PROGRAM II-1. The ATS Function.

```
      FUNCTION ATS(M,A,B,J)
C
C  M IS THE ORDER OF THE SUMMATION, JUST AS IN EQ.[11].
C  A AND B ARE THE TWO INPUT TAYLOR SERIES FOR THE TWO VARIABLES.
C  J IS THE INDEX OF THE ELEMENT OF THE A-ARRAY WHERE THE SUM BEGINS.
C     THIS ALLOWS THE SUM TO BEGIN AT I=J RATHER THAN I=1.
C
      DIMENSION A(1),B(1)
      ATS = 0.
      IF(J.GT.M) RETURN
      MA = M + 1
      DO 100 I=J,M
      L = MA - I
  100 ATS = ATS + A(I)*B(L)
      RETURN
      END
```

In a FORTRAN program using the ATS function, Eq.[11] would be written as

$$F(M) = ATS(M,V,W,1) \quad . \qquad [12]$$

Later in this book, all the expressions for the modified Leibnitz rule will be written in the form of Eq.[12] rather than Eq.[11].

In the case of division operation, use of the modified Leibnitz rule is only slightly more complicated than the above. Consider the quotient f(x) = v(x)/w(x). To resolve this function, we multiply both sides by w(x) and find the product w(x)*f(x) = v(x). Then, we can write the derivatives as follows

$$W(1)*F(m) + \sum_{i=2}^{m} W(i)\, F(m-i+1) = V(m) \quad . \qquad [13]$$

a simple solution for F(m) yields

$$F(m) = \frac{V(m) - \sum_{i=2}^{m} W(i)\, F(m-i+1)}{w(1)} \quad . \qquad [14]$$

Equation[14] can be used to evaluate F(m) because the lower-index terms of F(m-i+1) as well as V(m) and W(m) are known from previous applications of this recursive relation. This is the essence of automatic sequential differentiation. In a FORTRAN program using the ATS function, Eq.[14] would be written as

F(M) = (V(M) - ATS(M,W,F,2))/W(1) . [15]

The above expression shows the reason why the ATS function subprogram is written with the extra index of J, which allows for the sum to begin from some other term than the first term. We will show the importance of the J-index in the discussion on indeterminant forms and L'Hospital rule.

The square root of a function can be differentiated using an approach similar to the division operation. We must recognize the analyzable expression to be f(x)*f(x) = v(x), instead of f(x) = (v(x))**1/2. Then, the m-th derivative can be written as

$$2*F(1)*F(m) + \underset{i=2}{\overset{m-1}{\text{sum.}}}\ F(i)*F(m-i+1) = V(m)\ . \qquad [16]$$

Solving Eq.[16] for F(m), we obtain

$$F(m) = \frac{V(m) - \underset{i=2}{\overset{m-1}{\text{sum.}}}\ F(i)*F(n-i+1)}{2\ F(1)}\ . \qquad [17]$$

Equation[17] can be used to find the reduced derivatives of the square-root function because all the lower-index terms of F(i), F(m-i+1), and V(m) are known from previous applications of this recursive relation. In a FORTRAN program using the ATS function, FORTRAN statements needed to perform Eq.[17] are

```
F(M) = 0.
F(M) = (V(M) - ATS(M,F,F,2))/(2.*F(1))  .                [18]
```

The statement "F(M) = 0." is needed because the sum within the ATS function always runs from the index J to the index m. In this particular case, F(M) is unknown on the right-hand side.

## Simple Implicit Functions

A second important function subprogram for the automatic differentiation of analytic functions is the ATT function. It is developed from the need to differentiate the product of a function and to differentiate a function of another function. Consider the example f(x) = exp(y), where y = sin(ax). From calculus, the first derivative of f(x) is

$$f'(x) = \exp(y)\, y'(x) . \qquad [29]$$

In terms of the reduced derivatives, Eq.[20] is written as

$$F(2) = \exp(y)*Y(2) = F(1)*Y(2) . \qquad [21]$$

From this point, there are two different approaches to the evaluation of the higher-order derivatives. The first approach is to define two new variables

$$G(1):= F(2)/H , \quad \text{and} \quad Z(1):= Y(2)/H . \qquad [22]$$

The only purpose of defining these two variables is to adjust Eq.[21] to have the same form as the modified Leibnitz rule, Eq.[11]. thus,

$$G(1) = F(1)*Z(1) ,$$

$$\text{and} \quad G(M) = ATS(M,F,Z,1) . \qquad [23]$$

Of course, it should be noted that the shift in index from the F-array to the G-array and from the Y-array to the Z-array is carried throughout all the indices of the Taylor series terms

$$G(M) = F(M+1)*M/H , \quad \text{and} \quad Z(M) = Y(M+1)*M/H . \qquad [24]$$

The second approach to the automatic differentiation of this function is to find the higher-order derivatives by a small modification of Leibnitz rule. The derivatives with respect to x of the functions represented by terms in Eq.[21] are found to be

$$2*F(3) = 2*F(1)*Y(3) + F(2)*Y(2) ,$$

$$\text{and} \quad m*F(m+1) = \sum_{i=1}^{m} (m-i+1)*F(i)*Y(m-i+2) . \qquad [25]$$

Equation[25] has been written into a small FORTRAN function subprogram called the ATT function.

PROGRAM II-2. The ATT Function.

---

```
      FUNCTION ATT(M,A,B,J)
C
C   M IS THE ORDER OF THE SUMMATION.
C   A AND B ARE THE TWO INPUT TAYLOR SERIES FOR THE THE VARIABLES.
C    J IS THE INDEX OF THE ELEMENT OF THE A-ARRAY WHERE THE SUM BEGINS.
C      THIS ALLOWS THE SUM TO BEGIN AT I=J RATHER THAN I=1.
C
      DIMENSION A(1),B(1)
      ATT = 0.
      IF(J.GT.M) RETURN
      MA = M + 2
      DO 100 I=J,M
      L = MA - I
      AL = L - 1
  100 ATT = ATT + A(I)*B(L)*AL
      ATT = ATT/M
      RETURN
      END
```

---

the term "*AL" in the subprogram above is "*(m-i+1)" in Eq.[25]. The division by "M" just before "RETURN" in the subprogram is required by the "*M" on the lefthand side of Eq.[25]. In a FORTRAN program, Eq.[25] would be written as

```
      MA = M + 1
      F(MA) = ATT(M,F,Y,1) .
```
[26]

These function subprograms, ATS and ATT, are the foundation stones for the automatic sequential differentiation of all analytic functions. The reader will find in Appendix A all the recursive formulae for all commonly used analytic functions. The formulae are written in FORTRAN.

```
      SUBROUTINE EXPY(X,H,A,F)
      DIMENSION F(100),Y(100)
C
C     X = INITIAL VALUE
C     H = INCREMENT
C     A = COEFFICIENT
C
      ASQ = A*A
      HSQ = H*H
      Y(1) = SIN(A*X)
      F(1) = EXP(Y(1))
      Y(2) = A*H*COS(A*X)
      F(2) = F(1)*Y(2)
      DO 100 I=2,99
      IA = I + 1
      IZ = I - 1
      Y(IA) = - Y(IZ)*ASQ*HSQ/(I*IZ)
  100 F(IA) = ATT(I,F,Y,1)
      RETURN
      END
```

The above is a short subroutine for evaluating the Taylor series terms for the function f(x) = exp(y), with y = sin(ax). This small FORTRAN subprogram can be inserted into any program where all the Taylor terms of this particular function is desired.

Next, we will analyze the automatic differentiation of the function f(x) = ln(y), where y = sin(ax). The first derivative of this function is

$$f'(x) = \frac{y'}{y} . \qquad [27]$$

In terms of reduced derivatives, Eq.[27] becomes

F(2) = Y(2)/Y(1) , or Y(1)*F(2) = Y(2) . [28]

The higher-order derivatives are then found to be

2*Y(1)*F(3) + Y(2)*F(2) = 2*Y(3) ,

and $m*Y(1)*F(m+1) + \underset{i=2}{\overset{m}{\text{sum.}}} (m-i+1)*Y(I)*F(m-i+2) = m*Y(m+1)$ . [29]

Solving Eq.[29] for f(m+1), we have

$$F(m+1) = \frac{m*Y(m+1) - \underset{i=2}{\overset{m}{\text{sum.}}} (m-i+1)*Y(i)*F(m-i+2)}{Y(1)*m} . \qquad [30]$$

```
      SUBROUTINE LNY(X,H,A,F)
      DIMENSION F(100),Y(100)
C
C     X = INITIAL VALUE
C     H = INCREMENT
C     A = COEFFICIENT
C
      ASQ = A*A
      HSQ = H*H
      Y(1) = SIN(A*X)
      F(1) = ALOG(Y(1))
      Y(2) = A*H*COS(A*X)
      F(2) = Y(2)/Y(1)
      DO 100 I=2,99
      IA = I + 1
      IZ = I - 1
      Y(IA) = - Y(IZ)*A*A*H*H/(I*IZ)
  100 F(IA) = (Y(IA) - ATT(I,Y,F,2))/Y(1)
      RETURN
      END
```

A small segment of a FORTRAN program is given above for the evaluation of the Taylor series terms of this function. Note that the ATT-function index j begins at 2 in this example just as it did in the square root of a function. Thus, the index j does need to be other than one at times. We will examine this more closely in the discussion on indeterminacies in a later section.

As a final example in this section on the automatic differentiation of simple functions and operations, consider the function f(x) = x**y, with y = sin(ax). The first derivative of this function is given by

$$f'(x) = y*x^{y-1} + x^{y}\ \ln(x)\ \frac{dy}{dx} = f(x)*\left\{\frac{y}{x} + \ln(x)*y'\right\} . \qquad [31]$$

In terms of reduced derivatives, Eq.[31] becomes

$$F(2) = F(1)*U(2) ,$$

$$\text{where} \quad U(2) := V(1)*W(2) + V(2)*W(1) . \qquad [32]$$

The W-function in Eq.[32] is the natural logarithm function, w(x)=ln(x), and its recursive relation was given in Eq.[9] above. This recognition is important in this analysis. In the ATS method, since one is concerned with recursive relations, it is important to recognize other recursive relations within the larger problems. The higher-order terms for f(x) are found to be

$$m*F(m+1) = \sum_{i=1}^{m} (m-i+1)*F(i)*u(m-i+2) ,$$

$$\text{and} \quad U(m) = \sum_{i=1}^{m} V(i)*W(m-i+1) . \qquad [33]$$

The FORTRAN program for the evaluation of the Taylor series terms of the function f(x) = x**y, where y = sin(ax) is given in Program II-3.

This completes the discussion concerning the automatic sequential differentiation of simple functions and operations. The reader should look at Appendix A for a complete list of recursive relations (written in FORTRAN) for all the commonly used analytic functions and some difficult functions.

PROGRAM II-3. Differentiation of $f(x) = x^{\sin(ax)}$.

```
      SUBROUTINE XPSINX(X,H,A,F)
      DIMENSION Y(101),F(100),U(100),Z(101)
C
C     X = INITIAL VALUE
C     H = INCREMENT
C     A = COEFFICIENT
C
      ASQ = A*A
      HSQ = H*H
      Y(1) = SIN(A*X)
      Z(1) = ALOG(X)
      F(1) = X**Y(1)
      Y(2) = A*H*COS(A*X)
      Z(2) = H/X
      DO 100 I=2,99
      IA = I + 1
      IZ = I - 1
      U(IZ) = ATS(IZ,Y,Z,1)
      F(I) = ATT(IZ,F,U,1)
      Y(IA) = - ASQ*HSQ*Y(IZ)/(I*IZ)
  100 Z(IA) = - H*IZ*Z(I)/(X*I)
      RETURN
      END
```

## B. Sequential Differentiation of Complicated Functions.

We begin this section by analyzing the differentiation of the Bessel function of the first kind, $J_n(x)$. The automatic differentiation of the higher transcendental functions (such as the Bessel function) is obtained through the differential equation that generates the particular function. Bessel's equation is

$$x^2 * \frac{d^2 f}{dx^2} + x*\frac{df}{dx} + (x^2 - n^2)*f = 0 . \qquad [34]$$

There are at least two different ways to obtain the Taylor series terms for the Bessel function. The first method uses the introduction of many auxiliary functions and solves the problem by brute force, called the direct method. The second method involves the hand-differentiation of Bessel's equation up to the fourth derivative of y with respect to x. This indirect method yields a recursive relation that requires both less storage space and computation time than the direct method. Both methods will be discussed in turn.

Bessel Function (Direct Method)

We define three auxiliary variables

$$z(x):=x^2\ ,\quad v(x):=\frac{df}{dx}\ ,\quad \text{and}\quad w(x):=\frac{d^2f}{dx^2}\ . \qquad [35]$$

The next step is to find the recursive relations for each of these functions as well as the recursive relations for x and f. The recursive relation for x is given by

$$X(1) = x\ ,\quad X(2) = h\ ,\quad \text{and}\ X(i) = 0\ ,\quad \text{for}\ i > 2\ . \qquad [36]$$

The recursive relation for z(x) = x**2 is given by

$$Z(1) = x^{**}2\ ,\quad Z(2) = 2^*x^*h\ ,\quad Z(3) = h^{**}2\ ,$$

$$\text{and}\qquad Z(i) = 0\ ,\quad \text{for}\ i > 3\ . \qquad [37]$$

The V-array and W-array are related to the F-array by the relations given below.

$$V(i) = F(i+1)*\frac{i}{h}\ ,\ \text{and}\quad W(i) = F(i+2)*\frac{i*(i+1)}{h*h}\ . \qquad [38]$$

Equation[38] shows that the two auxiliary functions v and w have arrays that are simply shifted by one and two indices, respectively from the F-array. The factors i and (i+1) must be included because the factorials are included in the reduced derivatives, so are the factors h and h*h.

In terms of reduced derivatives, Eq.[34] becomes

$$Z(1)*W(1) + X(1)*V(1) + Z(1)*F(1) - n^2*F(1) = 0\ . \qquad [39]$$

The k-th derivative of the expression represented by Eq.[39] is given by

$$Z(1)*W(k) + \sum_{i=2}^{k} Z(i)*W(k-i+1) + \sum_{i=1}^{k} X(i)*V(k-i+1) +$$

$$+ \sum_{i=1}^{k} Z(i)*F(k-i+1) - n^2*F(k) = 0\ . \qquad [40]$$

Equation[40] can be solved for the unknown W(k) as

W(K) = (F(K)*N*N - ATS(K,Z,W,2) - ATS(K,X,V,1) -

- ATS(K,Z,F,1))/Z(1) . [41]

PROGRAM II-4. Bessel Function Derivatives (Direct Method).

```
      SUBROUTINE BESJD(N,X0,F0,DF,H,F)
      DIMENSION X(100),F(100),Z(100),V(100),W(100)
      REAL N
      DATA Z/100*0./
C
C     X0 = INITIAL VALUE FOR X.
C     F0 = INITIAL VALUE FOR THE BESSEL FUNCTION AT X0.
C     DF = INITIAL SLOPE FOR THE BESSEL FUNCTION AT X0.
C     N = ORDER OF THE BESSEL FUNCTION.
C     H = INCREMENT
C
      NSQ = N*N
      HSQ = H*H
      X(1) = X0
      F(1) = F0
      X(2) = H
      F(2) = H*DF
      Z(1) = X(1)*X(1)
      Z(2) = 2.*H*X(1)
      Z(3) = HSQ
C
C  THE ABOVE ARE NECESSARY INITIALIZATION STATEMENTS.
C
      V(1) = F(2)/H
      DO 100 K=3,100
      KZ = K - 1
      KY = K - 2
      X(K) = 0.
      W(KY) = (F(KY)*NSQ - ATS(KY,Z,W,2) - ATS(KY,X,V,1) -
     A  ATS(KY,Z,F,1))/Z(1)
C
C  THE ABOVE STATEMENT IS EQ.[41].
C
      V(KZ) = W(KY)*H/KY
  100 F(K) = V(KZ)*H/KZ
C
C  THE F-ARRAY CONTAINS THE FIRST ONE HUNDRED TERMS OF THE TAYLOR SERIES
C    FOR THE BESSEL FUNCTION.
C
      RETURN
      END
```

Program II-4 is the FORTRAN program for the evaluation of the Taylor series terms for the Bessel function according to Eq.[41]. Since this recursive relation was derived from a second-order differential equation, two initial values are needed. They are the values of f and df/dx at x=x0.

Bessel Function (Indirect Method)

We will next discuss the indirect method for the same task. This is more efficient. This solution of Bessel's equation requires more work on the part of the analyst/programmer, but the result is efficient in computer storage space and computation time. We rewrite Bessel's equation in terms of reduced derivatives with z(x) given in Eq.[37]. The result is

$$Z(1)*F(3)*\frac{2*1}{h^2} + X(1)*F(2)*\frac{1}{h} + Z(1)*F(1) - N^2*F(1) = 0 . \qquad [42]$$

The factors "(2*1)/h**2" and "1/h" in the first two terms are needed because the reduced derivatives contain factorial functions and powers of h. We differentiate, repeatedly with respect to x, functions represented by the terms in Eq.[42]. After the second differentiation, the resulting expression becomes the general recursive relation because Z(4) is zero. Then, we can solve for phe unknown in the general recursive relation and complete the analysis.

The unknown in Eq.[42] is F(3). The solution of which is

$$F(3) = - F(2)\ \frac{h}{2x} - F(1)\ \frac{h^2}{x^2}\ (x^2 - n^2) . \qquad [43]$$

Next, the first derivative with respect to x of the functions represented by the terms in Eq.[42] yields.

$$Z(1)*F(4)\frac{3*2}{h^2} + Z(2)*F(3)\frac{2*1}{h^2} + X(1)*F(3)\frac{2}{h} + X(2)*F(2)\frac{1}{h} +$$

$$+ Z(1)*F(2) + Z(2)*F(1) - n^2*F(2) = 0 . \qquad [44]$$

Solving Eq.[44] for F(4), we obtain

$$F(4) = - F(3)\frac{h}{x} - F(2)\frac{h^2}{6x^2}(x^2+1-n^2) - F(1)\frac{h^3}{3x} . \qquad [45]$$

The second derivatives with respect to x of the functions represented by the terms in Eq.[42] yields the expression

$$Z(1)*F(5)\frac{4*3}{h^2} + Z(2)*F(4)\frac{3*2}{h^2} + Z(3)*F(3)\frac{2*1}{h^2} + X(1)*F(4)\frac{3}{h} +$$

$$+ X(2)*F(3)\frac{2}{h} + Z(1)*F(3) + Z(2)*F(2) +$$

$$+ Z(3)*F(1) - n^2*F(3) = 0 . \qquad [46]$$

Solving Eq.[46] for F(5), we obtain

$$F(5) = - F(4)\frac{5h}{4x} - F(3)\frac{h^2}{12x^2}(x^2+4-n^2) - F(2)\frac{2h^3}{12x} - F(1)\frac{h^4}{12x^2} . \qquad [47]$$

Equation[47] is the general recursive relation for the Bessel function. When written for the k-th Taylor series term, Eq.[47] becomes

$$F(k) = - F(k-1)\frac{(2k-5)h}{(k-1)x} - F(k-2)\frac{h^2}{(k-1)(k-2)x^2}(x^2+k-3-n^2) -$$

$$- F(k-3)\frac{2\ h^3}{(k-1)(k-2)x} - F(k-4)\frac{h^4}{(k-1)(k-2)x^2} , \quad \text{for } k > 4. \qquad [48]$$

The case x=0 is particularly simple and will be discussed later in the section on L'Hospital rule.

Program II-5 is the FORTRAN program for the evaluation of the Taylor series terms for the Bessel function according to Eq.[48]. The Bessel function of a dependent variable, $J_n(y)$ with y=y(x), will be analyzed in the section on implicit complex functions.

PROGRAM II-5. Bessel Function Derivatives (Indirect Method).

```
      SUBROUTINE BESJIM(N,X0,F0,DF,H,F)
      DIMENSION F(100)
      REAL N
C
C     X0 = INITIAL VALUE FOR X.
C     F0 = INITIAL VALUE FOR THE BESSEL FUNCTION AT X0.
C     DF = INITIAL SLOPE FOR THE BESSEL FUNCTION AT X0.
C     N = ORDER OF THE BESSEL FUNCTION.
C     H = INCREMENT
C
      NSQ = N*N
      HSQ = H*H
      A = H/X0
      B = H*A/X0
      C = 2.*HSQ*A
      D = HSQ*ASQ
      G = X0*X0 - NSQ - 6.
      F(1) = F0
      F(2) = H*DF
C
C  THE ABOVE ARE NECESSARY INITIALIZATION STATEMENTS.
C
      F(3) = - F(2)*2.*A - F(1)*B*(G+6.)/2.
C
C  THE ABOVE STATEMENT IS EQ.[43].
C
      F(4) = - F(3)*A - F(2)*B*(G*7.)/6. - F(1)*C/6.
C
C  THE ABOVE STATEMENT IS EQ.[45].
C
      DO 100 K=5,100
      KZ = K - 1
      KY = K - 2
      KX = K - 3
      KW = K - 4
      PK = 2*K - 5
      GK = 2*K + G
      RK = KZ*KY
  100 F(K) = - (F(KZ)*PK*A*KC + F(KY)*B*GK + F(KX)*C + F(KW)*D)/RK
C
C  THE ABOVE STATEMENT IS EQ.[48].
C
C  THE F-ARRAY CONTAINS THE FIRST ONE HUNDRED TERMS OF THE TAYLOR SERIES
C    FOR THE BESSEL FUNCTION.
C
      RETURN
      END
```

## The Error Function, erf(x)

The Error function is perhaps one of the better known "higher transcendental" functions. It is defined as a definite integral

$$\mathrm{erf}(x) = \frac{2}{pi^{1/2}} \int_0^x \exp(-t^2)\, dt = \frac{2}{pi^{1/2}} \int_0^x f(t)\, dt\ , \qquad [49]$$

where f(t) = exp(-t**2), and pi = 3.1416... . The sequential differentiation of erf(x) caf be obtained by the following analysis. Define

$$g(t) := \int f(t)\, dt .$$

Then the Error function can be written as

$$\mathrm{erf}(x) = \frac{2}{pi^{1/2}} \{g(x) - g(0)\} . \qquad [50]$$

Next, we expand a Taylor series for g(t) about the point t=0 with an increment of x. This is unconventional but necessary in order to obtain g(x) below.

$$g(x) = g(0) + \frac{d\, g(0)}{dt} * x + \frac{d^2 g(0)}{dt^2} \frac{x^2}{2!} + \frac{d^3 g(0)}{dt^3} \frac{x^3}{3!} + \ldots . \qquad [51]$$

Hence,

$$g(x) - g(0) = \sum_{i=2}^{00} G_t(i) , \qquad [52]$$

with x being the increment in t. The subscript "t" on the G-array is to indicate that the reduced derivatives are taken with respect to t. This is necessary because we have in this analysis reduced derivatives taken both with respect to x and with respect to t. Continuing, erf(x) is

$$\mathrm{erf}(x) = \frac{2}{pi^{1/2}} \sum_{i=2}^{00} G_t(i) . \qquad [53]$$

Since g(t) is the integral of f(t), the reduced derivatives with respect to t of g(t) are given in terms of reduced derivatives with respect to t of f(t) in a simple manner.

$$G_t(2) = F_t(1)\frac{x}{1} , \quad G_t(n) = F_t(n-1)\frac{x}{n-1} . \qquad [54]$$

In terms of the reduced derivatives of f(t) with respect to t,

$$\mathrm{erf}(x) = \frac{2}{pi^{1/2}} \sum_{i=1}^{00} F_t(i)\frac{x}{i} . \qquad [55]$$

In the above development, we have ignored convergence questions because the error function has an essential singularity at infinity. The meaning of this fact will be made clear in Chapter III.

The first derivative of erf(x) with respect to x (using Eq.[54]) is given by

$$\frac{d}{dx}\,\mathrm{erf}(x) = C \sum_{i=2}^{00} \frac{d}{dx} G_t(i) = C \sum_{i=1}^{00} F_t(i) = C*f(x) \ . \qquad [56]$$

where C = 2/pi**(1/2). Then, the reduced derivatives with respect to x of erf(x) are given by

$$ERF_x(2) = C*F_x(1)\frac{h}{1} \ , \quad ERF_x(m) = C*F_x(m-1)\frac{h}{m-1} \ . \qquad [57]$$

We have reduced the complicated problem of evaluating the derivatives of erf(x) to the much simpler problem of evaluating the derivatives of the exponential functions, exp(-x**2) and exp(-t**2).

In order to evaluate the derivatives of the Error function, we define an auxiliary function, v(x):= -x**2 also v(t):= -t**2. Then

$$V(1) = -a^2, \quad V(2) = -2a*h \ , \quad V(3) = -h^2 \ , \text{ and } \ V(4) = 0 \ . \qquad [58]$$

where v(x) is being expanded with respect to x about the point x=a with increment h. For the evaluation of the F-array in Eq.[55], we need the reduced derivatives of v(t) with respect to t at the point t=0 with an increment of a. They are

$$V(1) = 0 \ , \quad V(2) = 0 \ , \quad V(3) = -a^2 \ , \text{ and } \ V(4) = 0 \ . \qquad [59]$$

The sequential differentiation of the function f(x) = exp(v) has been analyzed with the result given in Eq.[25] above. However, since V(4) = 0 as well as all higher-index terms of the V-array, we can write

$$F(1) = \exp(V(1)) \ , \quad F(2) = F(1)*V(2) \ ,$$

and $\quad F(m)*(m-1) = F(m-2)*V(3)*2 + F(m-1)*V(2)*1 \ , \quad \text{for } m > 1 \ . \qquad [60]$

Equation[60] can be used to find the reduced derivatives of both f(x) and f(t). Those of f(x) are needed in Eq.[57] and those of f(t) are needed in Eq.[55]. The reduced derivatives of v(x) with respect to x and of v(t) with respect to t are given in Eq.[58] and Eq.[59], respectively.

PROGRAM II-6. Differentiation of the Error Function, erf(x).

```
      SUBROUTINE ERFX(X0,H,ERF)
      DIMENSION ERF(100),F(100)
C
C     X0 = INITIAL VALUE
C     H = INCREMENT
      C = 2./SQRT(3.1416XXXXXXXXXX)
C
C  USE EQ.[55] TO EVALUATE THE ERROR FUNCTION.  FOR THIS EVALUATION, THE
C    ONLY NON-ZERO ELEMENT IN THE V-SERIES IS V(3)
C
      V3 = - X0*X0
      F(1) = 1.
      ERF(1) = F(1)*X0
      F(2) = 0.
      ERF(1) = ERF(1) + F(2)*X0/2.
      DO 100 I=3,100
      ZI = I - 1
      IY = I - 2
      F(I) = F(IY)*V3*2./ZI
  100 ERF(1) = ERF(1) + F(I)*X0/I
C
C  THE ABOVE PORTION OF THE PROGRAM IS THE IMPLEMENTATION OF EQ.[55].
C  ALTHOUGH THE EVEN-TERMS OF THE F-SERIES ARE ZERO, THE PROGRAM IS
C    WRITTEN IN FULL FOR THE SAKE OF CLARITY, AND UNDERSTANDING.
C
C  BEGIN THE EVALUATION OF THE DERIVATIVES OF THE ERROR FUNCTION,
C    USING EQ.[57] IN CONJUNCTION WITH EQ.[58] AND EQ.[60].
C
      V1 = - X0*X0
      V2 = - 2.*X0*H
      V3 = - H*H
      F(1) = EXP(V1)
      ERF(2) = C*F(1)*H
      F(2) = F(1)*V2
      DO 200 I=3,99
      IZ = I - 1
      IY = I - 2
      F(I) = (F(IY)*V3*2. + F(IZ)*V2)/IZ
  200 ERF(I) = C*F(IZ)*H/IZ
C
C  THE ERF-ARRAY CONTAINS THE FIRST ONE HUNDRED TERMS OF THE TAYLOR
C    SERIES FOR THE ERROR FUNCTION, ERF(X).
C
      RETURN
      END
```

Program II-6 can be used to find all the Taylor series terms for the error function, erf(x). Sequential differentiation of the error function is obtained from Eq.[57] in conjunction with Eq.[58] and Eq.[60].

The Error Function, erf(y)

The analysis of the Error function of a dependent variable is relatively simple, so it is included in this section. Program II-7 is the FORTRAN implementation of Eq.[62] below for the sequential differentiation of the Error function erf(y) with y = sin(ax). This program is only slightly more complicated than Program II-6.

PROGRAM II-7. The Error Function, erf(y) with y = sin(ax).

```
      SUBROUTINE ERFY(X0,H,A,ERF)
      DIMENSION ERF(100),F(100),V(100),Y(100)
C
C     X0 = INITIAL VALUE
C     H = INCREMENT
C     A = COEFFICIENT
C
      HSQ = H*H
      ASQ = A*A
      C = 2./SQRT(3.1416XXXXXXXXXX)
C
C  USE EQ.[55] TO EVALUATE THE ERROR FUNCTION.  FOR THIS EVALUATION, THE
C    ONLY NON-ZERO ELEMENT IN THE V-SERIES IS V(3)
C
      Y(1) = SIN(A*X0)
      V3 = - Y(1)*Y(1)
      F(1) = 1.
      ERF(1) = Y(1)
      DO 100 I=3,100,2
      ZI = I - 1
      IY = I - 2
      F(I) = F(IY)*V3*2./ZI
  100 ERF(1) = ERF(1) + F(I)*Y(1)/I
C
C  BEGIN THE EVALUATION OF THE DERIVATIVES OF THE ERROR FUNCTION,
C    USING EQ.[62] WITH EQ.[63] AND EQ.[64].
C
      V(1) = - Y(1)*Y(1)
      F(1) = EXP(V(1))
      Y(2) = A*H*COS(A*X0)
      DO 200 I=2,99
      IA = I + 1
      IZ = I - 1
      V(I) = - ATS(I,Y,Y,1)
      Y(IA) = ASQ*HSQ*Y(IZ)/(I*IZ)
      F(I) = ATT(IZ,F,V,1)
  200 ERF(I) = C*ATT(IZ,F,Y,1)
C
C  THE ERF-ARRAY CONTAINS THE FIRST ONE HUNDRED TERMS OF THE TAYLOR
C    SERIES FOR THE ERROR FUNCTION, ERF(Y) WITH Y = SIN(AX).
C
      RETURN
      END
```

Given erf(y) with $y = y(x)$, the equations from Eq.[49] to Eq.[55] remain valid with the replacement of all the X's by Y's. But, Eq.[56] is replaced by

$$\frac{d}{dx}\,\mathrm{erf}(y) = \frac{d}{dy}\,\mathrm{erf}(y)*\frac{dy}{dx} = C*f(y)*y' \ . \qquad [61]$$

In terms of reduced derivatives Eq.[61] is

$$ERF_x(2) = C*F_x(1)*Y_x(2) \ , \quad ERF_x(M+1) = C*ATT(M,F,Y,1) \ . \qquad [62]$$

Equation[62] is to replace Eq.[57]. The auxiliary function v(y):= -y**2 is sequentially differentiated with respect to x as follows.

$$V(1) = - Y(1)*Y(1) \ , \quad \text{and} \quad V(M) = - ATS(M,Y,Y,1) \ . \qquad [63]$$

Equation[63] is to replace Eq.[58]. The sequential differentiation of the function f(x) = exp(v) was given in Eq.[25].

$$F(1) = EXP(V(1)) \ , \quad F(2) = F(1)*V(2) \ ,$$

$$\text{and} \quad F(M+1) = ATT(M,F,V,1) \ . \qquad [64]$$

Equation[64] is to replace Eq.[60].

This completes the discussion of the analysis for the differentiation of the Error function.

## The Sine-Integral

The Sine integral is discussed here because it is a well known higher transcendental function that has been widely studied and tabulated. It is a definite integral

$$si(x) = \int_0^x \frac{\sin(p)}{p}\,dp = \int_0^x s(p)\,dp \ , \qquad [65]$$

where p is the dummy integration variable and s(p) = sin(p)/p. Just as in the case with the Error function, we define

$$r(p) := \int s(p)\,dp \ .$$

Then, the Sine integral can be written as

$$si(x) = r(x) - r(0) . \qquad [66]$$

Next, we expand a Taylor series for r(p) about the point p=0 with an increment of x. This is unconventional but necessary for obtaining

$$r(x) = r(0) + \frac{d\,r(0)}{dp}*x + \frac{d^2 r(0)}{dp^2}*\frac{x^2}{2!} + \frac{d^3 r(0)}{dp^3}*\frac{x^3}{3!} + \ldots . \qquad [67]$$

Hence,

$$r(x) - r(0) = \sum_{i=2}^{00} R_p(i) , \qquad [68]$$

with x being the increment in p. The subscript "p" on the R-array is to indicate differentiation with respect to p. The Sine integral is

$$si(x) = \sum_{i=2}^{00} R_p(i) . \qquad [69]$$

Since r(p) is the integral of s(p), the reduced derivatives with respect to p of r(t) are given in terms of reduced derivatives with respect to p of s(t) by

$$R_p(2) = S_p(1)\frac{x}{1} , \quad R_p(n) = S_p(n-1)\frac{x}{n-1} . \qquad [70]$$

In terms of reduced derivatives of s(t) with respect to p,

$$si(x) = \sum_{i=1}^{00} S_p(i)\frac{x}{i} . \qquad [71]$$

We have ignored convergence questions here because the Sine integral has an essential singularity at infinity.

The first derivative of si(x) with respect to X (using Eq.[70]) is given by

$$\frac{d}{dx} si(x) = \sum_{i=2}^{00} \frac{d}{dx} R_p(i) = \sum_{i=1}^{00} S_p(i) = s(x) . \qquad [72]$$

Then, the reduced derivatives with respect to x of si(x) are given by

$$SI(2) = S_x(1)\frac{h}{1} , \quad \text{and} \quad SI_x(m) = S_x(m-1)\frac{h}{m-1} . \qquad [73]$$

Equation[73] is similar to Eq.[57] for the Error function. To facilitate the evaluation of the reduced derivatives of the Sine integral, we define an auxiliary function w(x) = sin(x). Then, the reduced derivatives of w(x) with respect to x at x=a with increment h are

$$W(1) = \sin(a) , \quad W(2) = \cos(a)*h,$$

$$\text{and} \quad W(m) = -\frac{W(m-2)*h^2}{(m-1)(m-2)} , \qquad \text{for } n > 2. \quad [74]$$

For the evaluation of S's in Eq.[71], we need the reduced derivatives of w(p) = sin(p) with respect to p at p=0 with increment a. They are

$$W(1) = 0 , \quad W(2) = a ,$$

$$\text{and} \quad W(m) = -\frac{W(m-2)*a^2}{(m-1)(m-2)} , \qquad \text{for } m > 2. \quad [75]$$

The reduced derivatives of s(x) with respect to x at x=a are

$$S(1) = \frac{W(1)}{a} , \quad S(2) = \frac{W(2) - S(1)*h}{a} ,$$

$$\text{and} \quad S(m) = \frac{W(m) - S(m-1)*h}{a} , \qquad \text{for } m > 1. \quad [76]$$

The reduced derivatives of s(p) with respect to p are somewhat difficult to evaluate. The function is given by

$$s(p) = w(p)/p , \text{ or } s(p)*p = w(p) . \qquad [77]$$

Since the reduced derivatives are to be evaluated at p=0, this expression is meaningless as it stands. The derivatives with respect to p of the functions in Eq.[77] lead to

$$S_p(2)*p + S_p(1)*a = W_p(2) , \qquad [78]$$