\omega} (B-C_n E)
\]

where

\[
A_{2n-1} = \frac{L_{2n-1}}{(2n-1)! \omega^{2n}}, \quad B = \sum_{k=0}^{\infty} e^{-(2k+1)\pi/\omega}, \quad C_n = a^{2n-2}
\]

and

\[
D = \sum_{k=0}^{\infty} e^{-(2k+1)a\pi/\omega}, \quad E = aD
\]

computed results are tabulated in Table 3.4.

The second term of \( f(x) \) vanishes rapidly with increasing \( a \) and its actual contribution only exists for small \( t \). Since the first point in the process of the integration is at \( \pi/\omega \) for both \( S(\omega) \) and \( C(\omega) \) the accuracy of the final result decreases with higher \( a \) (because of the poor fit of the Taylor series expansion in the first cycle) but improves for large \( \omega \). The additional inaccuracy in \( C(\omega) \) also arises because of
Table 3.4: Computed results of equations (3.26) and (3.27) compared to the analytic solution. $D = 10$

<table>
<thead>
<tr>
<th>$\omega$</th>
<th>$\alpha$</th>
<th>Number of points summed</th>
<th>Digits of accuracy in $S(\omega)$</th>
<th>Digits of accuracy in $C(\omega)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>2</td>
<td>50</td>
<td>10</td>
<td>9</td>
</tr>
<tr>
<td>10</td>
<td>5</td>
<td>50</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>50</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>10</td>
<td>100</td>
<td>50</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>100</td>
<td>2</td>
<td>400</td>
<td>10</td>
<td>7</td>
</tr>
<tr>
<td>100</td>
<td>5</td>
<td>400</td>
<td>9</td>
<td>8</td>
</tr>
<tr>
<td>100</td>
<td>10</td>
<td>400</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>100</td>
<td>100</td>
<td>400</td>
<td>7</td>
<td>4</td>
</tr>
<tr>
<td>1000</td>
<td>2</td>
<td>3700</td>
<td>10</td>
<td>5</td>
</tr>
<tr>
<td>1000</td>
<td>5</td>
<td>3700</td>
<td>9</td>
<td>7</td>
</tr>
<tr>
<td>1000</td>
<td>10</td>
<td>3700</td>
<td>9</td>
<td>8</td>
</tr>
<tr>
<td>1000</td>
<td>100</td>
<td>3700</td>
<td>9</td>
<td>9</td>
</tr>
</tbody>
</table>
the significant contribution to the integral at the origin. In this case, it is preferable to start the integration, using this technique, from the second cycle and integrate separately for the first one. The results will have 9 to 10 digits accuracy (out of 12).

The fast technique of section 3.5 will also lead to similar results. The derivatives at infinity are zero and at the origin $1 - r^{n-1}$.

Using (3.21) and taking the first approximation

$$S(\omega) = \frac{1}{\omega} \left[ 1 - \frac{1}{\omega} \right] + \text{terms of } \frac{1}{\omega^{2n+1}} \quad n=3,5,\ldots$$

and from (3.22)

$$C(\omega) = \frac{1}{\omega^2} (1-1) + \ldots$$

These results fit the analytic solution when $\omega \gg 1, \omega \gg \alpha$. 
Chapter 4
Numerical Evaluation of Derivatives

4.1. Introduction

Differential calculus is based on taking a limit of \( \Delta y/\Delta x \) as the increment of \( \Delta x \) approaches zero. In many practical problems it may not be feasible to obtain the limit, especially when the function being differentiated is given in a tabular form for selected values of the independent variable. In such cases only approximate values of the derivatives can be determined. The need for a short algorithm to compute a set of derivatives of \( f(x) \) (at points \( x_{ok} \)) for equations (3.19) and (3.20), up to the \( n^{th} \) order, arises when \( f(x) \) is given at discrete points and the range of integration is very large.

The method discussed in this section obtains approximate expressions for the derivatives from Taylor series expansions [6]. The technique developed enables one to calculate derivatives at all points including the end points, with the same degree of accuracy and with a minimum number of multiplications.
4.2. Derivatives in Terms of Difference Ratios

Given a function \( f(x) \) at discrete points \( x = kh \), \( k = 0,1,2,3, \ldots, N+2 \) in the form \( f_k = f(kh) \), \( f_{k+1} = f(kh+h) \) where \( h \) is the interval of differencing, then one may use a Taylor series expansion to write,

\[
f_1 = f_o + \frac{h}{1!} f_o^{(1)} + \frac{h^2}{2!} f_o^{(2)} + \frac{h^3}{3!} f_o^{(3)} + \ldots
\]

and in a general form for the \( k^{\text{th}} \) value,

\[
f_{k+i} = f_k + \sum_{n=1}^{\infty} \frac{(\pm h)^n}{n!} f_k^{(n)}
\]

where

\[
i = 0,1,2,\ldots,(N+2-k)
\]

Applying (4.1) with \( k+1 \) gives,

\[
f_{k+1} = f_k + hf_k^{(1)} + \frac{h^2}{2} f_k^{(2)} + \frac{h^3}{6} f_k^{(3)} + \ldots \quad (4.2)
\]

\[
f_{k-1} = f_k - hf_k^{(1)} + \frac{h^2}{2} f_k^{(2)} - \frac{h^3}{6} f_k^{(3)} + \ldots \quad (4.3)
\]
Subtracting (4.3) from (4.2) yields the first derivative at point \( k \) as

\[
\frac{f_k^{(1)}}{2h} = \frac{f_{k+1} - f_{k-1}}{2h} - \frac{h^2}{6} f_k^{(3)} + \ldots
\]

In the same manner by first eliminating the \( f_k^{(1)} \) term the second derivative is

\[
\frac{f_k^{(2)}}{h^2} = \frac{1}{h^2} (f_{k+1} - 2f_k + f_{k-1}) - \frac{h^2}{12} f_k^{(4)} + \ldots
\]

and using (4.1) with \( k \pm 1, k \pm 2 \)

\[
\frac{f_k^{(3)}}{2h^3} = \frac{1}{2h^3} (f_{k+2} - 2f_{k+1} + 2f_{k-1} - f_{k-2}) - \frac{h^2}{4} f_k^{(5)} + \ldots
\]

e tc. for any order of \( f_k^{(n)} \). This method of differences has for the dominant term a numerator \((n+1)\) terms) divided by \( h^n \) for even \( n \) or \( 2h^n \) for \( n \) odd and a remainder \(\text{ (an infinite series) } \) of order \( h^2 \). The numerator of the even order derivatives can be found by taking from the Pascal triangle the coefficients from the \( 2n^{\text{th}} \) row with alternating signs \((\pm)\) starting with the plus.
The Pascal triangle is

<table>
<thead>
<tr>
<th>row</th>
<th>values</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1 1</td>
</tr>
<tr>
<td>2</td>
<td>1 2 1</td>
</tr>
<tr>
<td>3</td>
<td>1 3 3 1</td>
</tr>
<tr>
<td>4</td>
<td>1 4 6 4 1</td>
</tr>
<tr>
<td>5</td>
<td>1 5 10 10 5 1</td>
</tr>
<tr>
<td>6</td>
<td>1 6 15 20 15 6 1</td>
</tr>
<tr>
<td>7</td>
<td>1 7 21 35 35 21 7 1</td>
</tr>
</tbody>
</table>

and for example the sixth derivative is

\[
f_k^{(6)} = \frac{1}{h^6} (f_{k+3} - 6f_{k+2} + 15f_{k+1} - 20f_k + 15f_{k-1} - 6f_{k-2} + f_{k-3}) + \text{Remainder of order } h^2
\]

The coefficients of the numerator in the odd derivatives are determined in a different way by taking from the Pascal triangle the $2N^{\text{th}}$ line with alternating $\pm$ signs, and subtracting from it the coefficients of the same row after
shifting this line two places to the right.

For example, from line 6 we can derive the numerator coefficients of the 7th derivative as follows,

line 6 \quad +1 \; -6 \; +15 \; -20 \; +15 \; -6 \; +1

shift 2 places to right \quad +1 \; -6 \; +15 \; -20 \; +15 \; -6 \; +1

subtracting \quad +1 \; -6 \; +14 \; -14 \; 0 \; +14 \; -14 \; +6 \; -1

The final result is

\[ f_k^{(7)} = \frac{1}{2h^7} \left( f_{k+4} - 6f_{k+3} + 14f_{k+2} - 14f_{k+1} + 14f_k - 14f_{k-2} + 6f_{k-3} - f_{k-4} \right) + \text{Reminder of order } h^2 \]

The coefficients are symmetrically distributed about the center term and belong to \( f_{k+i} \), \( i = 0, 1, 2, \ldots, n/2 \) (or \( \frac{n+1}{2} \)).

The error involved in this method is of the order \( ah^2 f_k^{(n+2)} \) where \(|a|<1\).

This method can be reduced into a recursive formula,

\[ f_k^{(n)} = \frac{M_n}{A h^n} \quad (4.4) \]
where

\[ M_k^{2n} = M_{k+1}^{2n-2} - 2M_k^{2n-2} + M_{k-1}^{2n-2} \]  \hspace{1cm} (4.5)

\[ M_k^{2n+1} = M_{k+1}^{2n} - M_k^{2n-1} \]  \hspace{1cm} (4.6)

The even derivatives are determined only from even derivatives of lower order; but to evaluate odd derivatives all lower order \( M' \)'s must be calculated. The method developed enables one to calculate derivatives at all \( k \) if the derivatives at the two end points are given; otherwise, it is impossible with this technique to calculate the first or last \( n \) terms. The following form provides a successive determination of the end points derivatives with the same degree of accuracy, i.e. of order \( h^2 \). Applying (4.1) with \( i = 1, 2, 3, \ldots \), and going through the same procedure, the
derivatives determined will only depend on terms of \( f_k \)
where \( k > 0 \).

For example,

\[
f_o^{(1)} = \frac{1}{2h}(-3f_o + 4f_1 - f_2)
\]

\[
f_o^{(2)} = \frac{1}{2h^2}(4f_o - 10f_1 + 8f_2 - 2f_3)
\]

etc., for any \( n \). The number of terms needed in every
calculation is \( n + 2 \).

By precisely similar analysis the derivatives at the
other end are found. The coefficients are the same except
for an opposite sign and the \( f_k \) are in decreasing order.
The first derivative is,

\[
f_{N+2}^{(1)} = \frac{1}{2h}(3f_{N+2} - 4f_{N+1} + f_N)
\]

where \( N + 2 \) is the end term. The error involved in this
method is again of the same order as before \( ah^2 f^{(n+2)} \) with
\( |a| < 1 \).

The following formulae enable one to calculate the
derivatives at the ends using again the Pascal triangle.

\[ f^{(n)}_o = \frac{1}{2h^n} \sum_{k=0}^{n+1} (-1)^{k+n}(na_{n,k}^f k^k + 2a_{n-1,k}^f k^k) \quad (4.7) \]

and

\[ f^{(n)}_{N+2} = \frac{1}{2h^n} \sum_{k=0}^{n+1} (-1)^{k+n+1}(na_{n,k}^f k^{N+2-k} + 2a_{n-1,k}^f k^{N+2-k}) \quad (4.8) \]

where \( a_{n,k} \) are coefficients taken from Pascal triangle, as shown in Table 4.1. \( n \) refers to the row and \( k \) is the number of the column. These formulae were checked up to \( n = 7 \).

For convenience of computation, (4.7) and (4.8) should be changed to

\[ f^{(n)}_o = \frac{1}{Ah^n} M^n_0 \quad (4.9) \]

and

\[ f^{(n)}_{N+2} = \frac{1}{Ah^n} M^n_{N+2} \quad (4.10) \]
Table 4.1: Pascal triangle to use in (4.7) and (4.8)

<table>
<thead>
<tr>
<th>n</th>
<th>k</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>1</td>
<td>2</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>1</td>
<td>4</td>
<td>6</td>
<td>4</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>1</td>
<td>5</td>
<td>10</td>
<td>10</td>
<td>5</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>1</td>
<td>6</td>
<td>15</td>
<td>20</td>
<td>15</td>
<td>6</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>1</td>
<td>7</td>
<td>21</td>
<td>35</td>
<td>35</td>
<td>23</td>
<td>7</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>1</td>
<td>8</td>
<td>28</td>
<td>56</td>
<td>70</td>
<td>56</td>
<td>28</td>
<td>8</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td>1</td>
<td>9</td>
<td>36</td>
<td>84</td>
<td>126</td>
<td>84</td>
<td>36</td>
<td>9</td>
<td>...</td>
</tr>
</tbody>
</table>
where $A=1$ for $n$ even and $A = 2$ for $n$ odd as in (4.4) and

$$M^n_0 = \frac{A}{2} \sum_{k=0}^{n+1} (-1)^{k+n} (n_1, k + a_{n-1}, k) + a_{n-1}, k f_k$$ (4.11)

$$M^n_{N+2} = \frac{A}{2} \sum_{k=0}^{n+1} (-1)^{k+n+1} (n_1, k + a_{n-1}, k) + a_{n-1}, k f_{N+2-k}$$ (4.12)

where

$$a_{n,k} = a_{n-1,k-1} + a_{n-1,k}$$ (4.13)

4.3. Numerical Evaluation of Derivatives

Whenever a function $f(x)$ is given at discrete points $x = kh$, where $h$ is a constant interval and $k = 0, 1, 2, \ldots, N + 2$, and its derivatives up to order $n$ are required at these points, the method described in 4.2 is suggested. Use of the following procedure may shorten the computation even more.
a. Define \( n, N, h \)
store \( f(kh) \) in the first row of matrix \( M(n+1, N+3) \)

b. Calculate Pascal coefficients applying (4.13) and
store results in matrix \( MM(n,n+2) \).

c. Compute \( M^n_o \) and \( M^n_{N+2} \) using (4.11) and (4.12) and
store results in \( M(n+1,1) \) and \( M(n+1,N+3) \)

d. Compute \( M^n_k \) \((1< k< n+3)\) using (4.5) and (4.6) and
store in \( M(n+1,k) \). In case only even derivatives
are required the program may skip the odd \( n \).

e. Determine \( f_k^{(n)} \) by multiplying each row with \( 1/Ah^n \)
equations (4.4), (4.9), (4.10). A successive pro-
gression from one row to another saves computation
time.

It is advisable to calculate simultaneously an odd and an
even derivative.

This technique derives a derivative using about one
multiplication and 2 or 4 additions per point. The end
points require more multiplication and additions, \( 3(n+2) \)
per derivative.

The results in this approximate computation are strongly
affected by the interval \( h \) and the error goes as \( ah^2 f_k^{(n+2)} \).
Moreover derivatives are determined by taking differences
and multiplying with \(1/h^n\) for the \(n^{th}\) order. The round off errors that appear in the differences \(M_k^n\) values in (4.5), (4.6), (4.11), (4.12) are multiplied by the factor of \(1/h\) for each successive derivative and so decrease the number of accurate digits with every increase in the order. The number of decimal digits of inaccuracy that each consecutive derivative introduced is \(n \times \log_{10}(1/h)\). For example if \(h = .01\), the error is of order \(h^2\) or there are 4 decimal digits of accuracy. In a 12 decimal digit computer, this accuracy will nevertheless disappear by the 5th derivative \((5 \times \log_{10}(1/.01)) = 10\). Generally speaking, for a given number of decimal digits, one must effect a trade between the \(h^2\) accuracy and the \(n \times \log_{10}(1/h)\) error.

4.4. Computer Results:

Table 4.2 shows some results taken from a computer using 12 decimal digit words and the technique of this chapter.
Table 4.2: Derivatives of $e^{-t}$ using several values of $h$. Only 6 significant digits are shown.

<table>
<thead>
<tr>
<th>Derivative</th>
<th>$h=0.1$, $t=2.5$</th>
<th>$h=0.01$, $t=1.25$</th>
<th>$h=0.001$, $t=1.029$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f^{(0)}(t)$</td>
<td>0.0828499</td>
<td>0.778800</td>
<td>0.9714164</td>
</tr>
<tr>
<td>$f^{(1)}(t)$</td>
<td>-0.0822218</td>
<td>-0.778813</td>
<td>-0.9714166</td>
</tr>
<tr>
<td>$f^{(2)}(t)$</td>
<td>0.0821534</td>
<td>0.778807</td>
<td>0.9714165</td>
</tr>
<tr>
<td>$f^{(3)}(t)$</td>
<td>-0.0822904</td>
<td>-0.778819</td>
<td>-0.9714078</td>
</tr>
<tr>
<td>$f^{(4)}(t)$</td>
<td>0.0822219</td>
<td>0.779133</td>
<td>0.1492139</td>
</tr>
<tr>
<td>$f^{(5)}(t)$</td>
<td>-0.0823589</td>
<td>-0.778330</td>
<td></td>
</tr>
<tr>
<td>$f^{(6)}(t)$</td>
<td>0.0822874</td>
<td>-8.405720</td>
<td></td>
</tr>
<tr>
<td>$f^{(7)}(t)$</td>
<td>-0.0824365</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$f^{(8)}(t)$</td>
<td>0.0835825</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$f^{(9)}(t)$</td>
<td>-0.0797228</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
REFERENCES


