Handbook
of

INTEGRALS RELATED TO

HEAT CONDUCTION AND DIFFUSION

by

Donald E. Amos

December, 2006

ACKNOWLEDGEMENTS

This work was funded in part by

Beck Engineering Consultants Company
1935 Danbury West
Okemos, MI 48864-1873

for

Sandia National Laboratories
Purchase Order No. 30629
Kevin Dowding, Project Manager

Many of the integrals addressed in this document came from work being done by Dr. James V. Beck and associates. Dr. Beck's enthusiasm and support is greatly appreciated. Credit is due Jill Bielawski for her excellent job of mathematical typing.

PREFACE

This handbook of integrals grew out of a research project carried out by Beck Engineering Consultants Company. The research, to develop computer codes which compute solutions of the transient heat conduction equation to high accuracy, was funded by Sandia National Laboratories. These codes were subsequently used in code verification work to assess accuracies of more complex numeric codes. The integrals developed in this presentation were suggested by Dr. James V. Beck and associates during the course of the research from 2000 to 2005.

The work of Beck Engineering centered on the linear heat equation with constant coefficients in mostly rectangular geometries in all three dimensions. While many classical problems have been solved analytically, the solutions can achieve high accuracies with a reasonable amount of computer work only over very limited ranges of variables, especially in the time variable. In these cases, the classical solutions are, for the most part, only computable for the small and large times. However, newer, more flexible methods, which were able to fill-in the intermediate times, were applied to achieve (absolute) accuracies on the order of 10^{-10}.

The research in this volume was directed toward numerical evaluation of diffusion related integrals in subroutines which would return accurate answers over stated ranges of variables. In keeping with the goal of high accuracies, all subroutines were developed in double precision arithmetic. For the IBM PC and similar machines, this means about 16 digits. In many fundamental integrals, much effort was devoted to achieving relative accuracy (significant digits) rather than absolute accuracy (decimal places). Since it is difficult to compute to full precision and avoid all losses of significance, most subroutines returned overall accuracies of $O(10^{-13})$ or better over common ranges of variables. This specification allows several orders of magnitude latitude in the application to problems where only accuracies of 10^{-10} are required.

This handbook is presented in four parts. Chapter 1 presents a Table of Integrals with references to Chapter 2 where the main formulae are presented in handbook format. Formulae presented in Chapter 2 reference Chapter 3 where the derivations are presented in full detail in sub-sections called Folders. The Table of Contents of Chapter 3 lists the titles of 29 Folders along with a brief summary of the results of each Folder. Chapter 4 is devoted to the description of files containing FORTRAN codes used to test the formulae of Chapter 3 numerically. These FORTRAN codes are formatted as text files on a disk in a pocket at the end of this handbook.

References are often designated by A&S, EMOT or Beck et al. followed by possibly a volume number, page number, and equation number. A&S refers to the NBS Handbook of Mathematical Functions, also known as AMS 55, edited by Abramowitz and Stegun. EMOT refers to a five volume series called the Bateman Project edited by Erdeyi, Magnus, Oberhettinger and Tricomi. Three volumes titled Higher Transcendental Functions (HTF) are in handbook format for many of the functions of mathematical physics. Two other volumes are titled Tables of Integral Transforms (TIF). Beck et al. refers to Beck's book: Heat Conduction Using Green's Functions, Hemisphere Publishing Corp., 1992.

Updates and Changes to the August 2003 Edition

The previous edition of this work was dated August, 2003. Several changes and additions were made in the previous edition to make the presentation consistent and fill in gaps in the Table of Integrals and corresponding computer programs.

Change in Notation for I_1

In the initial stages of the research, $I_1(a,b,t)$, defined by

$$I_1(a,b,t)=\int_0^t e^{-a^2/\tau}\frac{erf(b/\sqrt{\tau})}{\sqrt{\tau}}d\tau$$

was investigated first. In subsequent investigations into other integrals with similar forms involving $\sqrt{\tau}$, it became apparent that the substitution $w=1/\sqrt{\tau}$ gave analytic integrands and were the preferred form for manipulation. *In the August 2003 edition of this work,* $I_1(a,b,t)$ was documented in Folders 1and 2 and the analytic form, documented in Folder 10, was denoted by

$$\bar{I}_1(a,b,T)=\int_T^\infty e^{-a^2w^2}\frac{erf(bw)}{w^2}dw=\frac{1}{2}I_1(a,b,t),\qquad T=1/\sqrt{t}\,,$$

and similarly for $\bar{I}_1^c(a,b,T)$. However the subroutine INTEGI1 always returned $\bar{I}_1(a,b,T)$ or $\bar{I}_1^c(a,b,T)$ (on the selection parameter KODE=1 or 2) and one had to multiply by 2 to get $I_1(a,b,t)$ or $I_1^c(a,b,T)$. ***In this edition***, we define

$$I_1(a,b,T)=\int_T^\infty e^{-a^2w^2}\frac{erf(bw)}{w^2}dw=\frac{1}{2}\int_0^t e^{-a^2/\tau}\frac{erf(b/\sqrt{\tau})}{\sqrt{\tau}}d\tau,\qquad T=1/\sqrt{t},$$

$$I_1^c(a,b,T)=\int_T^\infty e^{-a^2w^2}\frac{erfc(bw)}{w^2}dw=\frac{1}{2}\int_0^t e^{-a^2/\tau}\frac{erfc(b/\sqrt{\tau})}{\sqrt{\tau}}d\tau,\qquad T=1/\sqrt{t},$$

which conforms to the pattern developed for other integrals of similar form. The code for INTEGI1 was not changed in any way. INTEGI1 still returns values from the analytic forms which, in this edition, are denoted by $I_1(a,b,T)$ or $I_1^c(a,b,T)$ on KODE=1 or 2. One still has to multiply the results from INTEGI1 by 2 to get the alternate integral forms in terms of τ on (0,t). $\bar{I}_1(a,b,T)$ and $\bar{I}_1^c(a,b,T)$ do not exist in this edition.

Changes and Additions to Computer Programs

The table below presents the changes and additions to the computer codes that implement the formulae developed in Chapter 3. Any changes i*n the August 2003 edition* not listed below were transparent and did not change the way the code was used. Except for the addition of new subroutines and programs and a couple of other minor changes, the addition of the selection parameter KODE to the call list was made to the existing subroutines INTEGI3, INTEGI6, INTEGP, INTEGI13 and INTEGI14 (formerly INTEG14). KODE=1 designates the original integral using the error function erf(*) while KODE=2 designates the new addition for the co-error function integral using erfc(*). Each subroutine or program has a prologue which describes the function (or functions) being computed and should be consulted when updating call lists.

It is apparent from the changes discussed above that there is a compatibility issue if one mixes routines from the two editions. Calls based on the 2003 edition mixed with the new library (AMOSSUBS.FOR + BECKSUBS.FOR) from this edition need to be checked for compatibility. Similar considerations apply for the research codes in RESEARCH.FOR. In order to make this check easier, the following table documents the non-transparent changes. This table will tell a user whether the calls based on the 2003 edition will be compatible with the new library. If a user program contains a call to one of the routines listed in the table, then the subroutine call list in the user's code must be modified to conform with the call list in the updated library in order to work properly. For the main library, AMOSSUBS.FOR + BECKSUBS.FOR, these call lists are listed in the first subroutine on each file. For codes taken from the other files, one must search the file for the subroutine that was extracted for the user's application.

NEW in the **CHANGE** column means a new routine was added to the 2003 collection to make the new collection for this edition.

FILE	CHANGE	CODE NAME	COMMENT
BECKSUBS.FOR			
	CALL LIST	SUBROUTINE INTEGI3	KODE added
	CALL LIST	SUBROUTINE INTEGI6	KODE added
	CALL LIST	SUBROUTINE INTEGP	KODE added
	NEW	SUBROUTINE INTEGI2	
	NEW	SUBROUTINE INTEGI9	
	NEW	SUBROUTINE INTEGV5	
	NEW	SUBROUTINE INTEGI29	
BECKDRVR.FOR			
	NEW	PROGRAM I2COMP	
	NEW	PROGRAM I9COMP	
	NEW	PROGRAM V5COMP	
	NEW	PROGRAM PCOMP	
	NEW	PROGRAM QCOMP	
	DELETED	PROGRAM PQCOMP	
	NEW	PROGRAM I29COMP	
RESEARCH.FOR			
	CALL LIST	SUBROUTINE INTEGI13	KODE added
	DELETED	SUBROUTINE INTEG14	
	NEW	SUBROUTINE INTEGI14	Replaces INTEG14
	DELETED	PROGRAM I4BYSER	
	DELETED	PROGRAM I4BYQUAD	
	DELETED	PROGRAM J4BYSER	
	DELETED	PROGRAM J4BYQUAD	
	NEW	PROGRAM I4COMP	
	NEW	PROGRAM J4COMP	
	NEW	PROGRAM DGSCOMP	
AMOSSUBS.FOR			
	NEW	SUBROUTINE DQUAD8	No change in call list.
	CAPABILITY *		See * below.
AMOSDRVR.FOR			
	NO CHANGES		

* DQUAD8 was originally designed to compute quadratures on an infinite interval starting at X1 and progressing in steps (intervals) of length SIG. The initialization parameter INIT=0 starts the procedure and a convergence criterion specified by the parameter REL terminates the procedure, returning not only quadrature totals in QANS, but also the final point X2. The new capability allows DQUAD8 to compute exactly m steps of length SIG by setting $\mathrm{INIT} = -m$, m>0. To cover an interval (a,b), compute SIG by $\sigma = (b\text{-}a)/m$. On return from DQUAD8 with no error flag and with this value of SIG, X2=b and QANS is the total quadrature. REL still specifies the accuracy for each of the m quadratures.

1. Table of Integrals

This table of integrals presents the results of the computations carried out in Chapter 3. Chapter 2 presents the main results of Chapter 3 in handbook format.

(1.0) Integrals

Fundamental Integrals

	Integral	Ref.
(1.0.1)	$F(x)=\int_0^x \frac{\operatorname{erf}(w)}{w}dw$, x≥0	**(2.1)**
(1.0.2)	$G(x)=\int_x^\infty \frac{\operatorname{erfc}(w)}{w}dw$, x>0	**(2.1)**
(1.0.3)	$F(x)=\frac{\gamma}{2}+\ln(2x)+G(x)$ (γ =0.577215664901532860б…)	**(2.1)**
(1.0.4)	$\int_0^x e^{-a^2w^2}\ln w dw=\frac{\sqrt{\pi}}{2a}[\operatorname{erf}(ax)\ln x-F(ax)]$, x≥0	**(2.1)**
(1.0.5)	$\int_x^\infty e^{-a^2w^2}\ln w dw=\frac{\sqrt{\pi}}{2a}[\operatorname{erfc}(ax)\ln x+G(ax)]$, x>0	**(2.1)**
(1.0.6)	$\int_0^\infty e^{-a^2w^2}\ln w dw=-\frac{\sqrt{\pi}}{2a}\left[\frac{\gamma}{2}+\ln(2a)\right].$	**(2.1)**
(1.0.7)	$G_\nu(x)\equiv\int_1^\infty \frac{E_\nu(xw)}{w^\nu}dw=\frac{-\partial E_\nu(x)}{\partial\nu}=\int_1^\infty \frac{e^{-xw}\ln w}{w^\nu}dw$, x>0, ν>0	**(2.2)**
(1.0.8)	$G_{1/2}(x)=2\sqrt{\frac{\pi}{x}}G(\sqrt{x})$, x>0	**(2.2)**
(1.0.9)	$\int_x^\infty \frac{E_\mu(w)}{w^\nu}dw=\frac{1}{x^{\nu-1}}\frac{E_\nu(x)-E_\mu(x)}{\mu-\nu}$ or	**(2.2)**
(1.0.10)	$\int_1^\infty \frac{E_\mu(xw)}{w^\nu}dw=\frac{E_\nu(x)-E_\mu(x)}{\mu-\nu}$, x>0, μ>0, ν>0, μ≠ν	**(2.2)**
(1.0.11)	$\int_x^\infty \frac{E_\nu(w)}{w^\nu}dw=\frac{1}{x^{\nu-1}}G_\nu(x)$, x>0 or	**(2.2)**
(1.0.12)	$\int_1^\infty \frac{E_\nu(xw)}{w^\nu}dw=G_\nu(x)$, x>0, ν>0 (μ = ν above)	**(2.2)**
(1.0.13)	$\int_1^\infty \frac{E_{1/2}(xw)}{\sqrt{w}}dw=G_{1/2}(x)=2\sqrt{\frac{\pi}{x}}G(\sqrt{x})$	**(2.2)**
(1.0.14)	$I_n(b,x)=\int_x^\infty \frac{e^{-b^2w^2}\ln w}{w^n}dw$, $n=0, 1, 2, \ldots$, x > 0	**(2.2)**
(1.0.15)	$I_n(b,x)=\frac{\ln x}{2x^{n-1}}E_{(n+1)/2}(b^2x^2)+\frac{1}{4x^{n-1}}G_{(n+1)/2}(b^2x^2)$, x>0	**(2.2)**
(1.0.16)	$I_0(b,x)=\int_x^\infty e^{-b^2w^2}\ln w dw=\frac{\sqrt{\pi}}{2b}[\operatorname{erfc}(bx)\ln x+G(bx)]$	**(2.2)**

(1.0.17) $$\frac{(n+1)}{2b^2} I_{n+2} + I_n = \frac{\ln x}{2b^2} \cdot \frac{e^{-b^2x^2}}{x^{n+1}} + \frac{1}{4b^2 x^{n+1}} E_{(n+3)/2}(b^2x^2), \qquad n \ge 0$$ (2.2)

(1.0.18) $$J_\nu(a,x) = \int_x^\infty \frac{\mathrm{erf}(aw)}{w^\nu} dw = \frac{1}{(\nu-1)x^{\nu-1}} \left[\mathrm{erf}(ax) + \frac{ax}{\sqrt{\pi}} E_{\nu/2}(a^2x^2) \right], \qquad \nu > 1$$ (2.11)

(1.0.19) $$J_\nu^c(a,x) = \int_x^\infty \frac{erfc(aw)}{w^\nu} dw = \frac{1}{(\nu-1)x^{\nu-1}} \left[erfc(ax) - \frac{ax}{\sqrt{\pi}} E_{\nu/2}(a^2x^2) \right], \qquad \nu \ne 1$$ (2 .11)

(1.0.20) $$J_2(a,x) = \int_x^\infty \frac{\mathrm{erf}(aw)}{w^2} dw = \frac{\mathrm{erf}(ax)}{x} + \frac{a}{\sqrt{\pi}} E_1(a^2x^2), \qquad \nu = 2$$ (2.11)

(1.0.21) $$J_2^c(a,x) = \int_x^\infty \frac{\mathrm{erfc}(aw)}{w^2} dw = \frac{\mathrm{erfc}(ax)}{x} - \frac{a}{\sqrt{\pi}} E_1(a^2x^2), \qquad \nu = 2$$ (2.11)

(1.0.22) $$J_3(a,x) = \int_x^\infty \frac{\mathrm{erf}(aw)}{w^3} dw = \frac{\mathrm{erf}(ax)}{2x^2} + \frac{a}{x} \mathrm{ierfc}(ax), \qquad \nu = 3$$ (2.11)

(1.0.23) $$J_3^c(a,x) = \int_x^\infty \frac{\mathrm{erfc}(aw)}{w^3} dw = \frac{\mathrm{erfc}(ax)}{2x^2} - \frac{a}{x} \mathrm{ierfc}(ax) = \frac{2i^2 \mathrm{erfc}(ax)}{x^2}, \qquad \nu = 3$$ (2.11)

(1.0.24) $$I_5(a,b,x) = \int_x^\infty e^{-a^2w^2} erfc(bw) dw = \frac{1}{2d\sqrt{\pi}} \sum_{k=0}^\infty \frac{(1/2)_k}{k!} \left(\frac{a^2}{d^2} \right)^k E_{k+3/2}(d^2x^2), \quad \mathrm{a} \le \mathrm{b}$$ (2.3)

(1.0.25) $$I_5(a,b,x) = \frac{be^{-a^2x^2}}{a^2+b^2} \sum_{k=1}^\infty \left(\frac{4a^2}{a^2+b^2} \right)^{k-1} [(k-1)!\, i^{2k-1} erfc(bx)], \qquad a \le b$$ (2.3)

(1.0.26) $$= \frac{\sqrt{\pi}}{2a} erfc(ax)\, erfc(bx) - \frac{b}{a} I_5(b,a,x), \qquad d^2 = a^2 + b^2, \qquad \mathrm{a} > \mathrm{b}$$ (2.3)

(1.0.27) $$J_5(a,b,x) = \int_x^\infty e^{-a^2w^2} erf(bw) dw = \frac{\sqrt{\pi}}{2a} \mathrm{erfc}(ax) - I_5(a,b,x)$$ (2.3)

(1.0.28) $$U_5(a,b,x) = \int_0^x e^{-a^2w^2} erfc(bw) dw = \frac{1}{a\sqrt{\pi}} \tan^{-1}\frac{a}{b} - I_5(a,b,x)$$ (2.3)

(1.0.29) $$V_5(a,b,x) = \int_0^x e^{-a^2w^2} erf(bw) dw = \frac{\sqrt{\pi}}{2a} \mathrm{erf}(ax) - U_5(a,b,x)$$

$$= \frac{1}{a\sqrt{\pi}} \tan^{-1}\frac{b}{a} - J_5(a,b,x)$$ (2.3)

(1.0.30) $$\mathrm{I}_2(\mathrm{a,b,T}) = \int_0^{\mathrm{T}} \mathrm{erf(aw)erf(bw)dw} = \frac{1}{2}\int_{\mathrm{t}}^\infty \frac{\mathrm{erf}(\mathrm{a}/\sqrt{\tau})\mathrm{erf}(\mathrm{b}/\sqrt{\tau})}{\tau^{3/2}} d\tau, \qquad T = \frac{1}{\sqrt{t}}$$ (2.8)

(1.0.31) $$\mathrm{I}_2^{\mathrm{c}}(\mathrm{a,b,T}) = \int_T^\infty \mathrm{erfc}(aw)\,\mathrm{erfc}(bw) dw = \frac{1}{2}\int_0^t \frac{\mathrm{erf}(\mathrm{a}/\sqrt{\tau})\mathrm{erf}(\mathrm{b}/\sqrt{\tau})}{\tau^{3/2}} d\tau$$ (2.8)

(1.0.32) $$\mathrm{I}_2^{\mathrm{c}}(\mathrm{a,b,0}) = \int_0^\infty \mathrm{erfc(aw)erfc(bw)dw} = \frac{(2/\sqrt{\pi})}{\mathrm{a+b}+\sqrt{\mathrm{a}^2+\mathrm{b}^2}}$$ (2.8)

(1.0.33) $$\mathrm{I}_9(\mathrm{a,b,T}) = \int_0^{\mathrm{T}} \mathrm{w\,erf(aw)erf(bw)dw} = \frac{1}{2}\int_{\mathrm{t}}^\infty \frac{\mathrm{erf}(\mathrm{a}/\sqrt{\mathrm{u}})\mathrm{erf}(\mathrm{b}/\sqrt{\mathrm{u}})\mathrm{du}}{\mathrm{u}^2}$$ (2.8)

(1.0.34) $$\mathrm{I}_9^{\mathrm{c}}(\mathrm{a,b,T}) = \int_{\mathrm{T}}^\infty \mathrm{w\,erfc(aw)erfc(bw)dw} = \frac{1}{2}\int_0^{\mathrm{t}} \frac{\mathrm{erfc}(\mathrm{a}/\sqrt{\mathrm{u}})\mathrm{erfc}(\mathrm{b}/\sqrt{\mathrm{u}})\mathrm{du}}{\mathrm{u}^2}$$ (2.8)

(1.0.35) $$\mathrm{I}_{20}(\mathrm{a,b,T}) = \int_{\mathrm{T}}^\infty \mathrm{e}^{-\mathrm{a}^2\mathrm{x}^2} \mathrm{erf(bx)\ lnxdx},$$ (2.4)

(1.0.36) $$I_{20}^c(a,b,T)=\int_T^\infty e^{-a^2x^2}\operatorname{erfc}(bx)\ln x\,dx,$$ (2.4)

(1.0.37) $$P(a,b,T)=\int_T^\infty \frac{e^{-a^2w^2}\operatorname{erf}(bw)}{w}dw$$ (2.5)

(1.0.38) $$P^c(a,b,T)=\int_T^\infty \frac{e^{-a^2w^2}\operatorname{erfc}(bw)}{w}dw$$ (2.5)

(1.0.39) $$Q(a,b,T)=\int_T^\infty e^{-a^2w^2}E_1(b^2w^2)dw$$ (2.5)

Results in Terms of Fundamental Integrals (1.0.1)-(1.0.39), $T=1/\sqrt{t}$

(1.0.40) $$\int_T^\infty \frac{e^{-a^2w^2}\operatorname{erf}(bw)}{w^2}dw=\frac{1}{2}\int_0^t e^{-a^2/u}\operatorname{erf}(b/\sqrt{u})/\sqrt{u}du=I_1(a,b,T)$$ (2.6)

$$=\frac{e^{-a^2T^2}\operatorname{erf}(bT)}{T}+\frac{b}{\sqrt{\pi}}E_1[T^2(a^2+b^2)]-2a^2J_5(a,b,T)$$ (2.6)

(1.0.41) $$\int_T^\infty \frac{e^{-a^2w^2}\operatorname{erfc}(bw)}{w^2}dw=\frac{1}{2}\int_0^t e^{-a^2/u}\operatorname{erfc}(b/\sqrt{u})/\sqrt{u}du=I_1^c(a,b,T)$$ (2.6)

$$=\frac{e^{-a^2T^2}\operatorname{erfc}(bT)}{T}-\frac{b}{\sqrt{\pi}}E_1[T^2(a^2+b^2)]-2a^2I_5(a,b,T)$$ (2.6)

(1.0.42) $$\int_T^\infty \frac{e^{-a^2w^2}\operatorname{erf}(bw)}{w^3}dw=\frac{1}{2}\int_0^t e^{-a^2/u}\operatorname{erf}(b/\sqrt{u})du=I_{13}(a,b,T)$$ (2.7)

$$=\frac{e^{-a^2T^2}}{2T^2}\operatorname{erf}(bT)+\frac{b}{T}i\operatorname{erfc}(T\sqrt{a^2+b^2})-a^2P(a,b,T)$$

(1.0.43) $$\int_T^\infty \frac{e^{-a^2w^2}\operatorname{erfc}(bw)}{w^3}dw=\frac{1}{2}\int_0^t e^{-a^2/u}\operatorname{erfc}(b/\sqrt{u})du=I_{13}^c(a,b,T)$$ (2.7)

$$=\frac{1}{2T^2}E_2(a^2T^2)-I_{13}(a,b,T)$$

(1.0.44) $$\int_0^T \frac{\operatorname{erf}(aw)\operatorname{erf}(bw)}{w}dw=\frac{1}{2}\int_t^\infty \frac{\operatorname{erf}(a/\sqrt{\tau})\operatorname{erf}(b/\sqrt{\tau})}{\tau}d\tau=I_{19}(a,b,T)$$ (2.9)

(1.0.45) $$\int_T^\infty \frac{\operatorname{erfc}(aw)\operatorname{erfc}(bw)}{w}dw=\frac{1}{2}\int_0^t \frac{\operatorname{erfc}(a/\sqrt{\tau})\operatorname{erfc}(b/\sqrt{\tau})}{\tau}d\tau=I_{19}^c(a,b,T)$$ (2.9)

(1.0.46) $$\int_T^\infty \frac{\operatorname{erf}(aw)\operatorname{erfc}(bw)}{w}dw=\frac{1}{2}\int_0^t \frac{\operatorname{erf}(a/\sqrt{\tau})\operatorname{erfc}(b/\sqrt{\tau})}{\tau}d\tau=G(bT)-I_{19}^c(a,b,T)$$ (2.9)

(1.0.47) $$\int_T^\infty \frac{\operatorname{erf}(aw)\operatorname{erf}(bw)}{w^2}dw=\frac{1}{2}\int_0^t \frac{\operatorname{erf}\left(a/\sqrt{\tau}\right)\operatorname{erf}\left(b/\sqrt{\tau}\right)}{\tau^{1/2}}d\tau=I_6(a,b,T)$$ (2.10)

(1.0.48) $$\int_T^\infty \frac{\operatorname{erfc}(aw)\operatorname{erfc}(bw)}{w^2}dw=\frac{1}{2}\int_0^t \frac{\operatorname{erfc}(a/\sqrt{\tau})\operatorname{erfc}(b/\sqrt{\tau})}{\tau^{1/2}}d\tau=I_6^c(a,b,T)$$ (2.10)

(1.0.49) $$\int_T^\infty \frac{\operatorname{erf}(aw)\operatorname{erfc}(bw)}{w^2}dw=\frac{1}{2}\int_0^t \frac{\operatorname{erf}(a/\sqrt{\tau})\operatorname{erfc}(b/\sqrt{\tau})}{\tau^{1/2}}d\tau$$ (2.10)

$$=J_2(a,T)-I_6(a,b,T)=J_2^c(b,T)-I_6^c(a,b,T)$$

(1.0.50) $$\int_T^\infty \frac{\operatorname{erf}(aw)\operatorname{erf}(bw)}{w^3}dw=\frac{1}{2}\int_0^t \operatorname{erf}(a/\sqrt{\tau})\operatorname{erf}(b/\sqrt{\tau})d\tau=W_3(a,b,T)$$ (2.11)

(1.0.51) $$\int_T^\infty \frac{\operatorname{erfc}(aw)\operatorname{erfc}(bw)}{w^3}dw=\frac{1}{2}\int_0^t \operatorname{erfc}(a/\sqrt{\tau})\operatorname{erfc}(b/\sqrt{\tau})d\tau=W_3^c(a,b,T)$$ **(2.11)**

(1.0.52) $$\int_T^\infty \frac{\operatorname{erf}(aw)\operatorname{erfc}(bw)}{w^3}dw=\frac{1}{2}\int_0^t \operatorname{erf}(a/\sqrt{\tau})\operatorname{erfc}(b/\sqrt{\tau})d\tau$$ **(2.11)**

$$=J_3(a,T)-W_3(a,b,T)=J_3^c(b,T)-W_3^c(a,b,T)$$

(1.0.53) $$\int_T^\infty e^{-c^2w^2}\operatorname{erf}(aw)\operatorname{erf}(bw)dw=\frac{1}{2}\int_0^t e^{-c^2/u}\frac{\operatorname{erf}(a/\sqrt{u})\operatorname{erf}(b/\sqrt{u})}{u^{3/2}}du=I_3(a,b,c,T)$$ **(2.12)**

(1.0.54) $$\int_T^\infty e^{-c^2w^2}\operatorname{erfc}(aw)\operatorname{erfc}(bw)dw=\frac{1}{2}\int_0^t e^{-c^2/u}\frac{\operatorname{erfc}(a/\sqrt{u})\operatorname{erfc}(b/\sqrt{u})}{u^{3/2}}du$$ **(2.12)**

$$=J_3(a,b,c,T)=I_3^c(a,b,c,T)$$

(1.0.55) $$\int_T^\infty w\, e^{-c^2w^2}\operatorname{erf}(aw)\operatorname{erf}(bw)dw=\frac{1}{2}\int_0^t \frac{e^{-c^2/u}}{u^2}\operatorname{erf}\left(a/\sqrt{u}\right)\operatorname{erf}\left(b/\sqrt{u}\right)du=J_4(a,b,c,T)$$ **(2.13)**

(1.0.56) $$\int_T^\infty w\, e^{-c^2w^2}\operatorname{erf}(aw)dw=\frac{1}{2}\int_0^t \frac{e^{-c^2/u}}{u^2}\operatorname{erf}\left(a/\sqrt{u}\right)du=J_4(a,\infty,c,T)$$ **(2.13)**

(1.0.58) $$\int_T^\infty we^{-c^2w^2}\operatorname{erfc}(aw)\operatorname{erfc}(bw)dw=\frac{1}{2}\int_0^t \frac{e^{-c^2/u}}{u^2}\operatorname{erfc}\left(a/\sqrt{u}\right)\operatorname{erfc}\left(b/\sqrt{u}\right)du=J_4^c(a,b,c,T)$$ **(2.13)**

(1.0.59) $$\int_T^\infty we^{-c^2w^2}\operatorname{erfc}(aw)dw=\frac{1}{2}\int_0^t \frac{e^{-c^2/u}}{u^2}\operatorname{erfc}\left(a/\sqrt{u}\right)du=J_4^c(a,0,c,T)$$ **(2.13)**

(1.0.60) $$\int_T^\infty w^2e^{-c^2w^2}\operatorname{erf}(aw)\operatorname{erf}(bw)dw=\frac{1}{2}\int_0^t \frac{e^{-c^2/u}}{u^{5/2}}\operatorname{erf}\left(\frac{a}{\sqrt{u}}\right)\operatorname{erf}\left(\frac{b}{\sqrt{u}}\right)du=I_4(a,b,c,T)$$ **(2.13)**

(1.0.61) $$\int_T^\infty w^2e^{-c^2w^2}erf(aw)dw=\frac{1}{2}\int_0^t \frac{e^{-c^2/u}}{u^{5/2}}\operatorname{erf}\left(\frac{a}{\sqrt{u}}\right)du=I_4(a,\infty,c,T)$$ **(2.13)**

(1.0.62) $$\int_T^\infty w^2e^{-c^2w^2}\operatorname{erfc}(aw)\operatorname{erfc}(bw)dw=\frac{1}{2}\int_0^t \frac{e^{-c^2/u}}{u^{5/2}}\operatorname{erfc}\left(\frac{a}{\sqrt{u}}\right)\operatorname{erfc}\left(\frac{b}{\sqrt{u}}\right)du=I_4^c(a,b,c,T)$$ **(2.13)**

(1.0.63) $$\int_T^\infty w^2e^{-c^2w^2}\operatorname{erfc}(aw)dw=\frac{1}{2}\int_0^t \frac{e^{-c^2/u}}{u^{5/2}}\operatorname{erfc}\left(\frac{a}{\sqrt{u}}\right)du=I_4^c(a,0,c,T)$$ **(2.13)**

(1.0.64) $$\int_T^\infty e^{-c^2w^2}\frac{\operatorname{erf}(aw)\operatorname{erf}(bw)}{w^2}dw=\frac{1}{2}\int_0^t e^{-c^2/u}\frac{\operatorname{erf}(a/\sqrt{u})\operatorname{erf}(b/\sqrt{u})}{\sqrt{u}}du=I_{14}(a,b,c,T)$$ **(2.14)**

(1.0.65) $$\int_T^\infty e^{-c^2w^2}\frac{\operatorname{erfc}(aw)\operatorname{erfc}(bw)}{w^2}dw=\frac{1}{2}\int_0^t e^{-c^2/u}\frac{\operatorname{ercf}(a/\sqrt{u})\operatorname{erfc}(b/\sqrt{u})}{\sqrt{u}}du=I_{14}^c(a,b,c,T)$$ **(2.14)**

Let

(1.0.66) $$Y_n(a,b,T)=\int_T^\infty e^{-a^2w^2}i^n erfc(bw)dw,\quad \mathrm{V}_n(a,b,T)=\int_T^\infty we^{-a^2w^2}i^n erfc(bw)dw,\qquad n\ge -1$$ **(2.3)**

(1.0.67) $$Y_{-1}(a,b,T)=\frac{erfc(T\sqrt{a^2+b^2})}{\sqrt{a^2+b^2}},\qquad \mathrm{Y}_0(a,b,T)=I_5(a,b,T),$$

(1.0.68) $$V_{-1}(a,b,T)=\frac{e^{-(a^2+b^2)T^2}}{(a^2+b^2)\sqrt{\pi}},\ \mathrm{V}_0(a,b,T)=\frac{1}{2a^2}\left[e^{-a^2T^2}erfc(bT)-\frac{b}{\sqrt{a^2+b^2}}erfc(T\sqrt{a^2+b^2})\right]$$

(1.0.69) $$2nY_n-\left(1+\frac{b^2}{a^2}\right)Y_{n-2}=\frac{-b}{a^2}e^{-a^2T^2}i^{n-1}erfc(bT),\quad \mathrm{V}_n=\frac{1}{2a^2}\left[e^{-a^2T^2}i^n erfc(bT)-bY_{n-1}\right],\quad n\ge 1$$

In particular,

(1.0.70) $$Y_1(a,b,T)=\int_T^\infty e^{-a^2w^2}\,ierfc(bw)dw=\frac{1}{2a^2}\left[\sqrt{a^2+b^2}\,erfc(T\sqrt{a^2+b^2})-be^{-a^2T^2}\,erfc(bT)\right]$$

(1.0.71) $$V_1(a,b,T)=\int_T^\infty we^{-a^2w^2}\,ierfc(bw)dw=\frac{1}{2a^2}\left[e^{-a^2T^2}\,ierfc(bT)-bI_5(a,b,T)\right]$$

(1.0.72) $$Y_n(a,b,T)=\frac{be^{-a^2T^2}}{a^2+b^2}\sum_{k=1}^{\infty}\left(\frac{4a^2}{a^2+b^2}\right)^{k-1}\frac{\Gamma(n/2+k)}{\Gamma(n/2+1)}i^{n+2k-1}erfc(bT),\quad n\ge 0,\qquad a\le b$$

(1.0.73) $$Y_n(a,b,T)=\frac{\sqrt{\pi}}{2a}\sum_{k=0}^{n}(-1)^k\left(\frac{b}{a}\right)^k i^k erfc(aT)\,i^{n-k}erfc(bT)+(-1)^{n+1}\left(\frac{b}{a}\right)^{n+1}Y_n(b,a,T),\ a>b$$

Integrals Related To the Function U(a,b,t)

$$\mathbf{U(a,b,t)=e^{a^2t+2ab}erfc(a\sqrt{t}+b/\sqrt{t})}$$
$$\boldsymbol{a>0,\quad b>0,\quad c>0,\quad d>0,\quad t>0}$$

(1.0.74) $$V(a,b,t)=\int_0^t U(a,b,\tau)d\tau,$$ **(2.15)**

(1.0.75) $$I_{21}(a,b,c,t)=\int_0^t U(a,b,\tau)\mathrm{erf}(c/\sqrt{\tau})d\tau$$ **(2.15)**

(1.0.76) $$I_{21}^c(a,b,c,t)=\int_0^t U(a,b,\tau)\mathrm{erfc}(c/\sqrt{\tau})d\tau$$ **(2.15)**

(1.0.77) $$J_{21}(a,b,c,t)=\int_0^t \frac{U(a,b,\tau)e^{-c^2/\tau}}{\tau^{3/2}}d\tau$$ **(2.15)**

(1.0.78) $$I_{22}(a,b,c,t)=\int_0^t U(a,b,\tau)\frac{e^{-c^2/\tau}}{\sqrt{\tau}}d\tau$$ **(2.16)**

(1.0.79) $$J_{22}(a,b,c,t)=\int_0^t U(a,b,\tau)\sqrt{\tau}e^{-c^2/\tau}\,d\tau$$ **(2.16)**

(1.0.80) $$I_{24}(a,b,c,t)=\int_0^t \tau U(a,b,\tau)\mathrm{erf}\left(\frac{c}{\sqrt{\tau}}\right)d\tau,$$ **(2.17)**

(1.0.81) $$I_{24}^c(a,b,c,t)=\int_0^t \tau U(a,b,\tau)\mathrm{erfc}\left(\frac{c}{\sqrt{\tau}}\right)d\tau,$$ **(2.17)**

(1.0.82) $$J_{24}(a,b,t)=\int_0^t \tau U(a,b,\tau)d\tau,$$ **(2.17)**

(1.0.83) $$V_{24}(a,b,t)=\int_0^t V(a,b,\tau)d\tau,$$ **(2.17)**

(1.0.84) $$I_{25}(a,b,c,d,t)=\int_0^t U(a,b,\tau)U(c,d,\tau)d\tau$$ **(2.18)**

(1.0.85) $$I_{26}(a,b,c,d,t)=\int_0^t \tau U(a,b,\tau)U(c,d,\tau)d\tau$$ **(2.19)**

(1.0.86) $$J=\int e^{(a^2-b^2)x}\mathrm{erfc}\left(a\sqrt{x}+\frac{c}{\sqrt{x}}\right)dx$$ **(2.20)**

(1.0.87) $$I=\int xe^{(a^2-b^2)x}\mathrm{erfc}\left(a\sqrt{x}+\frac{c}{\sqrt{x}}\right)dx$$ **(2.20)**

Miscellaneous Integrals $x \geq 0$

(1.0.88) $$H_{23}(x) = \int_0^x e^{w^2}\,\mathrm{erfc}(w)dw = \frac{1}{2}\int_0^{x^2} e^v\,\mathrm{erfc}(\sqrt{v})\frac{dv}{\sqrt{v}}$$ **(2.21)**

(1.0.89) $$I_{23}(x) = \int_0^x e^{w^2}\,\mathrm{erfc}(w)\,w\,dw = \frac{1}{2}\int_0^{x^2} e^v\,\mathrm{erfc}(\sqrt{v})dv$$ **(2.21)**

(1.0.90) $$J_{23}(x) = \int_0^x e^{w^2}\,\mathrm{erfc}(w)w^2dw = \frac{1}{2}\int_0^{x^2} e^v\,\mathrm{erfc}(\sqrt{v})\sqrt{v}dv$$ **(2.21)**

Reduction formula ending in H_{23} or I_{23}

(1.0.91) $$\int_0^x e^{w^2}\,\mathrm{erfc}(w)w^\alpha dw = \frac{x^{\alpha-1}}{2}e^{x^2}\mathrm{erfc}(x) + \frac{x^\alpha}{\alpha\sqrt{\pi}} - \frac{(\alpha-1)}{2}\int_0^x e^{w^2}\,\mathrm{erfc}(w)w^{\alpha-2}dw$$ **(2.21)**

$\alpha = 2n$ or $2n+1$, $\quad n = 1, 2, \ldots$

(1.0.92) $$G_n(a,b,x) = \int_x^\infty \frac{e^{-a^2w^2}i^n\mathrm{erfc}(bw)}{w^n}dw, \qquad n = 1, 2, \ldots, a>0, \quad b>0$$ **(2.15)**

(1.0.93) $$I(a,b,x) = \int_x^\infty e^{-at-b/t}dt, \qquad a > 0,\ b > 0,\ x > 0$$ **(2.22)**

Inverse LaPlace Transform:

(1.0.94) $$L^{-1}\left[\frac{e^{-2b\sqrt{p}}}{\sqrt{p}(\sqrt{p}+a)^{n+1}}\right] = (2\sqrt{t})^n e^{a^2t+2ab}i^n\mathrm{erfc}(a\sqrt{t}+b/\sqrt{t})$$

n=0 gives

(1.0.95) $$L^{-1}\left[\frac{e^{-2b\sqrt{p}}}{\sqrt{p}(\sqrt{p}+a)}\right] = e^{a^2t+2ab}\mathrm{erfc}(a\sqrt{t}+b/\sqrt{t}) = U(a,b,t)$$

Useful expansions for small aT (See **(1.0.68)** and **(1.0.70)** and Chapter 3, Folder 9):

(1.0.96) $$erfc(T\sqrt{a^2+b^2}) = e^{-a^2T^2}\sum_{k=0}^{\infty}(-2)^k\,i^k erfc(bT)(bT\phi)^k, \qquad bT\phi = \frac{a^2T}{b+\sqrt{a^2+b^2}}$$

(1.0.97) $$ierfc(T\sqrt{a^2+b^2}) = e^{-a^2T^2}ierfc(bT) + e^{-a^2T^2}\sum_{k=2}^{\infty}(-2)^{k-1}k\,i^k erfc(bT)(bT\phi)^{k-1}$$

The general case is:

(1.0.98) $$i^n erfc(T\sqrt{a^2+b^2}) = e^{-a^2T^2}i^n erfc(bT) + e^{-a^2T^2}\sum_{k=n+1}^{\infty}(-2)^{k-n}C_n^k\,i^k erfc(bT)(bT\phi)^{k-n}$$

where C_n^k is a binomial coefficient.

Reciprocal relations for $E_{n+1/2}(x^2)$ and $i^n erfc(x)$ (See Chapter 3, Folder 28):

$$A(n,k) = (-1)^{k-1}2^{2k-1}(k-1)!\,C_{k-1}^{n-1} = \frac{(-1)^{k-1}2^{2k-1}(n-1)!}{(n-k)!}, \qquad 1 \leq k \leq n, \qquad n \geq 1$$

$$B(n,k) = \frac{(-1)^{k-1}C_{k-1}^{n-1}}{2^{2n-1}(n-1)!} = \frac{(-1)^{k-1}}{2^{2n-1}(k-1)!(n-k)!}, \qquad 1 \leq k \leq n, \qquad n \geq 1$$

(1.0.99) $$\frac{1}{\sqrt{\pi}} E_{n+1/2}(x^2) = \sum_{k=1}^{n} A(n,k)\, i^{2k-1} erfc(x) \ , \qquad n \geq 1$$

(1.0.100) $$i^{2n-1} erfc(x) = \sum_{k=1}^{n} B(n,k)[E_{k+1/2}(x^2)/\sqrt{\pi}], \qquad n \geq 1,$$

(1.0.101) $$\frac{2x}{\sqrt{\pi}} E_{n+1/2}(x^2) = \sum_{k=0}^{n} A(n+1,k+1)\, i^{2k} erfc(x) \ , \qquad n \geq 0$$

(1.0.102) $$i^{2n} erfc(x) = 2x \sum_{k=0}^{n} B(n+1,k+1)[E_{k+1/2}(x^2)/\sqrt{\pi}], \qquad n \geq 0$$

(1.1) Comments on the Use of Quadrature

Introduction

The main thrust of this work is directed toward the development of forms suitable for high accuracy computation (double precision arithmetic). It is not uncommon to find that representations of the desired integral break down computationally over some parameter ranges and a suitable alternative cannot be found. In these cases, a numerical quadrature is used as a default even though a quadrature is generally computationally more expensive. The quadratures used for this study are done on integrals whose integrands are analytic over the range of integration. This is achieved by a change of variable or a change in representation since the natural form of heat conduction solutions in the time variable often involves arguments of a square root or reciprocal square root. Historically, analytic forms (their Taylor series exist) are a kind of standard because a lot of work has been done to approximate these integrals by integration of polynomial approximations of the integrand (locally, a truncated Taylor series is a polynomial). Consequently, there are robust codes which will do the integration automatically without excessive programmer intervention. In these codes, only a subroutine for the function, the limits of integration, and an error tolerance REL need be specified. DGAUS8 written by R. E. Jones is one such routine which can be found on the SLATEC library and is used as part of this work. This subroutine was developed for high accuracy (double precision) applications and is well tested. It is adaptive, meaning that an 8-point Gauss formula is applied on subintervals where an error estimate is made for a given interval. If the estimate meets the required accuracy, then this interval is excluded from further subdivision and the routine moves on to test and subdivide until all subintervals meet the requirements or a specified limit is reached. Furthermore, one can expect significant digits from non-negative integrands because the 8-point Gauss formula has positive coefficients and losses of significant digits by differences of nearly equal quantities cannot occur. It is also assumed that a suitable double precision library of special functions is available to compute the integrand to the required accuracy.

Computing With DGAUS8 and DQUAD8

We apply DGAUS8 in a special way on infinite integrals to construct a routine called DQUAD8. To be a candidate for DQUAD8, the integral should converge rapidly. In these cases it is important to estimate a scale of integration in order to do the computation efficiently and not let the routine search for the region where most of the work needs to be done. Many of the integrands which result in quadratures are dominated by an exponential. In these cases, the scale of integration can be estimated by a "standard deviation", $\boldsymbol{\sigma}$. To be precise, we consider integrals of the form

(1.1.1) $$\mathrm{I} = \int_X^\infty f(x)dx$$

with

$$f(x) = g(x)e^{-ax} \qquad \text{or} \qquad f(x) = g(x)e^{-a^2x^2}$$

where g(x) is slowly varying and the estimate for $\boldsymbol{\sigma}$ is given by

(1.1.2) $$\sigma = \frac{\mathrm{m}}{\mathrm{a}}, \quad \mathrm{m} = 4 \text{ or } 5 \quad \text{or} \quad \sigma = \frac{\mathrm{m}}{\mathrm{a}\sqrt{2}}, \quad \mathrm{m}=3 \text{ or } 4.$$

Notice that when a is small then the scale of integration can be very large and when a is large the scale of integration can be very small. Then the quadrature proceeds according to the formula

(1.1.3) $$\mathrm{I}_{\mathrm{K}} = \sum_{\mathrm{k}=1}^{\mathrm{K}} \mathrm{Q}_{\mathrm{k}}, \qquad Q_k = \int_{X+(k-1)\sigma}^{X+k\sigma} f(x)dx,$$

and Q_k is computed by DGAUS8 with the REL parameter to specify the accuracy. The sum is terminated in DQUAD8 on a relative error test

(1.1.4) $$\left|Q_K / I_K\right| \le REL$$

This termination procedure is not rigorous, but it can be made rigorous by estimating the truncation error. Thus, if g(x) is bounded by M for x>R, then the truncation at R gives an estimate

(1.1.5) $$\left|T_R\right| \le \frac{M}{a} e^{-aR} = B_R \quad \text{or} \quad \left|T_R\right| \le \frac{M(\sqrt{\pi}/2)\mathrm{erfc}(aR)}{a} < \frac{M}{2a^2 R} e^{-a^2R^2} = B_R, \qquad \mathrm{R} > \mathrm{X},$$

where we have used an estimate of Mill's ratio to estimate erfc(aR) [A&S, 7.1.13]:

$$\frac{erfc(x)}{e^{-x^2}} \le \frac{2/\sqrt{\pi}}{x+\sqrt{x^2+4/\pi}} < \frac{1}{x\sqrt{\pi}}.$$

DQUAD8 returns not only an answer I_K but also the end point of the quadrature, $\mathrm{X}_{\mathrm{K}} = \mathrm{X} + \mathrm{K}\sigma$, after K applications of DGAUS8. Because DQUAD8 can also be called repeatedly with no change in the call list, DQUAD8 can be put into a loop. On each return from DQUAD8, the truncation error can be estimated by one of the formulas above with $R = X_K$. If the inequality

(1.1.6) $$\left|B_R / I_K\right| \le REL$$

is satisfied, then exit from the loop; otherwise, the looping continues until the estimate is satisfied or the maximum loop index is reached. Experience has indicated that the truncation error estimate is almost always satisfied on the first time through DQUAD8.

While the main thrust of DQUAD8 is to compute quadratures on infinite intervals, one can use this stepping procedure on finite intervals also. The details are given in the APPENDIX of Folder 21 in Chapter 3.

References: Chapter 3, Folder 21, APPENDIX

2. Summary of Functions

This chapter condenses the results of Chapter 3 into a handbook format. Essential formulas and properties are displayed along with references to computer codes listed in Chapter 4 and appropriate folders in Chapter 3 where the development is detailed.

Table of Contents

(2.6) Functions $I_1(a,b,T)$ and $I_1^c(a,b,T)$

$$I_1(a,b,T)=\int_T^\infty \frac{e^{-a^2w^2}\operatorname{erf}(bw)}{w^2}dw=\frac{1}{2}\int_0^t \frac{e^{-a^2/\tau}\operatorname{erf}(b/\sqrt{\tau})}{\sqrt{\tau}}d\tau$$

$$I_1^c(a,b,T)=\int_T^\infty \frac{e^{-a^2w^2}\operatorname{erfc}(bw)}{w^2}dw=\frac{1}{2}\int_0^t \frac{e^{-a^2/\tau}\operatorname{erfc}(b/\sqrt{\tau})}{\sqrt{\tau}}d\tau$$

$$a>0,\quad b>0,\quad T=1/\sqrt{t},\quad t>0$$

(2.7) Functions $I_{13}(a,b,T)$ and $I_{13}^c(a,b,T)$

$$I_{13}(a,b,T)=\int_T^\infty \frac{e^{-a^2w^2}\operatorname{erf}(bw)}{w^3}dw=\frac{1}{2}\int_0^t e^{-a^2/u}\operatorname{erf}(b/\sqrt{u})du$$

$$I^c{}_{13}(a,b,T)=\int_T^\infty \frac{e^{-a^2w^2}\operatorname{erfc}(bw)}{w^3}dw=\frac{1}{2}\int_0^t e^{-a^2/u}\operatorname{erfc}(b/\sqrt{u})du$$

$$a>0,\quad b>0,\quad T=1/\sqrt{t},\quad t>0$$

(2.8) Functions $I_2(a,b,T)$, $I_2^c(a,b,T)$, $I_9(a,b,T)$, and $I_9^c(a,b,T)$

$$I_2(a,b,T)=\int_0^T \operatorname{erf}(aw)\operatorname{erf}(bw)dw=\frac{1}{2}\int_t^\infty \frac{\operatorname{erf}(a/\sqrt{u})\operatorname{erf}(b/\sqrt{u})}{u^{3/2}}du,$$

$$I_2^c(a,b,T)=\int_T^\infty \operatorname{erfc}(aw)\operatorname{erfc}(bw)dw=\frac{1}{2}\int_0^t \frac{\operatorname{erfc}(a/\sqrt{u})\operatorname{erfc}(b/\sqrt{u})}{u^{3/2}}du$$

$$I_9(a,b,T)=\int_0^T w\operatorname{erf}(aw)\operatorname{erf}(bw)dw=\frac{1}{2}\int_t^\infty \frac{\operatorname{erf}(a/\sqrt{u})\operatorname{erf}(b/\sqrt{u})}{u^2}du,$$

$$I_9^c(a,b,T)=\int_T^\infty w\operatorname{erfc}(aw)\operatorname{erfc}(bw)dw=\frac{1}{2}\int_0^t \frac{\operatorname{erfc}(a/\sqrt{u})\operatorname{erfc}(b/\sqrt{u})}{u^2}du$$

$$a>0,\quad b>0,\quad T=1/\sqrt{t},\quad t>0$$

(2.9) Functions $I_{19}(a,b,T)$ and $I_{19}^c(a,b,T)$

$$I_{19}(a,b,T)=\int_0^T \frac{\operatorname{erf}(aw)\operatorname{erf}(bw)}{w}dw=\frac{1}{2}\int_t^\infty \frac{\operatorname{erf}(a/\sqrt{\tau})\operatorname{erf}(b/\sqrt{\tau})}{\tau}d\tau$$

$$I_{19}^c(a,b,T)=\int_T^\infty \frac{\operatorname{erfc}(aw)\operatorname{erfc}(bw)}{w}dw=\frac{1}{2}\int_0^t \frac{\operatorname{erfc}(a/\sqrt{\tau})\operatorname{erfc}(b/\sqrt{\tau})}{\tau}d\tau$$

$$a>0,\quad b>0,\quad T=1/\sqrt{t},\quad t>0$$

(2.10) Functions $I_6(a,b,T)$ and $I_6^c(a,b,T)$ and Related Integrals

$$I_6(a,b,T)=\int_T^\infty \frac{\operatorname{erf}(aw)\operatorname{erf}(bw)}{w^2}dw=\frac{1}{2}\int_0^t \frac{1}{\sqrt{u}}\operatorname{erf}\left(\frac{a}{\sqrt{u}}\right)\operatorname{erf}\left(\frac{b}{\sqrt{u}}\right)du$$

$$I_6^c(a,b,T)=\int_T^\infty \frac{\operatorname{erfc}(aw)\operatorname{erfc}(bw)}{w^2}dw=\frac{1}{2}\int_0^t \frac{1}{\sqrt{u}}\operatorname{erfc}\left(\frac{a}{\sqrt{u}}\right)\operatorname{erfc}\left(\frac{b}{\sqrt{u}}\right)du$$

$$a>0,\quad b>0,\quad T=1/\sqrt{t},\quad t>0$$

(2.11) Functions $W_3(a,b,T)$, $W_3^c(a,b,T)$ and Related Integrals

$$W_3(a,b,T) = \int_T^\infty \frac{\mathrm{erf}(aw)\mathrm{erf}(bw)}{w^3}dw = \frac{1}{2}\int_0^t \mathrm{erf}(a/\sqrt{\tau})\mathrm{erf}(b/\sqrt{\tau})d\tau$$

$$W_3^c(a,b,T) = \int_T^\infty \frac{\mathrm{erfc}(aw)\mathrm{erfc}(bw)}{w^3}dw = \frac{1}{2}\int_0^t \mathrm{erfc}(a/\sqrt{\tau})\mathrm{erfc}(b/\sqrt{\tau})d\tau$$

$$J_\nu(a,T) = \int_T^\infty \frac{\mathrm{erf}(aw)}{w^\nu}dw\ ,\ \nu > 1\ , \qquad J_\nu^c(a,T) = \int_T^\infty \frac{\mathrm{erfc}(aw)}{w^\nu}dw\ ,\ \nu \neq 1$$

$$a > 0, \quad b > 0, \quad T = 1/\sqrt{t}, \quad t > 0$$

(2.12) Functions $I_3(a,b,c,T)$ and $J_3(a,b,c,T)$

$$I_3(a,b,c,T) = \int_T^\infty e^{-c^2w^2}\mathrm{erf}(aw)\ \mathrm{erf}(bw)dw = \frac{1}{2}\int_0^t \frac{e^{-c^2/u}}{u^{3/2}}\mathrm{erf}\left(\frac{a}{\sqrt{u}}\right)\mathrm{erf}\left(\frac{b}{\sqrt{u}}\right)du\ ,$$

$$I_3^c(a,b,c,T) \equiv J_3(a,b,c,T) = \int_T^\infty e^{-c^2w^2}\mathrm{erfc}(aw)\ \mathrm{erfc}(bw)dw = \frac{1}{2}\int_0^t \frac{e^{-c^2/u}}{u^{3/2}}\mathrm{erfc}\left(\frac{a}{\sqrt{u}}\right)\mathrm{erfc}\left(\frac{b}{\sqrt{u}}\right)du$$

$$a > 0, \quad b > 0, \quad c>0, \quad T = 1/\sqrt{t}, \quad t > 0$$

(2.13) Functions $I_4(a,b,c,T)$ and $J_4(a,b,c,T)$

$$I_4(a,b,c,T) = \int_T^\infty w^2e^{-c^2w^2}\ \mathrm{erf}(aw)\mathrm{erf}(bw)dw = \frac{1}{2}\int_0^t \frac{e^{-c^2/u}}{u^{5/2}}\mathrm{erf}\left(\frac{a}{\sqrt{u}}\right)\mathrm{erf}\left(\frac{b}{\sqrt{u}}\right)du\ ,$$

$$I_4^c(a,b,c,T) = \int_T^\infty w^2e^{-c^2w^2}\mathrm{erfc}(aw)\mathrm{erfc}(bw)dw = \frac{1}{2}\int_0^t \frac{e^{-c^2/u}}{u^{5/2}}\mathrm{erfc}\left(\frac{a}{\sqrt{u}}\right)\mathrm{erfc}\left(\frac{b}{\sqrt{u}}\right)du$$

$$J_4(a,b,c,T) = \int_T^\infty w\ e^{-c^2w^2}\mathrm{erf}(aw)\mathrm{erf}(bw)dw = \frac{1}{2}\int_0^t \frac{e^{-c^2/u}}{u^2}\mathrm{erf}\left(\frac{a}{\sqrt{u}}\right)\mathrm{erf}\left(\frac{b}{\sqrt{u}}\right)du$$

$$J_4^c(a,b,c,T) = \int_T^\infty w\ e^{-c^2w^2}\mathrm{erfc}(aw)\mathrm{erfc}(bw)dw = \frac{1}{2}\int_0^t \frac{e^{-c^2/u}}{u^2}\mathrm{erfc}\left(\frac{a}{\sqrt{u}}\right)\mathrm{erfc}\left(\frac{b}{\sqrt{u}}\right)du$$

$$a > 0, \quad b > 0, \quad c>0, \quad T = 1/\sqrt{t}, \quad t > 0$$

(2.14) Functions $I_{14}(a,b,c,T)$ and $I_{14}^c(a,b,c,T)$

$$I_{14}(a,b,c,T) = \int_T^\infty e^{-c^2w^2}\frac{\mathrm{erf}(aw)\mathrm{erf}(bw)}{w^2}dw = \frac{1}{2}\int_0^t e^{-c^2/u}\frac{\mathrm{erf}(a/\sqrt{u})\mathrm{erf}(b/\sqrt{u})}{\sqrt{u}}du$$

$$I_{14}^c(a,b,c,T) = \int_T^\infty e^{-c^2w^2}\frac{\mathrm{erfc}(aw)\mathrm{erfc}(bw)}{w^2}dw = \frac{1}{2}\int_0^t e^{-c^2/u}\frac{\mathrm{erfc}(a/\sqrt{u})\mathrm{erfc}(b/\sqrt{u})}{\sqrt{u}}du$$

$$a > 0, \quad b > 0, \quad c>0, \quad T = 1/\sqrt{t}, \quad t > 0$$

(2.15) Functions $U(a,b,t)$, $V(a,b,t)$, $I_{21}(a,b,c,t)$, $I^c_{21}(a,b,c,t)$, $J_{21}(a,b,c,t)$, and $G_n(a,b,T)$

$$U(a,b,t) = e^{a^2t+2ab}\operatorname{erfc}(a\sqrt{t} + b/\sqrt{t})$$

$$V(a,b,t) = \int_0^t U(a,b,\tau)d\tau, \quad I_{21}(a,b,c,t) = \int_0^t U(a,b,\tau)\operatorname{erf}(c/\sqrt{\tau})d\tau$$

$$I^c_{21}(a,b,c,t) = \int_0^t U(a,b,\tau)\operatorname{erfc}(c/\sqrt{\tau})d\tau \qquad J_{21}(a,b,c,t) = \int_0^t \frac{U(a,b,\tau)e^{-c^2/\tau}}{\tau^{3/2}}d\tau$$

$$a > 0, \quad b > 0, \quad c > 0, \quad t > 0$$

$$G_n(a,b,T) = \int_T^\infty \frac{e^{-a^2w^2} i^n\operatorname{erfc}(bw)}{w^n}dw, \quad a > 0, \quad b \geq 0, \quad T > 0, \quad n \geq 0$$

(2.16) Functions $I_{22}(a,b,c,t)$ and $J_{22}(a,b,c,t)$

$$I_{22}(a,b,c,t) = \int_0^t U(a,b,\tau)\frac{e^{-c^2/\tau}}{\sqrt{\tau}}d\tau,$$

$$J_{22}(a,b,c,t) = \int_0^t U(a,b,\tau)\sqrt{\tau}\, e^{-c^2/\tau}\, d\tau$$

$$U(a,b,t) = e^{a^2t+2ab}\operatorname{erfc}(a\sqrt{t} + b/\sqrt{t})$$

$$a \geq 0, \quad b \geq 0, \quad c \geq 0, \quad t > 0$$

(2.17) Functions $I_{24}(a,b,c,t)$, $I^c_{24}(a,b,c,t)$ and Related Integrals

$$I_{24}(a,b,c,t) = \int_0^t \tau U(\tau)\operatorname{erf}\left(\frac{c}{\sqrt{\tau}}\right)d\tau, \qquad I^c_{24}(a,b,c,t) = \int_0^t \tau U(\tau)\operatorname{erfc}\left(\frac{c}{\sqrt{\tau}}\right)d\tau$$

$$J_{24}(a,b,t) = \int_0^t \tau U(\tau)d\tau, \qquad V_{24}(a,b,t) = \int_0^t V(\tau)d\tau$$

$$U(t) = e^{a^2t+2ab}\operatorname{erfc}(a\sqrt{t} + b/\sqrt{t})$$

$$V(t) = \int_0^t U(\tau)d\tau = \frac{2}{a}\sqrt{\frac{t}{\pi}}e^{-b^2/t} - \left(\frac{1}{a^2} + \frac{2b}{a}\right)\operatorname{erfc}\left(\frac{b}{\sqrt{t}}\right) + \frac{U(t)}{a^2}$$

$$a > 0, \quad b > 0, \quad c > 0, \quad t > 0$$

(2.18) Function $I_{25}(a,b,c,t)$

$$I_{25}(a,b,c,d,t) = \int_0^t U(a,b,\tau)U(c,d,\tau)d\tau$$

$$U(a,b,t) = e^{a^2t+2ab}\operatorname{erfc}(a\sqrt{t} + b/\sqrt{t})$$

$$a \geq 0, \quad b \geq 0, \quad c \geq 0, \quad d \geq 0, \quad t > 0$$

(2.19) Function $I_{26}(a,b,c,d,t)$

$$I_{26}(a,b,c,d,t) = \int_0^t \tau U(a,b,\tau)U(c,d,\tau)d\tau$$

$$U(a,b,t) = e^{a^2t+2ab}\operatorname{erfc}(a\sqrt{t} + b/\sqrt{t})$$

$$a \geq 0 \quad ,b \geq 0, \quad c \geq 0, \quad d \geq 0, \quad t > 0$$

(2.20) Indefinite Integrals

$$\mathbf{J} = \int \mathbf{e}^{(a^2-b^2)x}\,\mathbf{erfc}\left(a\sqrt{x} + \frac{c}{\sqrt{x}}\right)dx$$

$$\mathbf{I} = \int x e^{(a^2-b^2)x}\,\mathbf{erfc}\left(a\sqrt{x} + \frac{c}{\sqrt{x}}\right)dx$$

(2.21) Functions $H_{23}(x)$, $I_{23}(x)$ and $J_{23}(x)$

$$H_{23}(x) = \int_0^x e^{w^2}\,\mathrm{erfc}(w)\,dw = \frac{1}{2}\int_0^{x^2} e^{v}\,\mathrm{erfc}(\sqrt{v})\frac{dv}{\sqrt{v}}$$

$$I_{23}(x) = \int_0^x e^{w^2}\,\mathrm{erfc}(w)\,w\,dw = \frac{1}{2}\int_0^{x^2} e^{v}\,\mathrm{erfc}(\sqrt{v})\,dv$$

$$J_{23}(x) = \int_0^x e^{w^2}\,\mathrm{erfc}(w)\,w^2 dw = \frac{1}{2}\int_0^{x^2} e^{v}\,\mathrm{erfc}(\sqrt{v})\sqrt{v}\,dv\,,$$

$$x \geq 0$$

(2.22) Incomplete Bessel Function

$$I(a,b,X) = \int_X^{\infty} e^{-at-b/t}\,dt\,,$$

$$a > 0,\quad b > 0,\quad X > 0$$

(2.1) Functions F(x), G(x) and Related Integrals

$$\mathbf{F(x)=\int_0^x \frac{erf(w)}{w}dw,\quad G(x)=\int_x^\infty \frac{erfc(w)}{w}dw,\quad \int_0^x e^{-a^2w^2}\ln w\, dw,\quad \int_x^\infty e^{-a^2w^2}\ln w\, dw}$$

$$\boldsymbol{a>0,\quad x>0}$$

Alternate Formulas

(2.1.1) $$F(aX)=\int_0^X \frac{\operatorname{erf}(ax)}{x}dx\ ,\quad G(aX)=\int_X^\infty \frac{\operatorname{erfc}(ax)}{x}dx\,.$$

(2.1.2) $$\int_0^T e^{-a^2w^2}\ln w dw=\frac{\sqrt{\pi}}{2a}[\operatorname{erf}(aT)\ln T-F(aT)]$$

(2.1.3) $$\int_T^\infty e^{-a^2w^2}\ln w dw=\frac{\sqrt{\pi}}{2a}[\operatorname{erfc}(aT)\ln T+G(aT)].$$

(2.1.4) $$\int_0^\infty e^{-a^2w^2}\ln w dw=-\frac{\sqrt{\pi}}{2a}\left[\frac{\gamma}{2}+\ln(2a)\right].$$

Functional Relationships

(2.1.5) $$F(X)=\frac{\gamma}{2}+\ln(2X)+G(X)\qquad (\gamma=\text{Euler Constant}=0.5772156649015328606\ldots)$$

(2.1.6) $$\int_1^\infty e^{-ax}\frac{\ln x}{\sqrt{x}}dx=-\left[\frac{\partial}{\partial\nu}E_\nu(a)\right]_{\nu=1/2}=2\sqrt{\frac{\pi}{a}}G(\sqrt{a})=G_{1/2}(a)$$

where $G_\nu(a)$ is defined in Section **(2.2)** and explored fully in Chapter 3, Folder 18.

Convergent Series

(2.1.7) $$F(X)=\frac{2X}{\sqrt{\pi}}\sum_{k=0}^\infty \frac{(-1)^k X^{2k}}{k!(2k+1)^2}$$

(2.1.8) $$G(X)=\frac{1}{2}E_1(X^2)-\frac{1}{2\pi}\sum_{k=0}^\infty \frac{C_k}{(k+1/2)}E_{k+3/2}(X^2),\qquad C_k=\frac{(1/2)_k}{k!}$$

(2.1.9) $$G(X)=\frac{1}{2}E_1(X^2)-e^{-X^2}\ln 2+\frac{X^2}{2\pi}\sum_{k=0}^\infty C_k\frac{E_{k+1/2}(X^2)}{(k+1/2)^2}$$

Accelerated Series

(2.1.10) $$G(X)=\frac{1}{2}E_1(X^2)-\frac{1}{2\pi}H(X)$$

(2.1.11) $$H(X)=\sum_{k=0}^\infty \frac{C_k}{(k+1/2)}E_{k+3/2}(X^2)$$

$$=e^{-X^2}\sum_{k=1}^n W_k(-X^2)^{k-1}+\sqrt{\pi}\, ierfc(X)\sum_{k=1}^n S_{k,0}(-X^2)^{k-1}+(-X^2)^n\sum_{k=0}^\infty S_{n,k+1}E_{k+3/2}(X^2)$$

where

$$W_1 = \sum_{k=1}^{\infty} \frac{C_k}{(k+1/2)^2} = 2\pi \ln 2 - 4, \qquad W_n = \sum_{k=1}^{\infty} S_{n,k}, \qquad n \geq 1.$$

$$S_{1,k} = \frac{C_k}{(k+1/2)^2}, \quad k \geq 0. \qquad S_{n+1,k} = \frac{S_{n,k+1}}{(k+1/2)}, \quad k \geq 0, \quad n \geq 1.$$

Explicitly,

$$S_{10} = \frac{C_o}{(1/2)^2} = 4 \ , \ S_{n,k} = \frac{(1/2)_k}{(n+k-1)!(n+k-1/2)^2}, \qquad k \geq 0, \qquad n \geq 1$$

and numerical values for the W sequence are found in Chapter 3, Folder 16.

Asymptotic Series

(2.1.12) $$F(X) = \frac{2X}{\sqrt{\pi}} \sum_{k=0}^{\infty} \frac{(-1)^k X^{2k}}{k!(2k+1)^2} \qquad \text{for } X \to 0$$

(2.1.13) $$G(X) = F(X) - \frac{\gamma}{2} - \ln(2X) \qquad \text{(See also } \mathbf{(2.1.9)}\text{)} \qquad \text{for } X \to 0$$

(2.1.14) $$\mathrm{G(X)} = \frac{1}{2X\sqrt{\pi}} \sum_{k=0}^{N} \frac{(-1)^k (1/2)_k}{X^{2k}} E_{k+3/2}(X^2) + W_N,$$

$$|W_N| \leq \frac{1}{2X\sqrt{\pi}} \frac{(1/2)_{N+1}}{X^{2N+2}} E_{N+5/2}(X^2) \qquad \text{for } X \to \infty$$

(2.1.15) $$F(X) = \frac{\gamma}{2} + \ln(2X) + G(X) \qquad \text{for } X \to \infty$$

LaPlace Transform

(2.1.16) $$\int_0^{\infty} e^{-pt} \frac{\ln t}{\sqrt{t}} dt = -\frac{\sqrt{\pi}}{\sqrt{p}} [\gamma + \ln(4p)], \qquad p > 0.$$

Computer Subroutines

F(X): DOUBLE PRECISION FUNCTION DFERF(…)
G(X): DOUBLE PRECISION FUNCTION DGERFC(…)

References : Chapter 3, Folders 6, 16 and 18

(2.2) Functions $E_\nu(x)$, $G_\nu(x)$ and Related Integrals

$$E_\nu(x) = \int_1^\infty \frac{e^{-xt}}{t^\nu} dt, \quad G_\nu(x) = \int_1^\infty \frac{E_\nu(xt)}{t^\nu} dt = -\frac{\partial E_\nu(x)}{\partial \nu} = \int_1^\infty \frac{e^{-xt}\ln t}{t^\nu} dt, \quad x > 0$$

For Integer and Half Odd Integer Orders with Application to

$$I_n(b,T) = \int_T^\infty \frac{e^{-b^2x^2}\ln x}{x^n} dx, \qquad n = 0, 1, 2, \ldots, \qquad T > 0$$

Introduction

The relation

(2.2.1) $\displaystyle\int_x^\infty \frac{E_\mu(t)}{t^\nu} dt = \frac{1}{x^{\nu-1}} \frac{E_\nu(x) - E_\mu(x)}{\mu - \nu}$, or equivalently $\displaystyle\int_1^\infty \frac{E_\mu(xt)}{t^\nu} dt = \frac{E_\nu(x) - E_\mu(x)}{\mu - \nu}$, $\mu \neq \nu$

can be verified by differentiation w.r.t. x. Notice that if $\mu \to \nu$, we get the case where $\mu = \nu$ on the left, and we define $G_\nu(x)$ by

(2.2.2) $\displaystyle G_\nu(x) \equiv \int_1^\infty \frac{E_\nu(xt)}{t^\nu} dt = \frac{-\partial E_\nu(x)}{\partial \nu} = \int_1^\infty \frac{e^{-xt}\ln t}{t^\nu} dt$ or $\displaystyle\int_x^\infty \frac{E_\nu(t)}{t^\nu} dt = \frac{1}{x^{\nu-1}} G_\nu(x)$, $\mu = \nu$

Basic Properties of $E_\nu(x)$ and $G_\nu(x)$, $\nu > 1$

Basic properties of the exponential integral

(2.2.3) $$E_\nu(x) = \int_1^\infty \frac{e^{-xt}}{t^\nu} dt, \qquad \nu \geq 0$$

can be found in most handbooks. In this section we repeat some of these properties and note the similarity of $E_\nu(x)$ and $G_\nu(x)$

(2.2.4) $$E_{\nu+1}(x) < E_\nu(x)$$

(2.2.5) $$G_{\nu+1}(x) < G_\nu(x), \qquad \nu \geq 1$$

(2.2.6) $$\frac{e^{-x}}{x+\nu} < E_\nu(x) \leq \frac{e^{-x}}{x+\nu-1}, \qquad \nu \geq 1$$

(2.2.7) $$\frac{e^{-x}}{(x+\nu)(x+\nu+1)} < \frac{E_{\nu+1}(x)}{x+\nu} < G_\nu(x) \leq \frac{E_\nu(x)}{x+\nu-1} \leq \frac{e^{-x}}{(x+\nu-1)^2}, \qquad \nu \geq 1$$

(2.2.8) $$\nu E_{\nu+1}(x) + xE_\nu(x) = e^{-x}$$

(2.2.9) $$\nu G_{\nu+1}(x) + xG_\nu(x) = E_{\nu+1}(x) \qquad \nu \geq 1$$

(2.2.10) $$E'_\nu(x) = -E_{\nu-1}(x)$$

(2.2.11) $$G'_\nu(x) = -G_{\nu-1}(x), \qquad \nu > 1$$

(2.2.12) $$E_\nu(0) = 1/(\nu-1),$$

(2.2.13) $G_\nu(0) = 1/(\nu-1)^2 \qquad \nu > 1$

Closed Forms

(2.2.14) $E_{1/2}(x) = \sqrt{\pi}\,\mathrm{erfc}\sqrt{x})/\sqrt{x}, \qquad E_{3/2}(x) = 2\sqrt{\pi}\,\mathrm{ierfc}(\sqrt{x})$

(2.2.15) $G_{1/2}(x) = 2\sqrt{\dfrac{\pi}{x}}G(\sqrt{x})$

where G(x) is described in Section **(2.1)** and Chapter 3, Folder 16.

Convergent Series

(2.2.16) $$E_1(x) = -\gamma - \ln x - \sum_{n=1}^{\infty} \frac{(-x)^n}{nn!}$$

(2.2.17) $$E_2(x) = x[\gamma + \ln x - 1] + 1 - \sum_{n=2}^{\infty} \frac{(-x)^n}{(n-1)n!}$$

(2.2.18) $$G_1(x) = \frac{\pi^2}{12} + \frac{1}{2}E_1^2(x) - (\gamma + \ln x)\sum_{n=1}^{\infty}\frac{(-x)^n}{nn!} + \sum_{n=1}^{\infty}\frac{(-x)^n}{n^2 n!} - \sum_{k=2}^{\infty}\frac{(-x)^k}{k}A_k$$

(2.2.19) $$G_2(x) = 1 - \frac{\pi^2}{12}x - \frac{x}{2}(\gamma + \ln x)^2 + x(\gamma + \ln x) - x - x\sum_{n=1}^{\infty}\frac{(-x)^n}{n^2(n+1)n!}$$

$$+ x\sum_{n=2}^{\infty}\frac{(-x)^n}{n(n+1)}A_n - \frac{x}{2}\sum_{n=2}^{\infty}\frac{(-x)^n}{n+1}D_n$$

where

$$A_n = \frac{1}{n!}\sum_{k=1}^{n-1}\frac{C_k^n}{n-k}, \qquad D_n = \frac{1}{n!}\sum_{k=1}^{n-1}\frac{C_k^n}{k(n-k)}, \qquad n \ge 2$$

and C_k^n are binomial coefficients. Two other options for $G_2(x)$ are also given in Chapter 3, Folder18.

Application of $E_\nu(x)$ and $G_\nu(x)$ to $\boldsymbol{I_n(b,T) = \int_T^{\infty} \frac{e^{-b^2x^2}\ln x}{x^n}dx, \quad n = 0, 1, 2, \ldots, \quad T > 0}$

Explicit formula

(2.2.20) $$I_n(b,T) = \frac{\ln T}{2T^{n-1}}E_{(n+1)/2}(b^2T^2) + \frac{1}{4T^{n-1}}G_{(n+1)/2}(b^2T^2)$$

For $n = 0$ the results are computed in **(2.1)**:

(2.2.21) $$I_0(b,T) = \int_T^{\infty} e^{-b^2x^2}\ln x dx = \frac{\sqrt{\pi}}{2b}[\mathrm{erfc}(bT)\ln T + G(bT)]$$

For even integers we get the G_ν function of half-odd orders:

(2.2.22) $$I_{2n}(b,T) = \frac{\ln T}{2T^{2n-1}} E_{n+1/2}(b^2T^2) + \frac{1}{4T^{2n-1}} G_{n+1/2}(b^2T^2) \qquad n \geq 0$$

For odd integers, we get the G_ν functions of integer orders,

(2.2.23) $$I_{2n+1}(b,T) = \frac{\ln T}{2T^{2n}} E_{n+1}(b^2T^2) + \frac{1}{4T^{2n}} G_{n+1}(b^2T^2) \qquad n \geq 0$$

Recurrence for $I_n(b,T)$:

(2.2.24) $$\frac{(n+1)}{2b^2} I_{n+2} + I_n = \frac{\ln T}{2b^2} \cdot \frac{e^{-b^2T^2}}{T^{n+1}} + \frac{1}{4b^2T^{n+1}} E_{(n+3)/2}(b^2T^2) \qquad n \geq 0$$

and this shows that the even and odd sequences are de-coupled. Also, the amplification factors (for the growth and/or decay of homogeneous solutions)

$$\frac{2b^2}{n+1} \text{ or } \frac{n+1}{2b^2}$$

for forward or backward recurrence show that a stable recurrence occurs (ratios less than or equal to 1) by recursion away from $N_b = [2b^2]$ or N_b+1 since one of these is an even index and one is odd.
Notice that if T=1 then,

(2.2.25) $$I_n(b,1) = \frac{1}{4} G_{(n+1)/2}(b^2) = -\frac{1}{4}\left[\frac{\partial}{\partial \nu} \mathrm{E}_\nu(\mathrm{b}^2)\right]_{\nu=(n+1)/2}$$

Stability of Recurrence Relations

It is known that the recurrence for $E_\nu(x)$ is stable if one generates $E_\nu(x)$ for the order closest to x and recurs away from this index with the two-term recurrence relation above. This pattern also applies to $G_\nu(x)$ and this is what is done to generate sequences in the subroutines for the integer and half-odd integer orders.

Computer Subroutines

The computation of $E_\nu(x)$ and $G_\nu(x)$ is described in Chapter3, Folder18. These computations result in subroutines

***E*ν(*x*): SUBROUTINE DEXINT(…)** for sequences of positive integer orders
SUBROUTINE DHEXINT(…) for sequences of positive half-odd integer orders
***G*ν(*x*): SUBROUTINE DGEXINT(…)** for sequences of positive integer orders
SUBROUTINE DGHEXINT(…) for sequences of positive half-odd integer orders

References: Chapter 3, Folders 16 and 18

(2.3) Functions I(a,b,x), J(a,b,x), U(a,b,x), V(a,b,x) and Related Integrals

$$\mathbf{I(a,b,x)} = \int_x^\infty \mathbf{e^{-a^2w^2}\,erfc(bw)dw}, \quad \mathbf{J(a,b,x)} = \int_x^\infty \mathbf{e^{-a^2w^2}\,erf(bw)dw}$$

$$\mathbf{U(a,b,x)} = \int_0^x \mathbf{e^{-a^2w^2}\,erfc(bw)dw}, \quad \mathbf{V(a,b,x)} = \int_0^x \mathbf{e^{-a^2w^2}\,erf(bw)dw}$$

$$a \ge 0, \quad b \ge 0 \quad x>0$$

Other Notations

The I, J, U, V notation is commonly used to denote other integrals. Consequently, we often added a subscript 5 to these functions to distinguish them from other notations. The subscript 5 is a designation for Folder 5 in Chapter 3. Thus, I, J, U, and V can also be I_5, J_5, U_5, and V_5.

Convergent Series For I(a,b,x)

A series expansion is developed for

(2.3.1) $$I(a,b,x) = \frac{1}{2\sqrt{\pi}\sqrt{a^2+b^2}} G(a,b,x), \quad G(a,b,x) = \sum_{k=0}^{\infty} \frac{(1/2)_k}{k!}\left(\frac{a^2}{a^2+b^2}\right)^k E_{k+3/2}(d^2x^2) \quad a \le b$$

(2.3.2) $$I(a,b,x) = \frac{be^{-a^2x^2}}{a^2+b^2}\sum_{k=1}^{\infty}\left(\frac{4a^2}{a^2+b^2}\right)^{k-1}[(k-1)!\, i^{2k-1}erfc(bx)], \qquad a \le b$$

which converges for all a and b but the convergence is best for $a \le b$. A companion relation

(2.3.3) $$I(a,b,x) = \frac{\sqrt{\pi}}{2a}\mathrm{erfc}(ax)\,\mathrm{erfc}(bx) - \frac{b}{a}I(b,a,x) \qquad a > b$$

is used for $a > b$ since $I(b,a,x)$ contains the factor $[b^2/(a^2+b^2)]^k$ and the convergence is best for $a > b$. $G(a,b,x)$ is further broken down into

(2.3.4) $$\frac{G(a,b,x)}{\sqrt{a^2+b^2}} = 2\frac{e^{-d^2x^2}}{a}\tan^{-1}\frac{a}{b} - dx^2 S(a,b,x), \qquad d^2 = a^2+b^2$$

$$S(a,b,x) = \sum_{k=0}^{\infty} \frac{(1/2)_k}{k!(k+1/2)}\left(\frac{a^2}{a^2+b^2}\right)^k E_{k+1/2}(d^2x^2), \qquad a \le b$$

to break out the dominant behavior when x is small. Both of these series converge without the restrictions $a \le b$ *or* $a > b$ but the restrictions ensure that the ratios

$$a^2/d^2 \quad or \quad b^2/d^2$$

do not exceed ½ and each series can be terminated in no more than 50 terms with errors $O(10^{-15})$. These formulae are manipulated in Chapter 3, Folder 5 to obtain forms which are suitable for computation.

Convergent Series For J(a,b,x)

Notice that

(2.3.5) $$J(a,b,x) = I(a,0,x) - I(a,b,x) = \frac{\sqrt{\pi}}{2a}\mathrm{erfc}(ax) - I(a,b,x)$$

The complete computation for numerical evaluation is given in Chapter 3, Folder 5.

Convergent Series For U(a,b,x)

Notice that

(2.3.6) $$U(a,b,x) = I(a,b,0) - I(a,b,x) = \frac{1}{a\sqrt{\pi}}\tan^{-1}\frac{a}{b} - I(a,b,x)$$

Further details of computation are given in Chapter 3, Folder 5.

Convergent Series For V(a,b,x)

Notice that

(2.3.7) $$V(a,b,x) = \frac{\sqrt{\pi}}{2a}\operatorname{erf}(ax) - U(a,b,x) = \frac{1}{a\sqrt{\pi}}\tan^{-1}\frac{b}{a} - J(a,b,x)$$

The details for a more robust numerical evaluation are given in Chapter 3, Folder 5.

Functional Relations

(2.3.8) $$aI(a,b,x) + bI(b,a,x) = \frac{\sqrt{\pi}}{2}\operatorname{erfc}(ax)\operatorname{erfc}(bx)$$

(2.3.9) $$aJ(a,b,x) + bJ(b,a,x) = \frac{\sqrt{\pi}}{2}\left[1 - \operatorname{erf}(ax)\operatorname{erf}(bx)\right] = \frac{\sqrt{\pi}}{2}\left[erfc(ax) + erf(ax)erfc(bx)\right]$$

(2.3.10) $$J(a,b,x) + I(a,b,x) = \frac{\sqrt{\pi}}{2a}\operatorname{erfc}(ax)$$

(2.3.11) $$aU(a,b,x) + bU(b,a,x) = \frac{\sqrt{\pi}}{2}\left[1 - \operatorname{erfc}(ax)\operatorname{erfc}(bx)\right] = \frac{\sqrt{\pi}}{2}\left[erf(ax) + erfc(ax)erf(bx)\right]$$

(2.3.12) $$a\,V(a,b,x) + b\,V(b,a,x) = \frac{\sqrt{\pi}}{2}erf(ax)erf(bx)$$

(2.3.13) $$U(a,b,x) + V(a,b,x) = \frac{\sqrt{\pi}}{2a}\operatorname{erf}(ax)$$

Asymptotics for $\mathbf{x} \to +\infty$

(2.3.14) $$I(a,b,x) = \frac{1}{2b\sqrt{\pi}}\sum_{k=0}^{N}\frac{(-1)^k(1/2)_k}{(bx)^{2k}}E_{k+1}\left[x^2(a^2+b^2)\right] + W_N(a,b,x)\ ,$$

$$\left|W_N(a,b,x)\right| \le \frac{1}{2b\sqrt{\pi}}\frac{(1/2)_{N+1}}{(bx)^{2N+2}}E_{N+2}\left[x^2(a^2+b^2)\right] \qquad a \le b$$

(2.3.15) $$I(a,b,x) = \frac{\sqrt{\pi}}{2a}erfc(ax)erfc(bx) - \frac{b}{2a^2\sqrt{\pi}}\sum_{k=0}^{N}\frac{(-1)^k(1/2)_k}{(ax)^{2k}}E_{k+1}\left[x^2(a^2+b^2)\right] + R_N(a,b,x)\ , \qquad a>b$$

$$\left|R_N(a,b,x)\right| \le \frac{b}{a}\left|W_N(b,a,x)\right|$$

Now, we use relations

(2.3.16) $$\mathrm{J(a,b,}x) = \int_x^\infty \mathrm{e}^{-\mathrm{a}^2 w^2} \mathrm{erf(b}\,w)\,\mathrm{d}\,w = \frac{\sqrt{\pi}}{2a} erfc(ax) - I(a,b,x),$$

(2.3.17) $$\mathrm{U(a,b,}x) = \int_0^x \mathrm{e}^{-\mathrm{a}^2 w^2} \mathrm{erfc(b}\,w)\,\mathrm{d}\,w, = I(a,b,0) - I(a,b,x) = \frac{1}{a\sqrt{\pi}} \tan^{-1}\frac{a}{b} - I(a.b.x)$$

(2.3.18) $$\mathrm{V(a,b,}x) = \int_0^x \mathrm{e}^{-\mathrm{a}^2 w^2} \mathrm{erf(b}\,w)\,\mathrm{d}\,w = \frac{\sqrt{\pi}}{2a} erf(ax) - U(a,b,x) = \frac{1}{a\sqrt{\pi}} \tan^{-1}\frac{b}{a} - J(a,b,x)$$

to get the expansions for J, U, and V.

Related Functions

A generalization of I(a,b,T)

Details relating to the manipulation of

(2.3.19) $$Y_n(a,b,T) = \int_T^\infty e^{-a^2w^2} i^n erfc(bw)dw, \qquad n \geq -1$$
$$a > 0, \quad b \geq 0$$

are found in Chapter 3, Folder 29 for the series representations:

For Case I, $a \leq b$,

(2.3.20) $$Y_n(a,b,T) = \frac{be^{-a^2T^2}}{a^2+b^2} \sum_{k=1}^{\infty} \left(\frac{4a^2}{a^2+b^2} \right)^{k-1} \frac{\Gamma(n/2+k)}{\Gamma(n/2+1)} i^{n+2k-1} erfc(bT), \qquad \mathrm{n} \geq 0$$

and for case II, a>b,

(2.3.21) $$Y_n(a,b,T) = \frac{\sqrt{\pi}}{2a} \sum_{k=0}^{n} (-1)^k \left(\frac{b}{a}\right)^k i^k erfc(aT)\, i^{n-k} erfc(bT) + (-1)^{n+1} \left(\frac{b}{a}\right)^{n+1} Y_n(b,a,T)$$

with the special case

(2.3.22) $$Y_{-1}(a,b,T) = \frac{erfc(T\sqrt{a^2+b^2})}{\sqrt{a^2+b^2}}.$$

The recurrence

(2.3.23) $$Y_{n-2} = \frac{a^2}{a^2+b^2} \left[2nY_n + \frac{b}{a^2} e^{-a^2T^2} i^{n-1} erfc(bT) \right]$$

is numerically stable on backward recurrence and is used in subroutine INTEGI29 to generate even and odd sequences of Y_n.

Integral of I(a,b,T)

Integration of $Y_0(a,b,T) = I_5(a,b,T)$ for Case I, $a \leq b$, gives

(2.3.24) $$\int_T^\infty I_5(a,b,w)dw = \frac{b}{a^2+b^2} \sum_{k=1}^{\infty} \left(\frac{4a^2}{a^2+b^2} \right)^{k-1} [(k-1)!\, Y_{2k-1}(a,b,T)], \qquad a \leq b$$

and integration of Case II, $a > b$, in Chapter3, Folder 5 gives

(2.3.25) $$\int_T^\infty I_5(a,b,w)dw = \frac{\sqrt{\pi}}{2a} \int_T^\infty erfc(aw)erfc(bw)dw - \frac{b}{a} \int_T^\infty I_5(b,a,w)dw \qquad a > b$$

where the $I_5(b,a,T)$ integral on the right is computed above since the first parameter is less than the second parameter. The integral

$$\int_T^\infty erfc(aw)erfc(bw)dw$$

is computed in Chapter3, Folder 9, but we can get an alternate form from

$$Y_1(a,b,T) = \int_T^\infty e^{-a^2w^2}\,\mathrm{ierfc(bw)dw}$$

by integration by parts:

(2.3.26) $$\int_T^\infty erfc(aw)erfc(bw)dw = \frac{1}{b}\left[erfc(aT)\,ierfc(bT) - \frac{2a}{\sqrt{\pi}}Y_1(a,b,T\right] ,$$

with

(2.3.27) $$Y_1(a,b,T) = \frac{be^{-a^2T^2}}{a^2+b^2}\sum_{k=1}^{\infty}\left(\frac{4a^2}{a^2+b^2}\right)^{k-1}\frac{\Gamma(1/2+k)}{\Gamma(3/2)}i^{2k}\,erfc(bT),\quad \frac{\Gamma(1/2+k)}{\Gamma(3/2)} = 2\,(1/2)_k,\ \ a \le b$$

For this computation we can always choose $a \le b$ since the integral is symmetric in a and b and a can be chosen to be the smaller of the two parameters.

Representations of i^nerfc(z), n = -1,0,1

Take $b = 0$ in $I(a,b,x)$:

(2.3.28) $$\frac{\sqrt{\pi}}{2a}\mathrm{erfc}(ax) = \frac{1}{2\sqrt{\pi}}\frac{1}{a}\sum_{k=0}^{\infty} C_k E_{k+3/2}(a^2x^2)\,, \qquad C_k = (1/2)_k/k!$$

(2.3.29) $$\mathrm{erfc}(z) = \frac{1}{\pi}\sum_{k=0}^{\infty} C_k E_{k+3/2}(z^2)$$

Differentiate wrt z to get

(2.3.30) $$\frac{e^{-z^2}}{z} = \frac{1}{\sqrt{\pi}}\sum_{k=0}^{\infty} C_k E_{k+1/2}(z^2)$$

Integrate erfc(z) above

(2.3.31) $$\mathrm{ierfc}(z) = \frac{1}{\sqrt{\pi}}\mathrm{erfc}(z) - \frac{z}{2\pi}\sum_{k=0}^{\infty}\frac{(1/2)_k}{k!(k+1)}E_{k+3/2}(z^2)$$

LaPlace Transforms

(2.3.32) $$\int_0^\infty e^{-pt}\frac{\mathrm{erf}(\sqrt{\alpha t})}{\sqrt{t}}dt = \frac{2}{\sqrt{\pi p}}\tan^{-1}\sqrt{\frac{\alpha}{p}}$$

(2.3.33) $$\int_0^\infty e^{-pt}\sqrt{t}\,\mathrm{erf}(\sqrt{\alpha t})dt = \frac{1}{p\sqrt{\pi}}\left[\frac{1}{\sqrt{p}}\tan^{-1}\sqrt{\frac{\alpha}{p}} + \frac{\sqrt{\alpha}}{p+\alpha}\right]$$

(2.3.34) $$\int_0^\infty e^{-pt}\frac{\mathrm{erfc}(\sqrt{\alpha t})}{\sqrt{t}}dt = \frac{2}{\sqrt{\pi p}}\tan^{-1}\sqrt{\frac{p}{\alpha}}$$

(2.3.35) $$\int_0^\infty e^{-pt}\sqrt{t}\,\mathrm{erfc}(\sqrt{\alpha t})dt = \frac{1}{p\sqrt{\pi}}\left[\frac{1}{\sqrt{p}}\tan^{-1}\sqrt{\frac{p}{\alpha}} - \frac{\sqrt{\alpha}}{p+\alpha}\right]$$

Special Cases

(2.3.36) $$I(a,b,0) = \int_0^\infty e^{-a^2w^2} erfc(bw)dw = \frac{1}{a\sqrt{\pi}} \tan^{-1}\frac{a}{b}$$

(2.3.37) $$I(a,a,x) = \frac{\sqrt{\pi}}{4a} erfc^2(ax)$$

(2.3.38) $$J(a,b,0) = \int_0^\infty e^{-a^2w^2} \operatorname{erf}(bw)dw = \frac{1}{a\sqrt{\pi}} \tan^{-1}\frac{b}{a}$$

(2.3.39) $$V(0,b,x) = \int_0^x \operatorname{erf}(bw)dw = x\operatorname{erf}(bx) - \frac{1}{b\sqrt{\pi}}(1 - e^{-b^2x^2})$$

Inequalities

A restatement of **(2.3.8)** gives the relation

(2.3.40) $$W \equiv \frac{\sqrt{\pi}}{2}\operatorname{erfc}(ax)\operatorname{erfc}(bx) = aI(a,b,x) + bI(b,a,x)$$

Divide W by $a+b$. Then the right side is a convex linear combination of $I(a,b,x)$ and $I(b,a,x)$. Therefore

(2.3.41) $$\min[I(a,b,x), I(b,a,x)] \le \frac{W}{a+b} \le \max[I(a,b,x), I(b,a,x)]$$

Divide W by $\sqrt{a^2+b^2}$ and we have the dot product of two vectors, one of which has length 1. By the Cauchy inequality, we have

(2.3.42) $$\left(\frac{W}{\sqrt{a^2+b^2}}\right)^2 \le I^2(a,b,x) + I^2(b,a,x)$$

Notice that if $b = a$, then the max and min are the same and

(2.3.43) $$I(a,a,x) = \frac{W}{2a} = \frac{\sqrt{\pi}}{4a}\operatorname{erfc}^2(ax).$$

We have similar results for J(a,b,x):

(2.3.44) $$\overline{W} \equiv \frac{\sqrt{\pi}}{2}\left[1 - \operatorname{erf}(ax)\operatorname{erf}(bx)\right] = aJ(a,b,x) + bJ(b,a,x)$$

(2.3.45) $$\min[J(a,b,x),\, J(b,a,x)] \le \frac{\overline{W}}{a+b} \le \max[J(a,b,x),\, J(b,a,x)]$$

(2.3.46) $$\left(\frac{\overline{W}}{\sqrt{a^2+b^2}}\right)^2 \le J^2(a,b,x) + J^2(b,a,x)$$

If a = b then the max and min are equal and

(2.3.47) $$J(a,a,x) = \frac{\overline{W}}{2a} = \frac{\sqrt{\pi}}{4a}\left[1 - \operatorname{erf}^2(ax)\right].$$

Because of **(2.3.11)** and **(2.3.12),** we also have similar relations for U and V.

Computer Subroutines

I(a,b,x): SUBROUTINE INTEGI5(…)
J(a,b,x): SUBROUTINE INTEGJ5(…)
V(a,b,x): SUBROUTINE INTEGV5(…)
Y_n (a,b,T) : SUBROUTINE INTEGI29(…)

References: Chapter 3, Folder 5, Chapter 3, Folder 29

(2.4) Functions $\mathbf{I_{20}(a,b,T)}$ and $\mathbf{I^c_{20}(a,b,T)}$

$$\mathbf{I_{20}(a,b,T)=\int_T^\infty e^{-a^2x^2}erf(bx)\ln x dx,\qquad I^c_{20}(a,b,T)=\int_T^\infty e^{-a^2x^2}erfc(bx)\ln x dx,}$$

$$\boldsymbol{a>0,\ \ b>0,\ \ T\ge 0}$$

Series Representations

Case I, $a\le b$

(2.4.1) $$\mathrm{I^c_{20}(a,b,T)=I_5(a,b,T)\ln T+\frac{E_1(X)}{2a\sqrt{\pi}}\tan^{-1}\left(\frac{a}{b}\right)-\frac{S(a,b,T)}{4\sqrt{\pi}\sqrt{a^2+b^2}}}\qquad a\le b$$

where $I_5(a,b,T)$ is the I function of Chapter 3, Folder 5 and

$$S(a,b,T)=\sum_{k=0}^{\infty}\frac{C_k}{(k+1/2)}\left(\frac{a^2}{a^2+b^2}\right)^k E_{k+3/2}(X),\quad X=T^2(a^2+b^2),\qquad C_k=(1/2)_k/k!,\qquad k\ge 0$$

Notice that $a^2/(a^2+b^2)\le ½$ and the convergence of the S series is better than $O(2^{-k}k^{-3/2})$.

Case II, $a>b$ The reflexive relation

(2.4.2) $$\mathrm{I^c_{19}(a,b,T)=-erfc(aT)erfc(bT)\ln T+\frac{2a}{\sqrt{\pi}}I^c_{20}(a,b,T)+\frac{2b}{\sqrt{\pi}}I^c_{20}(b,a,T)}$$

is derived in Chapter 3, Folder 20 and used to compute for $a>b$:

(2.4.3) $$\begin{aligned}\mathrm{I^c_{20}(a,b,T)}&=\mathrm{\frac{\sqrt{\pi}}{2a}[I^c_{19}(a,b,T)+erfc(aT)erfc(bT)\ln T]-\frac{b}{a}I^c_{20}(b,a,T)}\qquad a>b\\&=I_5(a,b,T)\ln T+\frac{\sqrt{\pi}}{2a}I^c_{19}(a,b,T)-\frac{b}{a}\left[\mathrm{\frac{E_1(X)}{2b\sqrt{\pi}}\tan^{-1}\left(\frac{b}{a}\right)-\frac{S(b,a,T)}{4\sqrt{\pi}\sqrt{a^2+b^2}}}\right]\end{aligned}$$

where $\mathrm{I^c_{20}(b,a,T)}$ fits into Case I because the first parameter is smaller than the second parameter. $\mathrm{I^c_{19}(a,b,T)}$ is computed in Chapter 3, Folder19. $\mathrm{I^c_{20}(0,b,T)}$ is evaluated explicitly, and the results for $\mathrm{I^c_{20}(a,0,T)}$ are given in Chapter 3, Folder 16. For I_{20}, we can use erfc(x)=1-erf(x) and get a pair from

(2.4.4) $$I_{20}(a,b,T)=\frac{\sqrt{\pi}}{2a}[\mathrm{erfc}(aT)\ln T+G(aT)]-I^c_{20}(a,b,T)$$

and $I^c_{20}(a,b,T)$ above where G(x) is computed in Section **(2.1)**. However another analysis yields the pair

(2.4.5) $$I_{20}(a,b,T)=J_5(a,b,T)\ln T+\frac{\sqrt{\pi}}{2a}G(aT)-\left[\frac{E_1(X)}{2a\sqrt{\pi}}\tan^{-1}\left(\frac{a}{b}\right)-\frac{S(a,b,T)}{4\sqrt{\pi}\sqrt{a^2+b^2}}\right],\qquad a\le b$$

and for a>b, we have

(2.4.6) $$I_{20}(a,b,T)=I_{20}(a,b,0)-\frac{\sqrt{\pi}}{2a}[\mathrm{erf}(aT)\mathrm{erf}(bT)\ln T-I_{19}(a,b,T)]+\frac{b}{a}\left[I_{20}(b,a,0)-I_{20}(b,a,T)\right].$$

Results for $T\to 0$ in $I_{20}(a,b,0)$ and $I_{20}(b,a,0)$ are given in Chapter 3, Folder 20.

Computer Subroutine

$\mathbf{I_{20}(a,b,T)}$ **and** $\mathbf{I^{c}_{20}(a,b,T)}$ **: SUBROUTINE INTEGI20(…)** on KODE=1 and KODE=2

References: Chapter 3, Folder 20

(2.5) Functions $\mathbf{P(a,b,T)}$, $\mathbf{P^c(a,b,T)}$ and $\mathbf{Q(a,b,T)}$

$$\mathbf{P(a,b,T)} = \int_T^\infty \frac{e^{-a^2w^2}\,\mathrm{erf}(bw)}{w}\,dw, \quad \mathbf{P^c(a,b,T)} = \int_T^\infty \frac{e^{-a^2w^2}\,\mathrm{erfc}(bw)}{w}\,dw,$$

$$\mathbf{Q(a,b,T)} = \int_T^\infty e^{-a^2w^2} E_1(b^2w^2)\,dw$$

$$a > 0, \quad b > 0, \quad T \ge 0$$

Series Representations

(2.5.1) $$P(a,b,T) = \begin{cases} \dfrac{1}{2}E_1(a^2T^2) - G(\sqrt{X}) - \ln\left[\dfrac{2\sqrt{a^2+b^2}}{b+\sqrt{a^2+b^2}}\right]\mathrm{erfc}(\sqrt{X}) + \dfrac{\sqrt{X}}{2\sqrt{\pi}} S_1(a,b,X), & a \le b \\[2ex] \dfrac{1}{2}E_1(a^2T^2)\,\mathrm{erf}(bT) + \ln\left[\dfrac{b+\sqrt{a^2+b^2}}{a}\right]\mathrm{erfc}(\sqrt{X}) - \dfrac{bT}{\sqrt{\pi}} S_2(b,a,X), & a > b \end{cases}$$

(2.5.2) $$Q(a,b,T) = \begin{cases} \dfrac{\sqrt{\pi}}{a}\ln\left[\dfrac{a+\sqrt{a^2+b^2}}{b}\right]\mathrm{erfc}(\sqrt{X}) - TS_2(a,b,X) & a \le b \\[2ex] \dfrac{\sqrt{\pi}}{2a}E_1(b^2T^2)\,\mathrm{erfc}(aT) - \dfrac{\sqrt{\pi}}{a}\ln\left[\dfrac{2\sqrt{a^2+b^2}}{a+\sqrt{a^2+b^2}}\right]\mathrm{erfc}(\sqrt{X}) - \dfrac{\sqrt{\pi}}{a}G(\sqrt{X}) & \\ \qquad + \dfrac{\sqrt{X}}{2a}S_1(b,a,X) & a > b \end{cases}$$

$$S_1(a,b,X) = \sum_{k=1}^{\infty} C_k \left(\frac{a^2}{a^2+b^2}\right)^k \frac{E_{k+1/2}(X)}{k}, \qquad S_2(a,b,X) = \sum_{k=0}^{\infty} \left(\frac{a^2}{a^2+b^2}\right)^k \frac{E_{k+1}(X)}{2k+1}$$

$$X = T^2(a^2+b^2), \qquad C_k = \frac{(1/2)_k}{k!}, \quad k=1,2,\ldots$$

and the computation of

(2.5.3) $$G(x) = \int_x^\infty \frac{\mathrm{erfc}(v)}{v}\,dv$$

is described in **(2.1)**. Notice that whenever S_1 and S_2 are used, the powers of $a^2/(a^2+b^2)$ or $b^2/(a^2+b^2)$ are less than (or equal to) $1/2^k$, making the convergence of these series quite acceptable numerically.

Special Cases

(2.5.4)

$$P(a,b,0) = \ln\left[\frac{b+\sqrt{a^2+b^2}}{a}\right], \qquad Q(a,b,0) = \frac{\sqrt{\pi}}{a}\ln\left[\frac{a+\sqrt{a^2+b^2}}{b}\right]$$

$$P(a,0,T) = 0, \qquad Q(0,b,T) = \frac{\sqrt{\pi}}{b}\mathrm{erfc}(bT) - TE_1(b^2T^2)$$

Functional Relations

(2.5.5) $$P(a,b,T)+P^c(a,b,T)=\frac{1}{2}E_1(a^2T^2).$$

(2.5.6) $$P(a,b,T)=\frac{1}{2}E_1(a^2T^2)\operatorname{erf}(bT)+\frac{b}{\sqrt{\pi}}Q(b,a,T).$$

To compute $P^c(a,b,T)$ from **(2.5.5)** and **(2.5.1)**, the subtractions should be done analytically to retain significant digits.

Asymptotics

For bT large, we have

(2.5.7) $$P(a,b,T)\sim\frac{1}{2}E_1(a^2T^2)-\frac{1}{2bT\sqrt{\pi}}\sum_{k=0}^{\infty}(-1)^k\frac{(1/2)_k}{(bT)^{2k}}E_{k+3/2}(X),\qquad X=T^2(a^2+b^2)$$

(2.5.8) $$P^c(a,b,T)\sim\frac{1}{2bT\sqrt{\pi}}\sum_{k=0}^{\infty}(-1)^k\frac{(1/2)_k}{(bT)^{2k}}E_{k+3/2}(X),\qquad X=T^2(a^2+b^2)$$

(2.5.9) $$Q(a,b,T)\sim\frac{T}{2(bT)^2}\sum_{k=0}^{\infty}(-1)^k\frac{(k)!}{(bT)^{2k}}E_{k+3/2}(X),\qquad X=T^2(a^2+b^2)$$

For aT large, we use **(2.5.5)** in conjunction with **(2.5.8)** to compute P(a,b,T)

(2.5.10) $$P(a,b,T)=\frac{1}{2}E_1(a^2T^2)\operatorname{erf}(bT)+\frac{b}{\sqrt{\pi}}Q(b,a,T)$$

(2.5.11) $$P(a,b,T)\sim\frac{1}{2}E_1(a^2T^2)\operatorname{erf}(bT)+\frac{b}{\sqrt{\pi}}\left[\frac{T}{2(aT)^2}\sum_{k=0}^{\infty}(-1)^k\frac{(k)!}{(aT)^{2k}}E_{k+3/2}(X)\right],\quad X=T^2(a^2+b^2)$$

(2.5.12) $$P^c(a,b,T)\sim\frac{1}{2}E_1(a^2T^2)\mathit{erfc}(bT)-\frac{b}{\sqrt{\pi}}\left[\frac{T}{2(aT)^2}\sum_{k=0}^{\infty}(-1)^k\frac{(k)!}{(aT)^{2k}}E_{k+3/2}(X)\right],\ X=T^2(a^2+b^2)$$

For large aT in Q(a,b,T), we solve for Q from **(2.5.5),** exchange parameters and use **(2.5.6),**

(2.5.13) $$Q(a,b,T)=\frac{\sqrt{\pi}}{a}\left[\ P(b,a,T)-\frac{1}{2}E_1(b^2T^2)\operatorname{erf}(aT)\ \right].$$

(2.5.14) $$Q(a,b,T)\sim\frac{\sqrt{\pi}}{a}\left[\frac{1}{2}E_1(b^2T^2)\ \mathrm{erfc(aT)}-\frac{1}{2aT\sqrt{\pi}}\sum_{k=0}^{\infty}(-1)^k\frac{(1/2)_k}{(aT)^{2k}}E_{k+3/2}(X)\right],\quad X=T^2(a^2+b^2).$$

For $a\to 0$ we get

(2.5.15) $$P(a,b,T)=\frac{1}{2}E_1(a^2T^2)-G(bT)+O(a^2)\ =\ -\frac{\gamma}{2}-\ln(aT)-G(bT)+O(a^2)\qquad \mathrm{a}\to 0$$

where G(x) is defined in **(2.1)** with its properties displayed in Chapter 3, Folders 6 and 16.

Computer Subroutines

P(a,b,T) or $\mathbf{P^c(a,b,T)}$: SUBROUTINE INTEGP(…) with KODE=1 or KODE=2
Q(a,b,T): SUBROUTINE INTEGQ(…)

References: Chapter 3, Folders 6, 11, and 16

(2.6) Functions $I_1(a,b,T)$ and $I_1^c(a,b,T)$

$$I_1(a,b,T) = \int_T^{\infty} \frac{e^{-a^2w^2}\operatorname{erf}(bw)}{w^2}dw = \frac{1}{2}\int_0^t \frac{e^{-a^2/\tau}\operatorname{erf}(b/\sqrt{\tau})}{\sqrt{\tau}}d\tau$$

$$I_1^c(a,b,T) = \int_T^{\infty} \frac{e^{-a^2w^2}\operatorname{erfc}(bw)}{w^2}dw = \frac{1}{2}\int_0^t \frac{e^{-a^2/\tau}\operatorname{erfc}(b/\sqrt{\tau})}{\sqrt{\tau}}d\tau$$

$$a>0, \quad b>0, \quad T = \frac{1}{\sqrt{t}}, \quad t>0$$

Other Notations

See the Preface for a detailed explanation of the notation used in the edition dated August, 2003. In that version, I_1 and I_1^c were defined by the second integral of the title lines. In order to refer to the first integral, the symbols $\overline{I}_1$ or $\overline{I}_1^c$ were used. In this edition, I_1 and I_1^c are defined by the first integral of the title lines and there is no $\overline{I}_1$ or $\overline{I}_1^c$.

Representations

(2.6.1) $$I_1(a,b,T) = \frac{e^{-a^2T^2}\operatorname{erf}(bT)}{T} + \frac{b}{\sqrt{\pi}}E_1(X) - 2a^2J_5(a,b,T), \qquad X = (a^2+b^2)T^2$$

(2.6.2) $$I_1(a,b,T) = \frac{e^{-a^2T^2}erf(bT)}{T} + \frac{b}{\sqrt{\pi}}E_1(X) - a\sqrt{\pi}erfc(aT) + 2a^2I_5(a,b,T)$$
$$= -\frac{e^{-a^2T^2}erfc(bT)}{T} + \frac{b}{\sqrt{\pi}}E_1(X) + \frac{\sqrt{\pi}}{T}ierfc(aT) + 2a^2I_5(a,b,T)$$

(2.6.3) $$I_1(a,b,T) = \frac{\sqrt{\pi}}{T}\operatorname{erf}(bT)\operatorname{ierfc}(aT) + \frac{b}{\sqrt{\pi}}E_1(X) - 2abI_5(b,a,T)$$

(2.6.4) $$I_1^c(a,b,T) = \frac{e^{-a^2T^2}\operatorname{erfc}(bT)}{T} - \frac{b}{\sqrt{\pi}}E_1(X) - 2a^2I_5(a,b,T)$$

(2.6.5) $$I_1^c(a,b,T) = \frac{\sqrt{\pi}}{T}\operatorname{erfc}(bT)\operatorname{ierfc}(aT) - \frac{b}{\sqrt{\pi}}E_1(X) + 2abI_5(b,a,T)$$

where J_5 and I_5 are the J and I integrals of Folder 5 in Chapter 3,

(2.6.6) $$J_5(a,b,T) = J(a,b,T) = \int_T^{\infty} e^{-a^2x^2}\operatorname{erf}(bx)dx,$$
$$I_5(a,b,T) = I(a,b,T) = \int_T^{\infty} e^{-a^2x^2}\operatorname{erfc}(bx)dx$$

Functional Relations

(2.6.7) $$I_1(a,b,T) + I_1^c(a,b,T) = \frac{\sqrt{\pi}}{T}\operatorname{ierfc}(aT)$$

Asymptotics for small t

(2.6.8) $$I_1^c(a,b,T) \sim \frac{b}{2\sqrt{\pi}} \sum_{k=0}^{N} C_k \left(\frac{t}{b^2}\right)^{k+1} E_{k+2}\left(\frac{a^2+b^2}{t}\right) + R_N, \qquad T = 1/\sqrt{t}$$

(2.6.9) $$I_1(a,b,T) = \sqrt{\pi t}\ \mathrm{ierfc}(a/\sqrt{t}) - I_1^c(a,b,T)$$

$$C_k = (-1)^k (1/2)_k, \qquad k = 0,1,\ldots \qquad |R_N| \le \frac{b}{2\sqrt{\pi}} |C_{N+1}| \left(\frac{t}{b^2}\right)^{N+2} E_{N+3}\left(\frac{a^2+b^2}{t}\right)$$

Special Cases,T>0

(2.6.10) b=0, $I_1(a,0,T) = 0$

(2.6.11) a=0, $$I_1(0,b,T) = \int_T^\infty \frac{\mathrm{erf}(bw)}{w^2} dw = \frac{\mathrm{erf}(bT)}{T} + \frac{b}{\sqrt{\pi}} E_1(b^2T^2)$$

Computer Subroutines

$\mathbf{I_1(a,b,T)}$ **or** $\mathbf{I_1^c(a,b,T)}$ **: SUBROUTINE INTEGI1(…)** with KODE=1 or KODE=2

References: Chapter 3, Folders 1, 2, and 10

(2.7) Functions $\mathbf{I_{13}(a,b,T)}$ and $\mathbf{I^c_{13}(a,b,T)}$

$$\mathbf{I_{13}(a,b,T) = \int_T^\infty \frac{e^{-a^2w^2}\,erf(bw)}{w^3}dw = \frac{1}{2}\int_0^t e^{-a^2/u} erf(b/\sqrt{u})du\,,}$$

$$\mathbf{I^c_{13}(a,b,T) = \int_T^\infty \frac{e^{-a^2w^2}\,erfc(bw)}{w^3}dw = \frac{1}{2}\int_0^t e^{-a^2/u} erfc(b/\sqrt{u})du}$$

$$\mathbf{a > 0, \quad b > 0, \quad \mathit{T} = \frac{1}{\sqrt{\mathit{t}}}, \quad t > 0}$$

Representations

(2.7.1) $$I_{13}(a,b,T) = \frac{e^{-a^2T^2}}{2T^2}\mathrm{erf}(bT) + \frac{b}{T}i\,\mathrm{erfc}(T\sqrt{a^2+b^2}) - a^2P(a,b,T)$$

(2.7.2) $$I_{13}(a,b,T) = \frac{\mathrm{erf(bT)}}{2\mathrm{T}^2}\mathrm{E}_2(\mathrm{a}^2\mathrm{T}^2) + \frac{\mathrm{b}}{2\mathrm{T}\sqrt{\pi}}\mathrm{E}_{3/2}[\mathrm{T}^2(\mathrm{a}^2+\mathrm{b}^2)] - \frac{\mathrm{a}^2\mathrm{b}}{\sqrt{\pi}}\mathrm{Q(b,a,T)}$$

where P and Q are the functions of Chapter 3, Folder11. For the complementary integral, we have

(2.7.3) $$I^c_{13}(a,b,T) = \frac{e^{-a^2T^2}}{2T^2}erfc(bT) - \frac{a^2}{2}E_1(a^2T^2) - \frac{b}{T}ierfc(T\sqrt{a^2+b^2}) + a^2P(a,b,T)$$

(2.7.4) $$\mathrm{I^c_{13}(a,b,T)} = \frac{1}{2\mathrm{T}^2}\mathrm{E}_2(\mathrm{a}^2\mathrm{T}^2)\mathrm{erfc(bT)} - \frac{\mathrm{b}}{2\mathrm{T}\sqrt{\pi}}\mathrm{E}_{3/2}[\mathrm{T}^2(\mathrm{a}^2+\mathrm{b}^2)] + \frac{\mathrm{a}^2\mathrm{b}}{\sqrt{\pi}}\mathrm{Q(b,a,T)}$$

(2.7.5) $$\mathrm{I^c_{13}(a,b,T)} = \frac{1}{2\mathrm{T}^2}\mathrm{E}_2(\mathrm{a}^2\mathrm{T}^2) - \mathrm{I_{13}(a,b,T)}$$

where $E_{3/2}(x^2) = 2\sqrt{\pi}\,ierfc(x)$ .Significant digits in **(2.7.3)** can saved if the E and P functions are combined analytically before evaluation (See **(2.5.1)**).

Computer Subroutine

$\mathbf{I_{13}(a,b,T)}$ or $\mathbf{I^c_{13}(a,b,T)}$: SUBROUTINE INTEGI13(…) with KODE=1 or KODE=2

Reference: Chapter 3, Folder13

(2.8) Functions $I_2(a,b,T)$, $I_2^c(a,b,T)$, $I_9(a,b,T)$ and $I_9^c(a,b,T)$

$$I_2(a,b,T) = \int_0^T \operatorname{erf}(aw)\operatorname{erf}(bw)dw = \frac{1}{2}\int_t^\infty \frac{\operatorname{erf}(a/\sqrt{u})\operatorname{erf}(b/\sqrt{u})}{u^{3/2}}du,$$

$$I_2^c(a,b,T) = \int_T^\infty \operatorname{erfc}(aw)\operatorname{erfc}(bw)dw = \frac{1}{2}\int_0^t \frac{\operatorname{erfc}(a/\sqrt{u})\operatorname{erfc}(b/\sqrt{u})}{u^{3/2}}du$$

$$I_9(a,b,T) = \int_0^T w\operatorname{erf}(aw)\operatorname{erf}(bw)dw = \frac{1}{2}\int_t^\infty \frac{\operatorname{erf}(a/\sqrt{u})\operatorname{erf}(b/\sqrt{u})du}{u^2}$$

$$I_9^c(a,b,T) = \int_T^\infty w\operatorname{erfc}(aw)\operatorname{erfc}(bw)dw = \frac{1}{2}\int_0^t \frac{\operatorname{erfc}(a/\sqrt{u})\operatorname{erfc}(b/\sqrt{u})du}{u^2}$$

$$a>0, \quad b>0, \quad T=\frac{1}{\sqrt{t}}, \quad t>0$$

Representations

(2.8.1) $$I_2(a,b,T) = T\operatorname{erf}(aT)\operatorname{erf}(bT) + \frac{e^{-a^2T^2}}{a\sqrt{\pi}}\operatorname{erf}(bT) + \frac{e^{-b^2T^2}}{b\sqrt{\pi}}\operatorname{erf}(aT) - \frac{\sqrt{a^2+b^2}}{ab\sqrt{\pi}}\operatorname{erf}(T\sqrt{a^2+b^2})$$

Now from Chapter 3, Folder 7, we have

(2.8.2) $$I_2^c(a,b,T)=J_3(a,b,0,T)$$
$$=\frac{e^{-a^2T^2}\operatorname{erfc}(bT)}{a\sqrt{\pi}}+\frac{e^{-b^2T^2}\operatorname{erfc}(aT)}{b\sqrt{\pi}}-T\operatorname{erfc}(aT)\operatorname{erfc}(bT)-\frac{\sqrt{a^2+b^2}}{ab\sqrt{\pi}}\operatorname{erfc}(T\sqrt{a^2+b^2})$$

and the following expression resolves the indeterminant form for small a,

(2.8.3) $$J_3(a,b,0,T) = \operatorname{erfc}(aT)ierfc(bT)/b - S(a,b,0,T)$$

$$S(a,b,0,T) = \frac{a}{\pi b}\left[\frac{\sqrt{\pi}\operatorname{erfc}(TX)}{X+b} - \frac{bT}{2X}\sum_{m=0}^\infty \frac{(3/2)_m}{(m+1)!}\left(\frac{a^2}{a^2+b^2}\right)^m E_{m+3/2}(T^2X^2)\right]$$
$$X^2 = a^2+b^2, \qquad a \le b.$$

Since $I_2^c(a,b,T)$ is symmetric in a and b, we can always take a to be the smaller of the two parameters giving rapid convergence in the series since $a^2/(a^2+b^2) \le 1/2$ and the other terms are $O(1/\sqrt{m})$.

(2.8.4) $$I_2^c(a,b,0) = J_3(a,b,0,0) = \frac{1}{a\sqrt{\pi}} + \frac{1}{b\sqrt{\pi}} - \frac{\sqrt{a^2+b^2}}{ab\sqrt{\pi}} = \frac{(2/\sqrt{\pi})}{a+b+\sqrt{a^2+b^2}}$$

The power series for $I_2(a,b,T)$ is

(2.8.5) $$I_2(a,b,T) = \int_0^T erf(aw)erf(bw)dw$$
$$= \frac{4abT^3}{\pi}\sum_{k=0}^\infty \frac{U_k(a^2,b^2)T^{2k}}{2k+3} = \frac{4T(aT)(bT)}{\pi}\sum_{k=0}^\infty \frac{U_k(a^2T^2,b^2T^2)}{2k+3}$$

where

(2.8.6) $$U_k(a^2,b^2)=\sum_{m=0}^{k} C_m(a^2)C_{k-m}(b^2), \qquad C_k(a^2)=\frac{(-1)^k a^{2k}}{k!(2k+1)}, \quad k\ge 0$$

and

(2.8.7) $I_2^c(a,b,T)=$
$$\frac{(2/\sqrt{\pi})}{a+b+\sqrt{a^2+b^2}}-T+\frac{2T(aT)}{\sqrt{\pi}}\sum_{k=0}^{\infty}\frac{C_k(a^2T^2)}{2k+2}+\frac{2T(bT)}{\sqrt{\pi}}\sum_{k=0}^{\infty}\frac{C_k(b^2T^2)}{2k+2}-\frac{4T(aT)(bT)}{\pi}\sum_{k=0}^{\infty}\frac{U_k(a^2T^2,b^2T^2)}{2k+3}.$$

A computational form for $I_2^c(a,b,T)$ for aT and bT bounded away from zero (larger parameters) is

(2.8.8) $$I_2^c(a,b,T)=\frac{T}{(aT)(bT)}\left[\frac{ierfc(T\sqrt{a^2+b^2})}{\sqrt{\pi}}-ierfc(aT)ierfc(bT)\right]$$

From Chapter3, Folder 29, we also have

(2.8.9) $$I_2^c(a,b,T)=\int_T^{\infty} erfc(aw)erfc(bw)dw=\frac{1}{b}\left[erfc(aT)\,ierfc(bT)-\frac{2a}{\sqrt{\pi}}Y_1(a,b,T\right],$$

with

(2.8.10) $$Y_1(a,b,T)=\frac{be^{-a^2T^2}}{a^2+b^2}\sum_{k=1}^{\infty}\left(\frac{4a^2}{a^2+b^2}\right)^{k-1}\frac{\Gamma(1/2+k)}{\Gamma(3/2)}i^{2k}erfc(bT), \quad \frac{\Gamma(1/2+k)}{\Gamma(3/2)}=2\,(1/2)_k, \quad a\le b$$

For this computation we can always choose $a\le b$ since the integral is symmetric in a and b and a can be chosen to be the smaller of the two parameters. This makes the convergence of $Y_1(a,b,T)$ rapid since $a^2/(a^2+b^2)\le 1/2$ and the other factors are $O(1/\sqrt{k})$.

Other representations which resolve the indeterminant forms **(2.8.1)** and **(2.8.2)** for a or b to zero are

(2.8.11) $$\begin{aligned}I_2(a,b,T)&=Terf(aT)erf(bT)+\frac{e^{-b^2T^2}}{b\sqrt{\pi}}erf(aT)-2\left(\frac{1}{a\sqrt{\pi}}+R\right)e^{-a^2T^2/2}\sinh(a^2T^2/2)\\&-Re^{-a^2T^2}erf(bT)+RT\sqrt{a^2+b^2}\,e^{-a^2T^2}\sum_{n=1}^{\infty}(-2)^n i^n erfc(bT)(bT\phi)^{n-1}\end{aligned}$$

where

(2.8.12) $$R=\frac{1}{b\sqrt{\pi}}\cdot\frac{a}{b+\sqrt{a^2+b^2}}, \qquad \mathrm{bT}\phi=\frac{a^2T}{b+\sqrt{a^2+b^2}}$$

and, without loss of generality we take $a\le b$. For $I_2^c(a,b,T)$ and $a\le b$, we have

(2.8.13) $$\begin{aligned}I_2^c(a,b,T)&=erfc(aT)\frac{ierfc(bT)}{b}-Re^{-a^2T^2}erfc(bT)\\&-RTe^{-a^2T^2}\sqrt{a^2+b^2}\sum_{n=1}^{\infty}(-2)^n i^n erfc(bT)(bT\phi)^{n-1}\end{aligned}$$

The powers in the series are less than one if (aT) is less than $\left(1+\sqrt{2}\right)$ with $a\le b$ and the series converges rapidly since

$$i^n erfc(bT)\le i^n erfc(0)=\frac{1}{2^n\Gamma(n/2+1)}, \qquad n\ge 0.$$

For $I_9(a,b,T)$ and $I_9^c(a,b,T)$, we get

(2.8.14)
$$2I_9(a,b,T) = T\cdot I_2(a,b,T) - \frac{1}{a\sqrt{\pi}}V_5(a,b,T) - \frac{1}{b\sqrt{\pi}}V_5(b,a,T)$$
$$-\frac{\sqrt{a^2+b^2}}{ab\sqrt{\pi}}\left[T\,erf(T\sqrt{a^2+b^2}) - \frac{1}{\sqrt{\pi}\sqrt{a^2+b^2}}\left(1-e^{-T^2(a^2+b^2)}\right)\right]$$

(2.8.15)
$$2I_9(a,b,T) = T^2 erf(aT)erf(bT) + \frac{Te^{-a^2T^2}erf(bT)}{a\sqrt{\pi}} + \frac{Te^{-b^2T^2}erf(aT)}{b\sqrt{\pi}}$$
$$-\frac{1}{a\sqrt{\pi}}V_5(a,b,T) - \frac{1}{b\sqrt{\pi}}V_5(b,a,T) - \frac{1}{\text{ab}\pi}\left(1-e^{-T^2(a^2+b^2)}\right)$$

(2.8.16) $$2I_9^c(a,b,T) = T\cdot I_2^c(a,b,T) + \frac{1}{a\sqrt{\pi}}I_5(a,b,T) + \frac{1}{b\sqrt{\pi}}I_5(b,a,T) - \frac{1}{ab\sqrt{\pi}}ierfc(T\sqrt{a^2+b^2})$$

and

(2.8.17)
$$2I_9^c(a,b,T) = -\mathrm{T^2erfc(aT)erfc(bT)} + \frac{\mathrm{Te^{-a^2T^2}erfc(bT)}}{\mathrm{a}\sqrt{\pi}} + \frac{\mathrm{Te^{-b^2T^2}erfc(aT)}}{\mathrm{b}\sqrt{\pi}} + \frac{1}{\mathrm{a}\sqrt{\pi}}\mathrm{I_5(a,b,T)} + \frac{1}{\mathrm{b}\sqrt{\pi}}\mathrm{I_5(b,a,T)}$$
$$-\frac{1}{\mathrm{ab}\sqrt{\pi}}\frac{\mathrm{e^{-T^2(a^2+b^2)}}}{\sqrt{\pi}} = \frac{-ierfc(aT)ierfc(bT)}{ab} + \frac{1}{a\sqrt{\pi}}I_5(a,b,T) + \frac{1}{b\sqrt{\pi}}I_5(b,a,T)$$

The V_5 and I_5 functions are the V and I functions of Chapter3, Folder 5.

The corresponding expressions for $I_9(a,b,T)$ and $I_9^c(a,b,T)$ for $a \le b$ which remove the indeterminancies for a to zero are

(2.8.18) $$2I_9(a,b,T) = T^2 erf(aT)erf(bT) + \frac{Te^{-b^2T^2}erf(aT)}{b\sqrt{\pi}} - \frac{1}{b\sqrt{\pi}}V_5(b,a,T) + U(a,b,T)$$
$$U(a,b,T) = H_1(a,b,T) + H_2(a,b,T) - W(a,b,T)\ , \qquad \mathrm{a \le b}$$

and

(2.8.19) $$2I_9^c(a,b,T) = \frac{\mathrm{Terfc(aT)}}{\mathrm{b}}ierfc(bT) + \frac{1}{\mathrm{b}\sqrt{\pi}}\mathrm{I_5(b,a,T) + W(a,b,T)}, \qquad \mathrm{a \le b}$$

where

$$H_1(a,b,T) = \frac{1}{a\sqrt{\pi}}\left[Te^{-a^2T^2} - \frac{\sqrt{\pi}}{2a}erf(aT)\right], \qquad H_2(a,b,T) = \left[\frac{1}{a^2\pi}\tan^{-1}\left(\frac{a}{b}\right) - \frac{1}{ab\pi}\right]$$

$$\mathrm{W(a,b,T)} = -\frac{\mathrm{e^{-a^2T^2}ierfc(bT)}}{\mathrm{ab}\sqrt{\pi}} + \frac{1}{\mathrm{a}\sqrt{\pi}}\mathrm{I_5(a,b,T)} = \frac{1}{\mathrm{a}\sqrt{\pi}}\left[-\frac{\mathrm{e^{-a^2T^2}ierfc(bT)}}{\mathrm{b}} + \frac{\mathrm{ierfc(T\sqrt{a^2+b^2})}}{\mathrm{d}}\right] + \mathrm{G_9(a,b,T)}$$

$$G_9(a,b,T) = \frac{1}{2\pi d^2}\sum_{k=1}^{\infty}\frac{(1/2)_k}{k!}\left(\frac{a}{d}\right)^{2k-1}E_{k+3/2}\left(d^2T^2\right), \qquad d^2 = a^2+b^2.$$

and the representations which remove the indeterminancies for a to zero are

$$H_1(a,b,T) = \frac{2\mathrm{T}^2}{\sqrt{\pi}}\sum_{k=1}^{\infty}\frac{(-1)^k(aT)^{2k-1}}{(k-1)!(2k+1)}\ , \qquad H_2(a,b,T) = \frac{1}{b^2\pi}\sum_{k=1}^{\infty}\frac{(-1)^k(a/b)^{2k-1}}{(2k+1)}, \quad \frac{\mathrm{a}}{\mathrm{b}} < 1$$

$$\mathrm{W(a,b,T)} = \frac{\mathrm{(a/b)e^{-a^2T^2}}}{\sqrt{\pi}\sqrt{\mathrm{a^2+b^2}}(b+\sqrt{\mathrm{a^2+b^2}})}\left[-\mathrm{ierfc(bT)} + \mathrm{bT}\sum_{\mathrm{n=2}}^{\infty}(-2)^{\mathrm{n-1}}\mathrm{n\ i^n erfc(bT)(bT}\phi)^{\mathrm{n-2}}\right] + \mathrm{G}_9(a,b,T)$$

Asymptotics for small and large T

For $I_2^c(a,b,T)$, **(2.8.4)** gives the case for $T \to 0$. For $I_2(a,b,T)$, **(2.8.1)** gives

(2.8.20) $$I_2(a,b,T) \sim T - \frac{\sqrt{a^2+b^2}}{ab\sqrt{\pi}} \text{ for } T \to \infty .$$

For $I_9^c(a,b,T)$ we have

(2.8.21) $$I_9^c(a,b,0) = \frac{1}{2ab\pi}\left[\frac{b}{a}\tan^{-1}\left(\frac{a}{b}\right) + \frac{a}{b}\tan^{-1}\left(\frac{b}{a}\right) - 1\right].$$

For $I_9(a,b,T)$ and $T \to \infty$,

(2.8.22) $$I_9(a,b,T) \approx \frac{1}{2}\left[T^2 - \frac{1}{a^2\pi}\tan^{-1}\left(\frac{b}{a}\right) - \frac{1}{b^2\pi}\tan^{-1}\left(\frac{a}{b}\right) - \frac{1}{ab\pi}\right], \qquad \mathrm{T} \to \infty$$

follows from **(2.8.15)** with

$$V_5(a,b,T) \approx \frac{1}{a\sqrt{\pi}}\tan^{-1}\left(\frac{b}{a}\right), \qquad \mathrm{T} \to \infty .$$

Computer Subroutines

$\mathbf{I_2(a,b,T)}$ or $\mathbf{I_2^c(a,b,T)}$: **SUBROUTINE INTEGI2(…)** with KODE=1 or KODE=2
$\mathbf{I_9(a,b,T)}$ or $\mathbf{I_9^c(a,b,T)}$: **SUBROUTINE INTEGI9(…)** with KODE=1 or KODE=2

References: Chapter 3, Folders 7, 9, and 29

(2.9) Function $\mathbf{I_{19}(a,b,T)}$ and $\mathbf{I_{19}^{c}(a,b,T)}$

$$\mathbf{I_{19}(a,b,T)=\int_0^T \frac{erf(aw)erf(bw)}{w}dw=\frac{1}{2}\int_t^\infty \frac{erf(a/\sqrt{\tau})erf(b/\sqrt{\tau})}{\tau}d\tau}$$
$$\mathbf{I_{19}^{c}(a,b,T)=\int_T^\infty \frac{erfc(aw)erfc(bw)}{w}dw=\frac{1}{2}\int_0^t \frac{erfc(a/\sqrt{\tau})erfc(b/\sqrt{\tau})}{\tau}d\tau}$$
$$\boldsymbol{a>0,\quad b>0,\quad T=1/\sqrt{t},\quad t>0}$$

Representations

(2.9.1) $$\mathrm{I_{19}^{c}(a,b,T)=G(bT)erfc(aT)-\frac{a}{2\pi\sqrt{a^2+b^2}}\sum_{k=0}^{\infty}C_k\left(\frac{a^2}{a^2+b^2}\right)^k G_{k+3/2}(X)}$$

$$+\frac{a}{\sqrt{\pi}}\ln\left[\frac{b^2}{a^2+b^2}\right]I_5(a,b,T) \qquad a\le b$$

$$+\frac{a}{2\pi\sqrt{a^2+b^2}}\sum_{k=1}^{\infty}\left(\frac{a^2}{a^2+b^2}\right)^k E_{k+3/2}(X)\sum_{m=1}^{k}\frac{C_{k-m}}{m},\qquad X=T^2(a^2+b^2).$$

where $G(X)$ is the function of Chapter 3, Folder 16; $I_5(a,b,T)$ is the I function of Chapter 3, Folder 5, $C_k = (1/2)_k/k!$, $k \ge 0$, and $G_{k+3/2}(X)$ and $E_{k+3/2}(X)$ are the functions of Chapter 3, Folder18. The convergence of the series is rapid since $a^2/(a^2+b^2)$ is at most ½.

The representation for $a > b$ is obtained by exchanging a and b since $\mathrm{I_{19}^{c}(a,b,T)}$ is symmetric in a and b. For $\mathrm{I_{19}(a,b,T)}$ we get

(2.9.2) $\mathrm{I_{19}(a,b,T)=F(aT)-[G(bT)-I_{19}^{c}(a,b,T)]_{T\to 0}+[G(bT)-I_{19}^{c}(a,b,T)]},\quad \mathrm{a\le b}$

(2.9.3) $\mathrm{I_{19}(a,b,T)=F(bT)-[G(aT)-I_{19}^{c}(b,a,T)]_{T\to 0}+[G(aT)-I_{19}^{c}(b,a,T)]},\quad \mathrm{a> b}$

and the limit for $\mathrm{T\to 0}$ is computable from $\mathrm{I_{19}^{c}}$ above:

(2.9.4) $G(bT)-I_{19}^{c}(a,b,T)=G(bT)-G(bT)erfc(aT)-R(a,b,T)=G(bT)erf(aT)-R(a,b,T)$

where R is the regular part of $\mathrm{I_{19}^{c}}$,

(2.9.5) $$\mathrm{R(a,b,T)=\frac{a}{\sqrt{\pi}}\ln\left[\frac{b^2}{a^2+b^2}\right]I_5(a,b,T)-\frac{a}{2\pi\sqrt{a^2+b^2}}\sum_{k=0}^{\infty}C_k\left(\frac{a^2}{a^2+b^2}\right)^k G_{k+3/2}(X)}$$
$$a\le b$$
$$+\frac{a}{2\pi\sqrt{a^2+b^2}}\sum_{n=1}^{\infty}\left(\frac{a^2}{a^2+b^2}\right)^n E_{n+3/2}(X)\sum_{m=1}^{n}\frac{C_{n-m}}{m},\qquad X=T^2(a^2+b^2)$$

with $C_k = (1/2)_k/k!$, $G_{k+3/2}(0)=\dfrac{1}{(k+1/2)^2}$, $E_{n+3/2}(0)=\dfrac{1}{(n+1/2)}$, $I_5(a,b,0)=\dfrac{1}{a\sqrt{\pi}}\tan^{-1}\dfrac{a}{b}$

Then the difference is

(2.9.6) $$\left[\ G(bT) - I_{19}^{c}(a,b,T)\ \right]_{T\to 0} = -R(a,b,0)\,, \qquad a \le b$$

since from Chapter 3, Folder18,

(2.9.7) $$G(bT)erf(aT) = erf(aT)\left[\ F(bT) - \frac{\gamma}{2} - \ln(2bT)\ \right] \to 0 \quad as \quad T \to 0\,.$$

For a>b we simply exchange a and b in the formulas since both I_{19} and I_{19}^{c} are symmetric in a and b.

Computer Subroutine

$\mathbf{I_{19}(a,b,T)}$ or $\mathbf{I_{19}^{c}(a,b,T)}$ **: SUBROUTINE INTEGI19(…)** with KODE=1 or KODE=2

References: Chapter 3, Folders 5, 16, 18 and 19

(2.10) Functions $I_6(a,b,T)$ and $I_6^c(a,b,T)$ and Related Integrals

$$\mathbf{I_6(a,b,T)=\int_T^\infty \frac{erf(aw)erf(bw)}{w^2}dw=\frac{1}{2}\int_0^t \frac{1}{\sqrt{u}}erf\left(\frac{a}{\sqrt{u}}\right)erf\left(\frac{b}{\sqrt{u}}\right)du},$$

$$\mathbf{I_6^c(a,b,T)=\int_T^\infty \frac{erfc(aw)erfc(bw)}{w^2}dw=\frac{1}{2}\int_0^t \frac{1}{\sqrt{u}}erfc\left(\frac{a}{\sqrt{u}}\right)erfc\left(\frac{b}{\sqrt{u}}\right)du}$$

$$\mathbf{a>0,\quad b>0,\quad} \boldsymbol{T=\frac{1}{\sqrt{t}}},\quad \mathbf{t>0}$$

Other Notations

The symbol I(a,b,T) is a common symbol for integrals and is used in Folder 6 where the development occurs. We chose to define this integral for this part of the presentation as $I_6(a,b,T)$ to make it unique.

Representations

Symmetric Form

(2.10.1)

$$\begin{aligned} I_6(a,b,T) \;=\; & \frac{\mathrm{erf}(aT)\mathrm{erf}(bT)}{T}+\frac{aE_1(a^2T^2)}{\sqrt{\pi}}\mathrm{erf}(bT) \\ & +\frac{bE_1(b^2T^2)}{\sqrt{\pi}}\mathrm{erf}(aT)+\frac{2}{\sqrt{\pi}}\left[a\ln\left(\frac{b+\sqrt{a^2+b^2}}{a}\right)+b\ln\left(\frac{a+\sqrt{a^2+b^2}}{b}\right)\right]\mathrm{erfc}(\sqrt{X}) \\ & -\frac{2abT}{\pi}\left[\sum_{k=0}^{\infty}\left(\frac{a^2}{a^2+b^2}\right)^k\frac{E_{k+1}(X)}{2k+1}+\sum_{k=0}^{\infty}\left(\frac{b^2}{a^2+b^2}\right)^k\frac{E_{k+1}(X)}{2k+1}\right], \end{aligned}$$

$T=1/\sqrt{t}$, $X=(a^2+b^2)/t=T^2(a^2+b^2)$.

Non-symmetric form for $a\le b$ is

(2.10.2)

$$\begin{aligned} I_6(a,b,T)= & \,\mathrm{erf}(aT)\left[\frac{\mathrm{erf}(bT)}{T}+\frac{b}{\sqrt{\pi}}E_1(b^2T^2)\right]+\frac{a}{\sqrt{\pi}}E_1(a^2T^2) \\ & -\frac{2a}{\sqrt{\pi}}G(\sqrt{X})+\frac{2}{\sqrt{\pi}}\left[-a\ln\left(\frac{2\sqrt{a^2+b^2}}{b+\sqrt{a^2+b^2}}\right)+b\ln\left(\frac{a+\sqrt{a^2+b^2}}{b}\right)\right]\mathrm{erfc}(\sqrt{X}) \\ & -\frac{2abT}{\pi}E_1(X)+\frac{a\sqrt{X}}{\pi}\sum_{k=1}^{\infty}\frac{(1/2)_k}{k!}\left(\frac{a^2}{a^2+b^2}\right)^k\frac{E_{k+1/2}(X)}{k} \\ & -\frac{2abT}{\pi}\sum_{k=1}^{\infty}\left(\frac{a^2}{a^2+b^2}\right)^k\frac{E_{k+1}(X)}{2k+1}, \end{aligned}$$

$T=1/\sqrt{t}$, $X=(a^2+b^2)/t=T^2(a^2+b^2)$

Non-symmetric form for $a\ge b$

Since $I_6(a,b,T)$ is symmetric in a and b, we exchange a and b above for the case $a\ge b$.

The symmetric form is computationally efficient for *a* and *b* relatively close together. When *a* and *b* are widely separated, one of the series in $a^2/(a^2+b^2)$ or $b^2/(a^2+b^2)$ can be slowly convergent. Therefore the non-symmetric form was developed for overall computational use. In the non-symmetric forms, the series in $a^2/(a^2+b^2)$ or $b^2/(a^2+b^2)$ are rapidly convergent since each ratio is at most ½. The G function is addressed in Chapter 3, Folders 6 and 16.

From Chapter 3, Folder 15, we have $I_6(a,b,T)$ in terms of the basic functions P and Q of Folder 11,

(2.10.3) $$I_6(a,b,T) = \frac{\operatorname{erf}(aT)\operatorname{erf}(bT)}{T} + \frac{2a}{\sqrt{\pi}}P(a,b,T) + \frac{2b}{\sqrt{\pi}}P(b,a,T)$$

(2.10.4) $$I_6(a,b,T) = \operatorname{erf}(aT)\left[\frac{\operatorname{erf}(bT)}{T} + \frac{b}{\sqrt{\pi}}E_1(b^2T^2)\right] + \frac{2a}{\sqrt{\pi}}P(a,b,T) + \frac{2ab}{\pi}Q(a,b,T)$$

(2.10.5) $$P(a,b,T) + P^c(a,b,T) = \frac{1}{2}E_1(a^2T^2)$$

(2.10.6) $$P(a,b,T) = \frac{1}{2}E_1(a^2T^2)\operatorname{erf}(bT) + \frac{b}{\sqrt{\pi}}Q(b,a,T)$$

(2.10.7) $$I_6(a,b,T) = \frac{erf(aT)erf(bT)}{T} + \frac{1}{\sqrt{\pi}}\left[aE_1(a^2T^2)erf(bT) + bE_1(b^2T^2)erf(aT)\right]$$
$$+\frac{2ab}{\pi}[Q(a,b,T)+Q(b,a,T)]$$

(2.10.8) $$I_6^c(a,b,T) = \frac{\mathrm{erfc(aT)erfc(bT)}}{\mathrm{T}} - \frac{2\mathrm{a}}{\sqrt{\pi}}\mathrm{P^c(a,b,T)} - \frac{2\mathrm{b}}{\sqrt{\pi}}\mathrm{P^c(b,a,T)}$$

(2.10.9) $$I_6^c(a,b,T) = \mathrm{erfc(aT)}\left[\frac{\mathrm{erfc(bT)}}{\mathrm{T}} - \frac{\mathrm{b}}{\sqrt{\pi}}\mathrm{E_1(b^2T^2)}\right] - \frac{2\mathrm{a}}{\sqrt{\pi}}\mathrm{P^c(a,b,T)} + \frac{2\mathrm{ab}}{\pi}\mathrm{Q(a,b,T)}$$

(2.10.10) $$I_6^c(a,b,T) = \frac{erfc(aT)erfc(bT)}{T} - \frac{1}{\sqrt{\pi}}\left[aE_1(a^2T^2)erfc(bT) + bE_1(b^2T^2)erfc(aT)\right]$$
$$+\frac{2ab}{\pi}[Q(a,b,T)+Q(b,a,T)]$$

__Special Cases__

(2.10.11) $$\mathrm{I_6(a,b,0)} = \int_0^\infty \frac{\mathrm{erf(aw)erf(bw)}}{\mathrm{w}^2}\mathrm{dw} = \frac{2}{\sqrt{\pi}}\left[\mathrm{a}\ln\left(\frac{\mathrm{b}+\sqrt{\mathrm{a}^2+\mathrm{b}^2}}{\mathrm{a}}\right) + \mathrm{b}\ln\left(\frac{\mathrm{a}+\sqrt{\mathrm{a}^2+\mathrm{b}^2}}{\mathrm{b}}\right)\right]$$

__Quadrature__

In Chapter 3, Folder 3, a special quadrature procedure was developed. to overcome the slow convergence of the tail of I(a,b,T). This involves using the approximation erf(x)=1 + $O(10^{-16)})$ for $x \geq 6$.The procedure is as follows:

Let $W_m = \min\left(\frac{6}{a}, \frac{6}{b}\right)$, $W_M = \max\left(\frac{6}{a}, \frac{6}{b}\right)$, $X = \min(a,b)$, $T = 1/\sqrt{t}$.

Then,

Case I: $T \leq W_m$, $$I = \int_T^{W_m} \frac{\operatorname{erf}(aw)\operatorname{erf}(bw)}{w^2}dw + P(X,W_m,W_M) + \frac{1}{W_M}$$

Case II: $W_m < T \leq W_M$, $$I = P\left(X,T,W_M\right) + \frac{1}{W_M}$$

Case III: $W_M < T < \infty$, $$I = 1/T = \sqrt{t}$$

where $$\frac{1}{W_M} + P(X, L, W_M) = \frac{erf(XL)}{L} + \frac{erfc(XW_M)}{W_M} + \frac{X}{\sqrt{\pi}}\left[E_1(X^2L^2) - E_1(X^2W_M^2)\right]$$

and the integral on $T \leq W_m$ is the quadrature part. For computation, the SLATEC library code DGAUS8 is a suitable, high accuracy quadrature code.

Computer Subroutines

$\mathbf{I_6(a,b,T)}$ **or** $\mathbf{I_6^c(a,b,T)}$: **SUBROUTINE INTEGI6(…)** with KODE=1 or K0DE=2
SUBROUTINE I6QUAD(…)

References: Chapter 3, Folders 3, 6, 11, 15

(2.11) Functions $\mathbf{W_3(a,b,T)}$ and $\mathbf{W_3^c(a,b,T)}$ and Related Integrals

$$\mathbf{W_3(a,b,T) = \int_T^\infty \frac{erf(aw)\,erf(bw)}{w^3}dw = \frac{1}{2}\int_0^t\ erf(a/\sqrt{\tau})\,erf(b/\sqrt{\tau})\,d\tau}$$

$$\mathbf{W_3^c(a,b,T) = \int_T^\infty \frac{erfc(aw)\,erfc(bw)}{w^3}dw = \frac{1}{2}\int_0^t\ erfc(a/\sqrt{\tau})\,erfc(b/\sqrt{\tau})\,d\tau}$$

$$\mathbf{J_\nu(a,T) = \int_T^\infty \frac{erf(aw)}{w^\nu}dw\ ,\ \nu > 1\ , \qquad J_\nu^c(a,T) = \int_T^\infty \frac{erfc(aw)}{w^\nu}dw\ ,\ \nu \neq 1}$$

$$\mathbf{a > 0, \quad b > 0, \quad T = \frac{1}{\sqrt{t}}, \quad t > 0}$$

Explicit Representations

(2.11.1) $$W_3(a,b,T) = \frac{\mathrm{erf}(aT)\,\mathrm{erf}(bT)}{2T^2} + \frac{1}{\sqrt{\pi}}[aI_1(a,b,T) + bI_1(b,a,T)]$$

(2.11.2) $$W_3^c(a,b,T) = \frac{\mathrm{erfc}(aT)\mathrm{erfc}(bT)}{2T^2} - \frac{1}{\sqrt{\pi}}[aI_1^c(a,b,T) + bI_1^c(b,a,T)]$$

(2.11.3) $$J_\nu(a,T) = \int_T^\infty \frac{\mathrm{erf}(aw)}{w^\nu}dw = \frac{1}{(\nu-1)T^{\nu-1}}\left[\mathrm{erf}(aT) + \frac{aT}{\sqrt{\pi}}E_{\nu/2}(a^2T^2)\right], \qquad \nu > 1$$

(2.11.4) $$J_\nu^c(a,T) = \int_T^\infty \frac{\mathrm{erfc}(aw)}{w^\nu}dw = \frac{1}{(\nu-1)T^{\nu-1}}\left[\mathrm{erfc}(aT) - \frac{aT}{\sqrt{\pi}}E_{\nu/2}(a^2T^2)\right], \qquad \nu \neq 1$$

(2.11.5) $$J_2(a,T) = \int_T^\infty \frac{\mathrm{erf}(aw)}{w^2}dw = \frac{\mathrm{erf}(aT)}{T} + \frac{a}{\sqrt{\pi}}E_1(a^2T^2), \qquad \nu = 2$$

(2.11.6) $$J_2^c(a,T) = \int_T^\infty \frac{\mathrm{erfc}(aw)}{w^2}dw = \frac{\mathrm{erfc}(aT)}{T} - \frac{a}{\sqrt{\pi}}E_1(a^2T^2), \qquad \nu = 2$$

(2.11.7) $$J_3(a,T) = \frac{\mathrm{erf}(aT)}{2T^2} + \frac{a}{T}\mathrm{ierfc}(aT) \qquad \nu = 3$$

(2.11.8) $$J_3^c(a,T) = \frac{\mathrm{erfc}(aT)}{2T^2} - \frac{a}{T}\mathrm{ierfc}(aT) = \frac{2i^2erfc(aT)}{T^2} \qquad \nu = 3$$

Functional Relations

(2.11.9) $$W_3^c(a,b,T) - W_3(a,b,T) = \frac{1}{2T^2} - J_3(a,T) - J_3(b,T) = J_3^c(a,T) - J_3(b,T)$$

(2.11.10) $$J_3(a,T) + J_3^c(a,T) = \frac{1}{2T^2}$$

(2.11.11) $$J_\nu(a,T) + J_\nu^c(a,T) = \frac{1}{(\nu-1)T^{\nu-1}} \qquad \nu > 1.$$

Computer Subroutines

$\mathbf{W_3(a,b,T)}$ **or** $\mathbf{W_3^c(a,b,T)}$ **: SUBROUTINE INTEGW3(…)** with KODE=1 or KODE=2

References Chapter 3**,** Folder 10

(2.12) Functions $\mathbf{I_3(a,b,c,T)}$ and $\mathbf{I_3^c(a,b,c,T) = J_3(a,b,c,T)}$

$$\mathbf{I_3(a,b,c,T) = \int_T^\infty e^{-c^2w^2} erf(aw)\, erf(bw)dw = \frac{1}{2}\int_0^t \frac{e^{-c^2/u}}{u^{3/2}} erf\left(\frac{a}{\sqrt{u}}\right) erf\left(\frac{b}{\sqrt{u}}\right) du},$$

$$\mathbf{I_3^c(a,b,c,T) = J_3(a,b,c,T) = \int_T^\infty e^{-c^2w^2} erfc(aw)\, erfc(bw)dw = \frac{1}{2}\int_0^t \frac{e^{-c^2/u}}{u^{3/2}} erfc\left(\frac{a}{\sqrt{u}}\right) erfc\left(\frac{b}{\sqrt{u}}\right) du}$$

$$\mathbf{a \ge 0,\ b \ge 0,\ c>0,\ T = \frac{1}{\sqrt{t}},\quad t>0}$$

Series Representations

(2.12.1) $$I_3(a,b,c,T) = \frac{\sqrt{\pi}}{2c}\mathrm{erfc}(cT) - I_5(c,a,T) - I_5(c,b,T) + J_3(a,b,c,T)$$

(2.12.2) $$J_3(a,b,c,T) = \int_T^\infty e^{-c^2w^2}\ \mathrm{erfc}(aw)\,\mathrm{erfc}(bw)dw$$

CASE I ($a,c \le b$) ($c \le a \le b$ or $a \le c \le b$)

(2.12.3) $$J_3(a,b,c,T) = \mathrm{erfc}(aT)I_5(c,b,T) - \frac{2a}{\sqrt{\pi}}\int_T^\infty e^{-a^2w^2} I_5(c,b,w)dw = \mathrm{erfc}(aT)I_5(c,b,T) - S(a,b,c,T)$$

$$S(a,b,c,T) = \frac{a}{\pi d}\left[\frac{\sqrt{\pi}\,\mathrm{erfc}(TX)}{X+d}\sum_{k=0}^\infty \frac{(1/2)_k}{(k+1)!}\left(\frac{c^2}{d^2}\right)^k P_k(y) - \frac{(TX)d}{2X^2}\sum_{m=0}^\infty \frac{(3/2)_m}{(m+1)!}E_{m+3/2}(T^2X^2)\sum_{k=0}^m \binom{m}{k}\frac{(a^2/X^2)^{m-k}(c^2/X^2)^k}{2k+1}\right]$$

$d^2(b,c) = b^2+c^2$, $X^2 = a^2+b^2+c^2$, $y = a^2/(X+d)^2$, $P_o(y) = 1$

$P_k(y) = F(-k, 1, k+2, y)$, $k \ge 1$ $a,c \le b$.

Here F(-k, 1, k+2, y) is the Gauss hypergeometric function, which for this application, can be computed with the series

(2.12.4) $$F(a,b,c,z) = \sum_{m=0}^\infty \frac{(a)_m(b)_m}{m!\,(c)_m}z^m,\qquad (a)_0 = 1,\quad (a)_m = (a)(a+1)...(a+m-1),\quad m \ge 1$$

With the first parameter a negative integer(-k), this F function is a polynomial of degree k since $(-k)_m = (-k)(-k+1)...(-k+m-1)$ is zero for m=k+1 and higher indices. I_5 is the I function of Chapter 3, Folder5.

An alternate formulation of this case is presented in Chapter 3, Folder 28 where S(a,b,c,T) is computed by the series

(2.12.5) $$S(a,b,c,T) = \frac{2a}{\sqrt{\pi}}\int_T^\infty e^{-a^2w^2} I_5(c,b,w)dw = \frac{a}{\pi d}\sum_{k=0}^\infty C_k\left(\frac{c^2}{d^2}\right)^k R_k(a,d,T),$$

where

$$R_k(a,d,T) = \int_T^\infty e^{-a^2w^2}E_{k+3/2}(d^2w^2)dw,\qquad k \ge 0,$$

$$d^2 = b^2 + c^2 \qquad C_k = (1/2)_k/k!.$$

Here $R_k(a,d,T)$ is expanded in the form

$$R_k(a,d,T) = \sqrt{\pi}\sum_{m=1}^{k+1} A(k+1,m)Y_{2m-1}(a,d,T)$$

where $Y_{2m-1}(a,d,T)$ is computed by backward recurrence on the two term relation

$$Y_{n-2} = \frac{a^2}{a^2+d^2}\left[2nY_n + \frac{d}{a^2}e^{-a^2T^2}i^{n-1}erfc(dT)\right]$$

starting at some large *odd* index N. The details of computation as well as the coefficients A(k+1,m) are given in Chapter 3, Folder 28. Another more desirable form for S is presented in Chapter 3, Folder 29:

(2.12.6) $$S(a,b,c,T) = \frac{2ab}{\sqrt{\pi}\sqrt{b^2+c^2}}\sum_{k=1}^{\infty}\left(\frac{4c^2}{b^2+c^2}\right)^{k-1}[(k-1)!Y_{2k-1}(\sqrt{a^2+c^2},b,T)]\,, \qquad a\,,\ c \le b$$

where sequences of the integral

$$Y_n(a,b,T) = \int_T^{\infty} e^{-a^2w^2}i^n erfc(bw)dw, \qquad a > 0, \quad b \ge 0, \quad n \ge -1$$

are computed by a series in conjunction with the recurrence above in subroutine INTEGI29.

CASE II ($a \le b \le c$)

(2.12.7) $$J_3(a,b,c,T) = \frac{\sqrt{\pi}}{2c}\mathrm{erfc}(aT)\,\mathrm{erfc}(bT)\,\mathrm{erfc}(cT) - \frac{a}{c}J_3(b,c,a,T) - \frac{b}{c}J_3(a,c,b,T)$$

Notice that c is larger than a and b. This is precisely the criteria to compute $J_3(b,c,a,T)$ and $J_3(a,c,b,T)$ by Case I. This relation is analogous to

(2.12.8) $$I_5(a,b,T) = \frac{\sqrt{\pi}}{2a}\mathrm{erfc}(aT)\,\mathrm{erfc}(bT) - \frac{b}{a}I_5(b,a,T)$$

in Chapter 3, Folder 5 when $a \ge b$.

Power Series for small aT, bT, cT

(2.12.9) $$I_3(a,b,c,T) = I_3(a,b,c,0) - \int_0^T e^{-c^2w^2}\,\mathrm{erf}(aw)\,\mathrm{erf}(bw)dw$$

And using the power series for each erf function, we get

Case I, *a, c* $\le b$

(2.12.10) $$I_3(a,b,c,0) = \frac{1}{c\sqrt{\pi}}\tan^{-1}\frac{a}{c} - S(a,b,c,0)$$

$$S(a,b,c,0) = \frac{a}{d\sqrt{\pi}}\cdot\frac{1}{X+d}\sum_{k=0}^{\infty}\frac{(1/2)_k}{(k+1)!}\left(\frac{c^2}{d^2}\right)^k P_k(y)$$

where $d^2 = b^2+c^2$, $X^2 = a^2+b^2+c^2$, $y = a^2/(X+d)^2$, $P_0(y) = 1$, $P_k(y) = F(-k, 1, k+2, y)$, $k \ge 1$ for a, $c \le b$ and $a = \min(a,b)$, $b = \max(a,b)$. In Case II, c takes the place of b as the dominant parameter.

Case II, $a \le b \le c$

(2.12.11) $$I_3(a,b,c,0) = \frac{a}{c}S(b,c,a,0) + \frac{b}{c}S(a,c,b,0)$$

Now the power series for the integral is

(2.12.12) $$\int_0^T e^{-c^2w^2}\operatorname{erf}(aw)\operatorname{erf}(bw)dw = \frac{4abT^3}{\pi}\sum_{k=0}^{\infty}\frac{V_kT^{2k}}{2k+3}$$

$$V_kT^{2k} = \sum_{n=0}^{k} D_n(c^2T^2)U_{k-n}(a^2T^2,b^2T^2), \quad U_k(a^2T^2,b^2T^2) = \sum_{m=0}^{k} C_m(a^2T^2)C_{k-m}(b^2T^2)$$

$$C_k(x^2) = \frac{(-1)^k x^{2k}}{k!(2k+1)}, \qquad D_k(x^2) = \frac{(-1)^k x^{2k}}{k!}$$

Series Expansion for Large c

(2.12.13)

$$\int_T^{\infty} e^{-c^2w^2}\operatorname{erf}(aw)\operatorname{erf}(bw)dw == \frac{2ab}{\pi c^3}\sum_{k=0}^{\infty}(-1)^k Y_{k+1}(c^2T^2)\frac{\Gamma(k+3/2)}{\Gamma(k+1)}\sum_{m=0}^{k}\binom{k}{m}\frac{(a^2/c^2)^m}{2m+1}\cdot\frac{(b^2/c^2)^{k-m}}{2(k-m)+1}$$

where

(2.12.14) $Y_k(x) = \Gamma(k+1/2, x)/\Gamma(k+1/2)$.

Then

(2.12.15) $$Y_{k+1} = Y_k + \frac{e^{-x}x^{k+1/2}}{\Gamma(k+3/2)} = Y_k + T_k, \qquad k = 0,1,\ldots$$

where $$T_{k+1} = T_k \cdot \frac{x}{(k+3/2)} \text{ with } T_o = \frac{e^{-x}\sqrt{x}}{(\sqrt{\pi}/2)} \qquad \text{and} \qquad Y_o = \operatorname{erfc}(\sqrt{x})$$

Because of the way in which the terms interact, the condition

(2.12.16) $$\frac{a^2+b^2}{c^2} \le \frac{1}{2p} \quad \text{with} \quad p = \begin{cases} 1 & \text{for} \quad c^2T^2 \le 5 \\ c^2T^2 - 4 & \text{for} \quad c^2T^2 > 5 \end{cases}$$

was determined experimentally to give accurate results. Notice also that the binomial coefficients can be generated very rapidly and efficiently by additions using the Pascal triangle relation
$B_{k+1,m} = B_{k,m} + B_{k,m-1}, \quad m = 1,k.$

Quadrature for $\mathbf{I_3(a,b,c,T)}$

For the quadrature, we follow the procedure of Chapter 3, Folder 3 by replacing the erf functions with 1 when the arguments exceed 6. As noted, the error is uniformly $O(10^{-15})$ for $\operatorname{erf}(x) = 1$ when $x \ge 6$. Then, we have the same cases as in Folder 3:

Let $W_m = \min\left(\frac{6}{a},\frac{6}{b}\right)$, $\quad W_M = \max\left(\frac{6}{a},\frac{6}{b}\right)$, $\quad X = \min(a,b)$, and $\quad T = \frac{1}{\sqrt{t}}$:

Case I: $1/\sqrt{t} = T \le W_m$

(2.12.17) $$I_3(a,b,c,T) = \int_T^{W_m} e^{-c^2w^2}\,\mathrm{erf}(aw)\,\mathrm{erf}(bw)dw + J_5(c,X,W_m) + I_5(c,X,W_M)$$

Case II: $W_m < T \le W_M$

(2.12.18) $$I_3(a,b,c,T) = J_5(c,X,T) + I_5(c,X,W_M)$$

Case III: $W_M < T < \infty$

(2.12.19) $$I_3(a,b,c,T) = \frac{\sqrt{\pi}}{2c}\mathrm{erfc}(cT)$$

where $J_5(a,b,x)$ and $I_5(a,b,x)$ are the I functions of Chapter3, Folder5.

Special Cases

$c = 0$ (Chapter 3, Folder 9 for $\int_0^T \mathrm{erf}(aw)\,\mathrm{erf}(bw)dw$)

(2.12.20) $$J_3(a,b,0,T) = \int_T^{\infty} \mathrm{erfc}(aw)\,\mathrm{erfc}(bw)dw$$

$$= \frac{\mathrm{erfc}(aT)\mathrm{ierfc}(bT)}{b} + \frac{e^{-a^2T^2}\,\mathrm{erfc}(bT)}{a\sqrt{\pi}} - \frac{\sqrt{a^2+b^2}}{ab\sqrt{\pi}}\mathrm{erfc}(T\sqrt{a^2+b^2})$$

a=0 and c=0

(2.12.21) $$J_3(0,b,0,T) = \int_T^{\infty} \mathrm{erfc}(bw)dw = \frac{\mathrm{ierfc}(bT)}{b}$$

c=0 and T=0

(2.12.22) $$J_3(a,b,0,0) = \int_0^{\infty} \mathrm{erfc}(aw)\,\mathrm{erfc}(bw)dw = \frac{(2/\sqrt{\pi})}{a+b+\sqrt{a^2+b^2}}$$

a=0, c=0, T=0

(2.12.23) $$J_3(0,b,0,0) = \frac{1}{b\sqrt{\pi}} \text{ since } i\mathrm{erfc}(0) = 1/\sqrt{\pi}$$

Inequalities

We derived the relation

(2.12.24) $$W \equiv \frac{\sqrt{\pi}}{2}\mathrm{erfc}(aT)\,\mathrm{erfc}(bT)\,\mathrm{erfc}(cT) = aJ_3(b,c,a,T) + bJ_3(a,c,b,T) + cJ_3(a,b,c,T)$$

Following the derivation in the APPENDIX of Chapter 3, Folder 5, divide both sides by $a+b+c$ and we get a convex linear combination of the J_3's. Then

(2.12.25) $\min[J_3(b,c,a,T), J_3(a,c,b,T), J_3(a,b,c,T)] \le \dfrac{W}{a+b+c} \le \max[J_3(b,c,a,T), J_3(a,c,b,T), J_3(a,b,c,T)]$

Also, if we divide by $\sqrt{a^2+b^2+c^2}$ we have the dot product of two vectors on the right, one of which has length 1. By the Cauchy Inequality, we get

(2.12.26) $$\left(\frac{W}{\sqrt{a^2+b^2+c^2}}\right)^2 \le J_3^2(b,c,a,T) + J_3^2(a,c,b,T) + J_3^2(a,b,c,T)$$

Notice that if $c = b = a$, then the max and min are the same and $J_3(a,a,a,T) = \dfrac{W}{3a} = \dfrac{\sqrt{\pi}}{6a}\operatorname{erfc}^3(aT)$ as it should be: $\int_T^\infty e^{-a^2w^2}\operatorname{erfc}^2(aw)dw = \dfrac{\sqrt{\pi}}{6a}\operatorname{erfc}^3(aT)$ since $-\dfrac{2ae^{-a^2w^2}}{\sqrt{\pi}}dw$ is the differential of erfc(aw).

Numerical Considerations

In Chapter 3, Folder 7f, leading terms of each expression are combined to reduce losses of significance by small differences of large numbers. These are implemented in the computer subroutine INTEGI3.

Computer Subroutine

$\mathbf{I_3(a,b,c,T)}$ **or** $\mathbf{I_3^c(a,b,c,T)}$ **: SUBROUTINE INTEGI3(…)** with KODE=1 or KODE=2

References: Chapter 3, Folder 7, Folder 28, Folder 29

(2.13) Functions $I_4(a,b,c,T), I_4^c(a,b,c,T), J_4(a,b,c,T)$ and $J_4^c(a,b,c,T)$

$$I_4(a,b,c,T) = \int_T^\infty w^2 e^{-c^2w^2} \operatorname{erf}(aw)\operatorname{erf}(bw)dw = \frac{1}{2}\int_0^t \frac{e^{-c^2/u}}{u^{5/2}} \operatorname{erf}\left(\frac{a}{\sqrt{u}}\right)\operatorname{erf}\left(\frac{b}{\sqrt{u}}\right)du,$$

$$I_4^c(a,b,c,T) = \int_T^\infty w^2 e^{-c^2w^2} \operatorname{erfc}(aw)\operatorname{erfc}(bw)dw = \frac{1}{2}\int_0^t \frac{e^{-c^2/u}}{u^{5/2}} \operatorname{erfc}\left(\frac{a}{\sqrt{u}}\right)\operatorname{erfc}\left(\frac{b}{\sqrt{u}}\right)du$$

$$J_4(a,b,c,T) = \int_T^\infty w\, e^{-c^2w^2} \operatorname{erf}(aw)\operatorname{erf}(bw)dw = \frac{1}{2}\int_0^t \frac{e^{-c^2/u}}{u^2} \operatorname{erf}\left(\frac{a}{\sqrt{u}}\right)\operatorname{erf}\left(\frac{b}{\sqrt{u}}\right)du$$

$$J_4^c(a,b,c,T) = \int_T^\infty w\, e^{-c^2w^2} \operatorname{erfc}(aw)\operatorname{erfc}(bw)dw = \frac{1}{2}\int_0^t \frac{e^{-c^2/u}}{u^2} \operatorname{erfc}\left(\frac{a}{\sqrt{u}}\right)\operatorname{erfc}\left(\frac{b}{\sqrt{u}}\right)du$$

$$a \ge 0, \quad b \ge 0, \quad c>0, \quad T = \frac{1}{\sqrt{t}}, \quad t > 0$$

Series Representations

For $I_4(a,b,c,T)$ and $I_4(a,b,c,T)$ we have

(2.13.1) $$I_4(a,b,c,T) = \frac{Te^{-c^2T^2}\operatorname{erf}(aT)\operatorname{erf}(bT)}{2c^2} + \frac{1}{2c^2\sqrt{\pi}}\left\{\frac{ae^{-(a^2+c^2)T^2}\operatorname{erf}(bT)}{(a^2+c^2)} + \frac{be^{-(b^2+c^2)T^2}\operatorname{erf}(aT)}{(b^2+c^2)}\right\}$$
$$+\frac{ab}{2c^2\sqrt{\pi}}\left\{\frac{1}{a^2+c^2}+\frac{1}{b^2+c^2}\right\}\frac{\operatorname{erfc}(TX)}{X} + \frac{1}{2c^2}I_3(a,b,c,T)$$

(2.13.2) $$I_4^c(a,b,c,T) = \frac{Te^{-c^2T^2}erfc(aT)erfc(bT)}{2c^2} - \frac{1}{2c^2\sqrt{\pi}}\left\{\frac{ae^{-(a^2+c^2)T^2}erfc(bT)}{(a^2+c^2)} + \frac{be^{-(b^2+c^2)T^2}erfc(aT)}{(b^2+c^2)}\right\}$$
$$+\frac{ab}{2c^2\sqrt{\pi}}\left\{\frac{1}{a^2+c^2}+\frac{1}{b^2+c^2}\right\}\frac{\operatorname{erfc}(TX)}{X} + \frac{1}{2c^2}I_3^c(a,b,c,T)$$

where $X^2 = a^2+b^2+c^2$ and $I_3(a,b,c,T)$ and $I_3^c(a,b,c,T)$ are the functions of **(2.12)** and Chapter 3, Folder 7. For $b \to \infty$ and b=0,

(2.13.3) $$I_4(a,\infty,c,T) = \int_T^\infty w^2 e^{-c^2w^2} erf(aw)dw = \frac{1}{2c^2}\left[Te^{-c^2T^2}erf(aT) + \frac{a}{a^2+c^2}\frac{e^{-(a^2+c^2)T^2}}{\sqrt{\pi}} + J_5(c,a,T)\right]$$

(2.13.4) $$I_4^c(a,0,c,T) = \int_T^\infty w^2 e^{-c^2w^2} erfc(aw)dw = \frac{1}{2c^2}\left[Te^{-c^2T^2}erfc(aT) - \frac{a}{a^2+c^2}\frac{e^{-(a^2+c^2)T^2}}{\sqrt{\pi}} + I_5(c,a,T)\right]$$

where the J_5 and I_5 functions are the functions of **(2.3)** which are computed in Chapter 3, Folder5. Using the symmetry in a and b we can relate these functions by

(2.13.5) $$I_4(a,b,c,T) = \int_T^\infty w^2 e^{-c^2w^2}dw - I_4^c(a,0,c,T) - I_4^c(0,b,c,T) + I_4^c(a,b,c,T)$$

(2.13.6) $$I_4^c(a,b,c,T) = \int_T^\infty w^2 e^{-c^2w^2}dw - I_4(a,\infty,c,T) - I_4(\infty,b,c,T) + I_4(a,b,c,T)$$

where

$$\int_T^\infty w^2 e^{-c^2w^2}dw = \frac{1}{2c^2}\left[Te^{-c^2T^2} + \frac{\sqrt{\pi}}{2c}erfc(cT)\right].$$

For $J_4(a,b,c,T)$ and $J_4^c(a,b,c,T)$ we have

(2.13.7) $$J_4(a,b,c,T)=\frac{e^{-c^2T^2}}{2c^2}\operatorname{erf}(aT)\operatorname{erf}(bT)+\frac{1}{c^2\sqrt{\pi}}\left[bJ_5(\sqrt{b^2+c^2},a,T)+aJ_5(\sqrt{a^2+c^2},b,T)\right]$$

(2.13.8) $$J_4^c(a,b,c,T)=\frac{e^{-c^2T^2}}{2c^2}erfc(aT)erfc(bT)-\frac{1}{c^2\sqrt{\pi}}\left[bI_5(\sqrt{b^2+c^2},a,T)+aI_5(\sqrt{a^2+c^2},b,T)\right]$$

where $J_5(a,b,T)$ and $I_5(a,b,T)$ are the functions of **(2.3)** and Chapter 3, Folder 5. For $b\to\infty$ and b=0,

(2.13.9) $$J_4(a,\infty,c,T)=\frac{1}{2c^2}\left[e^{-c^2T^2}erf(aT)+\frac{a}{\sqrt{a^2+c^2}}erfc(T\sqrt{a^2+c^2})\right]$$

(2.13.10) $$J_4^c(a,0,c,T)=\frac{1}{2c^2}\left[e^{-c^2T^2}erfc(aT)-\frac{a}{\sqrt{a^2+c^2}}erfc(T\sqrt{a^2+c^2})\right]$$

and again using the symmetry in a and b we have the relations

(2.13.11) $$J_4(a,b,c,T)=\frac{1}{2c^2}e^{-c^2T^2}-J_4^c(a,0,c,T)-J_4^c(0,b,c,T)+J_4^c(a,b,c,T)$$

(2.13.12) $$J_4^c(a,b,c,T)=\frac{1}{2c^2}e^{-c^2T^2}-J_4(a,\infty,c,T)-J_4(\infty,b,c,T)+J_4(a,b,c,T)\ .$$

<u>Quadrature for $\mathbf{I_4(a,b,c,T)}$</u>

The outline for this procedure is set in Chapter 3, Folder3 where we replace erf(x) function with 1 when $x\ge 6$ according to the estimate erf(x)=1 + $O(10^{-16})$ and integrate analytically to estimate the tail when c is small. Then the procedure is:

Let $W_m=\min\left(\frac{6}{a},\frac{6}{b}\right)$, $W_M=\max\left(\frac{6}{a},\frac{6}{b}\right)$ $X=\min(a,b)$, $T=\frac{1}{\sqrt{t}}$

<u>Case I</u>: $T\le W_m$

(2.13.13) $$I_4(a,b,c,T)=\int_T^{W_m}w^2e^{-c^2w^2}\operatorname{erf}(aw)\operatorname{erf}(bw)dw+P(c,X,W_m,W_M)$$

<u>Case II</u>: $W_m<T\le W_M$

(2.13.14) $$I_4(a,b,c,T)=P(c,X,T,W_M)$$

<u>Case III</u>: $W_M<T<\infty$

(2.13.15) $$I_4(a,b,c,T)=\frac{1}{2c^2}\left[Te^{-c^2T^2}+\frac{\sqrt{\pi}}{2c}\operatorname{erfc}(cT)\right]$$

where

(2.13.16) $$P(c,X,L,U)=\frac{1}{2c^2}\left\{Le^{-c^2L^2}\operatorname{erf}(XL)+Ue^{-c^2U^2}\operatorname{erfc}(XU)+\frac{\sqrt{\pi}}{2c}\operatorname{erfc}(cL)\right.$$
$$\left.+\frac{(X/\sqrt{\pi})}{X^2+c^2}\left[e^{-(X^2+c^2)L^2}-e^{-(X^2+c^2)U^2}\right]-\left[I_5(c,X,L)-I_5(c,X,U)\right]\right\}$$

and I_5 is the integral I in Folder 5.

<u>Quadrature for</u> $\mathbf{J_4(a,b,c,T)}$

The outline for this procedure is set in Chapter 3, Folder 3 where we replace erf(x) function with 1 when $x \ge 6$ according to the estimate erf(x)=1 + $O(10^{-16})$. Then the procedure is:

Let $\quad W_m = \min\left(\frac{6}{a},\frac{6}{b}\right), \quad W_M = \max\left(\frac{6}{a},\frac{6}{b}\right) \quad X = \min(a,b), \quad T = \frac{1}{\sqrt{t}}$

Then,

<u>Case I</u>, $\quad T \le W_m$

(2.13.17) $$J_4(a,b,c,T) = \int_T^{W_m} we^{-c^2w^2}\,\mathrm{erf}(aw)\,\mathrm{erf}(bw)\,dw + S(c,X,W_m,W_M) + \frac{1}{2c^2}e^{-c^2W_M^2}$$

<u>Case II</u>, $\quad W_m < T \le W_M$

(2.13.18) $$J_4(a,b,c,T) = S(c,X,T,W_M) + \frac{1}{2c^2}e^{-c^2W_M^2}$$

<u>Case III</u>, $\quad W_M < T < \infty$

(2.13.19) $$J_4(a,b,c,T) = \frac{1}{2c^2}e^{-c^2T^2}$$

where

(2.13.20) $$\frac{1}{2c^2}e^{-c^2W_M^2} + S(c,X,L,W_M) = \frac{1}{2c^2}\left[e^{-c^2L^2}\,erf(XL) + e^{-c^2W_M^2}\,erfc(XW_M)\right] + \frac{X}{2c^2\sqrt{c^2+X^2}}\left[erfc(L\sqrt{c^2+X^2}) - erfc(W_M\sqrt{c^2+X^2})\right]$$

<u>Computer Subroutines</u>

$\mathbf{I_4(a,b,c,T)}$: **SUBROUTINE I4QUAD(…)**
SUBROUTINE I4SER(…)
$\mathbf{J_4(a,b,c,T)}$: **SUBROUTINE J4QUAD(…)**
SUBROUTINE J4SER(…)

References: Chapter 3, Folder 8

(2.14) Functions $\mathbf{I_{14}(a,b,c,T)}$ and $\mathbf{I^c_{14}(a,b,c,T)}$

$$\mathbf{I_{14}(a,b,c,T) = \int_T^\infty e^{-c^2w^2}\frac{erf(aw)erf(bw)}{w^2}dw = \frac{1}{2}\int_0^t e^{-c^2/u}\frac{erf(a/\sqrt{u})erf(b/\sqrt{u})}{\sqrt{u}}du},$$

$$\mathbf{I^c_{14}(a,b,c,T) = \int_T^\infty e^{-c^2w^2}\frac{erfc(aw)erfc(bw)}{w^2}dw = \frac{1}{2}\int_0^t e^{-c^2/u}\frac{erfc(a/\sqrt{u})erfc(b/\sqrt{u})}{\sqrt{u}}du}$$

$$\mathbf{a>0,\quad b>0,\quad \mathit{T}=\frac{1}{\sqrt{\mathit{t}}}\quad t>0}$$

Series Representation

(2.14.1) $I_{14}(a,b,c,T) = \dfrac{I_2(a,b,T)\ e^{-c^2T^2}}{T^2} +$

$$-2c^2\left[I_3(a,b,c,T) + \frac{1}{a\sqrt{\pi}}P(\sqrt{a^2+c^2},b,T) + \frac{1}{b\sqrt{\pi}}P(\sqrt{b^2+c^2},a,T) - \frac{\sqrt{a^2+b^2}}{ab\sqrt{\pi}}P(c,\sqrt{a^2+b^2},T)\right]$$

$$-2\left[\frac{1}{a\sqrt{\pi}}I_{13}(\sqrt{a^2+c^2},b,T) + \frac{1}{b\sqrt{\pi}}I_{13}(\sqrt{b^2+c^2},a,T) - \frac{\sqrt{a^2+b^2}}{ab\sqrt{\pi}}I_{13}(c,\sqrt{a^2+b^2},T)\right]$$

where from Chapter3, Folder 9

(2.14.2) $I_2(a,b,T) = T\,\mathrm{erf}(aT)\,\mathrm{erf}(bT) + \dfrac{e^{-a^2T^2}}{a\sqrt{\pi}}\mathrm{erf}(bT) + \dfrac{e^{-b^2T^2}}{b\sqrt{\pi}}\mathrm{erf}(aT) - \dfrac{\sqrt{a^2+b^2}}{ab\sqrt{\pi}}\mathrm{erf}(T\sqrt{a^2+b^2})$,

$I_{13}(a,b,T)$ is computed in Chapter 3, Folder 13, $P(a,b,T)$ is computed in Chapter 3, Folder 11, and $I_3(a,b,c,T)$ is computed in Chapter 3, Folder 7. Similarly for $I^c_{14}(a,b,c,T)$,

(2.14.3) $I^c_{14}(a,b,c,T) = -\dfrac{I^c_2(a,b,T)\ e^{-c^2T^2}}{T^2} +$

$$+2c^2\left[-I^c_3(a,b,c,T) + \frac{1}{a\sqrt{\pi}}P^c(\sqrt{a^2+c^2},b,T) + \frac{1}{b\sqrt{\pi}}P^c(\sqrt{b^2+c^2},a,T) - \frac{\sqrt{a^2+b^2}}{ab\sqrt{\pi}}P^c(c,\sqrt{a^2+b^2},T)\right]$$

$$+2\left[\frac{1}{a\sqrt{\pi}}I^c_{13}(\sqrt{a^2+c^2},b,T) + \frac{1}{b\sqrt{\pi}}I^c_{13}(\sqrt{b^2+c^2},a,T) - \frac{\sqrt{a^2+b^2}}{ab\sqrt{\pi}}I^c_{13}(c,\sqrt{a^2+b^2},T)\right]$$

where from Chapter 3, Folder 9

(2.14.4) $I^c_2(a,b,w) = \text{-}w\,\mathrm{erfc}(aw)\mathrm{erfc}(bw) + \dfrac{e^{-a^2w^2}}{a\sqrt{\pi}}\mathrm{erfc}(bw) + \dfrac{e^{-b^2w^2}}{b\sqrt{\pi}}\mathrm{erfc}(aw) - \dfrac{\sqrt{a^2+b^2}}{ab\sqrt{\pi}}\mathrm{erfc}(w\sqrt{a^2+b^2})$

$$= \frac{T}{(aT)(bT)}\left[\frac{\mathrm{ierfc}(T\sqrt{a^2+b^2})}{\sqrt{\pi}} - \mathrm{ierfc}(aT)\mathrm{ierfc}(bT)\right]$$

and the other complementary functions are treated in the Folders cited above for $I_{14}(a,b,c,T)$.
Using the relation erfc(x)=1-erf(x) we also have

(2.14.5) $$I^c_{14}(a,b,c,T) = \frac{\sqrt{\pi}}{T}ierfc(cT) - I_1(c,a,T) - I_1(c,b,T) + I_{14}(a,b,c,T)$$

(2.14.6) $$I_{14}(a,b,c,T)=\frac{\sqrt{\pi}}{T}ierfc(cT)-I_1^c(c,a,T)-I_1^c(c,b,T)+I_{14}^c(a,b,c,T)$$

where $I_1(c,a,T)$, $I_1(c,b,T)$, $I_1^c(c,a,T)$ *and* $I_1^c(c,b,T)$ are computed in Chapter 3, Folder10 and displayed in Section **(2.6).**

Computer Subroutines

$\mathbf{I_{14}(a,b,c,T)}$ **or** $\mathbf{I_{14}^c(a,b,c,T)}$**: SUBROUTINE INTEGI14(…)** with KODE=1 or KODE=2

References Chapter 3, Folders14 and 10

(2.15) Functions $\mathbf{U(a,b,t), V(a,b,t), I_{21}(a,b,c,t), I^c_{21}(a,b,c,t), J_{21}(a,b,c,t), G_n(a,b,T)}$

$$\mathbf{U(a,b,t) = e^{a^2t+2ab}erfc(a\sqrt{t} + b/\sqrt{t})}$$

$$\mathbf{V(a,b,t) = \int_0^t U(a,b,\tau)d\tau, \quad I_{21}(a,b,c,t) = \int_0^t U(a,b,\tau)erf(c/\sqrt{\tau})d\tau}$$

$$\mathbf{I^c_{21}(a,b,c,t) = \int_0^t U(a,b,\tau)erfc(c/\sqrt{\tau})d\tau \qquad J_{21}(a,b,c,t) = \int_0^t \frac{U(a,b,\tau)e^{-c^2/\tau}}{\tau^{3/2}}d\tau}$$

$$a > 0, \quad b > 0, \quad c > 0, \quad t > 0$$

$$\mathbf{G_n(a,b,T) = \int_T^\infty \frac{e^{-a^2w^2}i^n erfc(bw)}{w^n}dw}, \quad a > 0, \quad b \geq 0, \quad T > 0, \quad n \geq 0$$

Representations

(2.15.1) $$V(a,b,t) = \frac{2}{a}\sqrt{\frac{t}{\pi}}e^{-b^2/t} - \left(\frac{1}{a^2}+\frac{2b}{a}\right)\mathrm{erfc}\left(\frac{b}{\sqrt{t}}\right) + \frac{U(t)}{a^2}$$

$$V(a,b,t) = \frac{2\sqrt{t}}{a}ierfc\left(\frac{b}{\sqrt{t}}\right) + \frac{1}{a^2}\left[U(a,b,t) - erfc\left(\frac{b}{\sqrt{t}}\right)\right]$$

(2.15.2) $$I_{21}(a,b,c,t) = V(t)\mathrm{erf}(c/\sqrt{t}) + \frac{c}{\sqrt{\pi}}\left[\frac{2}{a\sqrt{\pi}}E_1(X) - 2\left(\frac{1}{a^2}+\frac{2b}{a}\right)I_5(c,b,T) + \frac{2\sqrt{t}}{a^2\sqrt{\pi}}S_1(a,b,c,t)\right]$$

(2.15.3) $$I^c_{21}(a,b,c,t) = V(t)\mathrm{erfc}(c/\sqrt{t}) - \frac{c}{\sqrt{\pi}}\left[\frac{2}{a\sqrt{\pi}}E_1(X) - 2\left(\frac{1}{a^2}+\frac{2b}{a}\right)I_5(c,b,T) + \frac{2\sqrt{t}}{a^2\sqrt{\pi}}S_1(a,b,c,t)\right]$$

where $X = (b^2+c^2)/t$, I_5 is the I function of Folder 5 and $S_1(a,b,c,t)$ in computational form suitable for a quadrature is

(2.15.4) $$S_1(a,b,c,t) = e^{-X}\int_0^\infty \frac{e^{-2Bw-w^2}}{c^2+(b+w\sqrt{t})^2}dw, \qquad B = a\sqrt{t} + b/\sqrt{t}\,.$$

For J_{21}, we get

(2.15.5) $$J_{21}(a,b,c,t) = \frac{2\sqrt{t}}{\sqrt{\pi}}S_1(a,b,c,t)\,.$$

A series for large parameter L ($L \geq 2$) is

$$X = (b^2+c^2)/t, \qquad B = a\sqrt{t} + b/\sqrt{t}\,,$$

(2.15.6) $$S_1(a,b,c,t) = \frac{e^{-X}}{2tB}\sum_{k=0}^{\infty}\frac{U_k(x)}{L^k}[e^{B^2}E_{(k+3)/2}(B^2)],$$

$$L = \frac{B}{\sqrt{a^2t+c^2/t}}, \quad x = \frac{a\sqrt{t}}{\sqrt{a^2t+c^2/t}},$$

and $U_k(x)$, $k \geq 0$, are Chebyshev polynomials of the second kind which can be generated by forward recurrence on their three-term recurrence relation.

The expressions for $V(t)$, I_{21} and I_{21}^c above contain reciprocal powers of a, but the integrals are analytic functions of a. Therefore, to avoid losses of significance by small differences of large numbers when a is small, we develop the power series in the parameter a. The results are:

(2.15.7) $$U(t) = e^{a^2 t + 2ab}\,\text{erfc}(a\sqrt{t} + b/\sqrt{t}) = \sum_{n=0}^{\infty} i^n\,\text{erfc}\left(\frac{b}{\sqrt{t}}\right)(-2a\sqrt{t})^n$$

(2.15.8) $$V(t) = 4t\sum_{n=2}^{\infty} i^n\,\text{erfc}\left(\frac{b}{\sqrt{t}}\right)(-2a\sqrt{t})^{n-2}$$

(2.15.9) $$I_{21}(a,b,c,t) = V(t)\,\text{erf}\left(\frac{c}{\sqrt{t}}\right) + \frac{4c\sqrt{t}}{\sqrt{\pi}}e^{-X}\sum_{n=2}^{\infty}(-2a\sqrt{t})^{n-2}y_n\,, \qquad 0 \leq a\sqrt{t} \leq 1$$

(2.15.10) $$I_{21}^c(a,b,c,t) = V(t)\,\text{erfc}\left(\frac{c}{\sqrt{t}}\right) - \frac{4c\sqrt{t}}{\sqrt{\pi}}e^{-X}\sum_{n=2}^{\infty}(-2a\sqrt{t})^{n-2}y_n$$

(2.15.11) $$J_{21}(a,b,c,t) = \frac{e^{-X}}{\sqrt{t}}\sum_{n=0}^{\infty}(-2a\sqrt{t})^n y_n \qquad X = (b^2+c^2)/t,$$

where

(2.15.12) $$y_n = 2T^{n-1}e^X G_n(c,b,T), \qquad , \; G_n(c,b,T) = \int_T^{\infty} e^{-c^2w^2}\frac{i^n\,\text{erfc}(bw)}{w^n}dw \qquad T = \frac{1}{\sqrt{t}}.$$

An extensive stability analysis on the 3-term recurrence for y_n shows that the recurrence must be started with two (quadrature) values near the index [2X].

Recurrence for $\mathbf{y_n = 2T^{n-1}e^X G_n(c,b,T)}$

y_n is scaled to avoid extremes in values from exponentials:

(2.15.13) $$y_n = 2T^{n-1}e^X G_n(c,b,T), \qquad X = (b^2+c^2)/t = (b^2+c^2)T^2$$

(2.15.14) $$n(n+1)y_{n+1} + 2n(bT)y_n + Xy_{n-1} = e^{b^2T^2}i^{n-1}\,\text{erfc}(bT)$$

(2.15.15) $$y_0 = \frac{2e^X}{T}\int_T^{\infty} e^{-c^2w^2}\,\text{erfc}(bw) = \frac{2e^X}{T}I_5(c,b,T)\,, \quad X = (b^2+c^2)/t, \quad T = \frac{1}{\sqrt{t}},$$

(2.15.16) $$y_1 = -bTy_0 + \frac{1}{\sqrt{\pi}}[e^X E_1(X)],$$

Here $I_5(c,b,T)$ is the I function of Folder 5. A stability analysis indcates that the recurrence must be carried out by recurring away from the index N=[2X] to keep homogeneous solutions of the difference equation from amplifying rounding errors.

Special Cases

$c = 0$

(2.15.17) $I_{21}(a,b,0,t) = 0,$ $I_{21}^c(a,b,0,t) = V(t)$

$a = 0$ for $I_{21}(0,b,c,t)$ Using erf(x)=1-erfc(x), we have

(2.15.18) $I_{21}(0,b,c,t) = 2J_3^c(b,T) - 2W_3^c(b,c,T) = \dfrac{erfc(bT)}{T^2} - \dfrac{2b}{T} ierfc(bT) - 2W_3^c(b,c,T)$.

We chose this form because the computational results are much better than the results in terms of J_3 and W_3. Then, using the formula for W_3^c from Chapter 3, Folder 10

$$I_{21}(0,b,c,t) = \frac{\text{erf}(cT)\,\text{erfc}(bT)}{T^2} - \frac{2b}{T} ierfc(bT) + \frac{2}{\sqrt{\pi}}\left[bI_1^c(b,c,T) + I_1^c(c,b,T)\right]$$

Combining the iterated coerror functions using the formula from Folder 10,

$$I_1^c(b,c,T) = \frac{\sqrt{\pi}}{T} erfc(cT) ierfc(bT) - \frac{c}{\sqrt{\pi}} E_1(X) + 2bcI_5(c,b,T),$$

improves the computation somewhat by minimizing losses of significance

(2.15.19) $I_{21}(0,b,c,t) = \dfrac{4}{T^2} i^2 erfc(bT) erf(cT) + \dfrac{2c}{\sqrt{\pi}}\left[-\dfrac{b}{\sqrt{\pi}} E_1(X) + 2b^2 I_5(c,b,T) + I_1^c(c,b,T)\right]$

The series for $0 \le a\sqrt{t} \le 1$ with a=0 gives

(2.15.20) $I_{21}(0,b,c,t) = 4ti^2 \text{ erfc}\left(\dfrac{b}{\sqrt{t}}\right)\text{erf}\left(\dfrac{c}{\sqrt{t}}\right) + \dfrac{4c\sqrt{t}}{\sqrt{\pi}} e^{-X} y_2(c,b,T),$ $\mathrm{T} = 1/\sqrt{\mathrm{t}}$.

$a = 0$ for $I_{21}^c(0,b,c,t)$

For $I_{21}^c(0,b,c,t)$, we have

(2.15.21) $I_{21}^c(0,b,c,t) = \int_0^t \text{ erfc}\left(\dfrac{b}{\sqrt{\tau}}\right)\text{erfc}\left(\dfrac{c}{\sqrt{\tau}}\right) d\tau = 2W_3^c(b,c,T),$ $\mathrm{T} = 1/\sqrt{\mathrm{t}}$

where W_3^c is computed in Chapter 3, Folder 10b. From above, we also have

(2.15.22) $I_{21}^c(0,b,c,t) = 4ti^2 \text{ erfc}\left(\dfrac{b}{\sqrt{t}}\right)\text{erfc}\left(\dfrac{c}{\sqrt{t}}\right) - \dfrac{4c\sqrt{t}e^{-X}}{\sqrt{\pi}} y_2(c,b,T),$ $\mathrm{T} = 1/\sqrt{\mathrm{t}}$.

$a = 0$ for $J_{21}(0,b,c,t)$

(2.15.23) $J_{21}(0,b,c,t) = \int_0^t \text{ erfc}\left(\dfrac{b}{\sqrt{\tau}}\right)\dfrac{e^{-c^2/\tau}}{\tau^{3/2}} d\tau,$ $\tau = \dfrac{1}{w^2}$

$$= 2\int_T^\infty e^{-c^2w^2} \text{ erfc}(bw) dw = 2I_5(c,b,T), \qquad T = \frac{1}{\sqrt{t}}.$$

$\underline{a = 0,\ c = 0 \text{ for }} J_{21}(0,b,0,t)$

(2.15.24) $$J_{21}(0,b,0,t) = 2I_5(0,b,T) = 2\int_T^\infty \operatorname{erfc}(bw)dw = \frac{2}{b} i\operatorname{erfc}(bT)$$

$\underline{a = 0,\ b = 0 \text{ for }} I_{21}(0,0,c,t)$

(2.15.25) $$I_{21}(0,0,c,t) = \int_0^t \operatorname{erf}\left(\frac{c}{\sqrt{\tau}}\right)d\tau \qquad \text{(see case for } a = 0\text{)}$$
$$= \frac{\operatorname{erf}(cT)}{T^2} + \frac{2c}{T} i\operatorname{erfc}(cT)$$

$\underline{a = 0 \text{ for }} S_1(0,b,c,t)$ (see also case for $a = 0$ for $J_{21}(0,b,c,t)$)

(2.15.26) $$S_1(0,b,c,t) = \frac{\sqrt{\pi}}{\sqrt{t}} I_5(c,b,T), \qquad T = \frac{1}{\sqrt{t}}.$$

$\underline{a = 0,\ b = 0 \text{ for }} \mathrm{S}_1(0,0,c,t)$

From above,

(2.15.27) $$S_1(0,0,c,t) = \frac{\sqrt{\pi}}{\sqrt{t}} I_5(c,0,T) = \sqrt{\frac{\pi}{t}}\int_T^\infty e^{-c^2w^2}dw = \frac{\pi}{2c} T\operatorname{erfc}(cT), \qquad T = \frac{1}{\sqrt{t}}$$

$\underline{\text{b=0 for }} \mathrm{I}_{21}(a,0,c,t)$

(2.15.28) $$I_{21}(a,0,c,t) = V(a,0,t)\operatorname{erf}\left(\frac{c}{\sqrt{t}}\right) + \frac{c\sqrt{t}}{\sqrt{\pi}}\sum_{n=0}^\infty \frac{(-a\sqrt{t})^n}{\Gamma(\frac{n}{2}+2)} E_{\frac{n+3}{2}}(\frac{c^2}{t})$$

Notice also that

$$V(a,0,t) = 2\ \mathrm{I}_{23}(a\sqrt{t})/a^2 \quad , a \neq 0$$

(2.15.29) $$\mathrm{I}_{23}(x) = (e^{x^2} erfc(x) - 1)/2 + x/\sqrt{\pi} = x^2\sum_{n=0}^\infty \frac{(-x)^n}{\Gamma(\frac{n}{2}+1)(n+2)} = \frac{x^2}{2}\sum_{n=0}^\infty \frac{(-x)^n}{\Gamma(\frac{n}{2}+2)}$$

where I_{23} and related functions are found in Folder 23.

Computer Subroutines

V(a,b,t): DOUBLE PRECISION FUNCTION DVOFT(…)
$\mathbf{I_{21}(a,b,c,t)}$ or $\mathbf{I^c_{21}(a,b,c,t)}$: SUBROUTINE INTEGI21(…) with KODE=1 or KODE=2
$\mathbf{J^c_{21}(a,b,c,t)}$: SUBROUTINE INTEGJ21(…)
$\mathbf{S_1(a,b,c,t)}$: SUBROUTINE INTEGS1(…)
$\mathbf{y_n = 2T^{n-1}e^X G_n(c,b,T)}$: SUBROUTINE GNSEQ(…)

References: Chapter 3, Folder 21

(2.16) Functions $\mathbf{I_{22}(a,b,c,t)}$ and $\mathbf{J_{22}(a,b,c,t)}$

$$\mathbf{I_{22}(a,b,c,t) = \int_0^t U(a,b,\tau)\frac{e^{-c^2/\tau}}{\sqrt{\tau}}d\tau}$$

$$\mathbf{J_{22}(a,b,c,t) = \int_0^t U(a,b,\tau)\sqrt{\tau}\, e^{-c^2/\tau}\, d\tau}$$

$$\mathbf{U(a,b,t) = e^{a^2t+2ab}erfc(a\sqrt{t} + b/\sqrt{t})}$$

$$\boldsymbol{a \geq 0, \quad b \geq 0, \quad c \geq 0, \quad t > 0}$$

Representations

(2.16.1) $$I_{22}(a,b,c,t) = \frac{1}{a\sqrt{\pi}}E_1(X) - \frac{2\sqrt{t}}{a\sqrt{\pi}}S_2(a,b,c,t)$$

(2.16.2) $$J_{22}(a,b,c,t) = \frac{t}{a\sqrt{\pi}}E_2(X) - \frac{1}{a^2\sqrt{\pi}}\left(b+\frac{1}{2a}\right)E_1(X) + \frac{\sqrt{t}}{a^2}e^{-X}[e^{B^2}\text{ erfc}(B)]$$

$$-\frac{2c^2\sqrt{t}}{a^2\sqrt{\pi}}S_1(a,b,c,t) + \frac{\sqrt{t}}{a^3\sqrt{\pi}}S_2(a,b,c,t)$$

where $S_1(a,b,c,t)$ was defined in Chapter 3, Folder 21,

(2.16.3) $$S_1(a,b,c,t) = e^{-X}\int_0^\infty \frac{e^{-2Bw-w^2}}{c^2+(b+w\sqrt{t})^2}dw, \qquad B = a\sqrt{t} + b/\sqrt{t}$$

$$X = (b^2+c^2)/t$$

and

(2.16.4) $$S_2(a,b,c,t) = e^{-X}\int_0^\infty \frac{e^{-2Bw-w^2}(b+w\sqrt{t})}{c^2+(b+w\sqrt{t})^2}dw.$$

We also have for $L = B/\sqrt{a^2t+c^2/t} \geq 2$,

(2.16.5) $$S_1(a,b,c,t) = \frac{e^{-X}}{2Bt}\sum_{k=0}^\infty \frac{U_k(x)}{L^k}[e^{B^2}E_{(k+3)/2}(B^2)], \qquad B = a\sqrt{t} + b/\sqrt{t}$$

$$X = (b^2+c^2)/t$$

(2.16.6) $$S_2(a,b,c,t) = \frac{e^{-X}}{2\sqrt{t}}\sum_{k=0}^\infty \frac{T_k(x)}{L^k}[e^{B^2}E_{(k+2)/2}(B^2)], \qquad x = \frac{a\sqrt{t}}{\sqrt{a^2t+c^2/t}}$$

where $T_k(x)$ and $U_k(x)$ are Chebyshev polynomials of the first and second kinds. Both polynomials can be generated by forward recurrence on their three-term recurrence relations. While each series converges for $L > 1$, we apply the series for $L \geq 2$ to obtain rapid convergence in a numerical evaluation. If $L \geq 2$ is not satisfied, then $S_1(a,b,c,t)$ and $S_2(a,b,c,t)$ are computed by quadrature according to the procedure in the APPENDIX of Chapter 3, Folder 21.

The forms developed above have numerical problems for small a because of small differences of large numbers (an indeterminant form for $a \to 0$). Therefore we develop the power series for small a, $(a\sqrt{t} \le 1)$,

(2.16.7) $$I_{22}(a,b,c,t) = 2e^{-X}\left[b\sum_{n=0}^{\infty} y_n(-2a\sqrt{t})^n + \sqrt{t}\sum_{n=0}^{\infty} (n+2)y_{n+2}(-2a\sqrt{t})^n\right]$$

(2.16.8) $$J_{22}(a,b,c,t) = 4\sqrt{t}e^{-X}\left[b^2\sum_{n=0}^{\infty} y_{n+2}(-2a\sqrt{t})^n + b\sqrt{t}\sum_{n=0}^{\infty} (2n+5)y_{n+3}(-2a\sqrt{t})^n\right.$$
$$\left. + t\sum_{n=0}^{\infty} (n+2)(n+4)y_{n+5}(-2a\sqrt{t})^n\right]$$

where

(2.16.9) $$y_n = 2T^{n-1}e^X G_n(c,b,t) = 2T^{n-1}e^X\int_T^{\infty} e^{-c^2w^2}\frac{\mathrm{i}^n\mathrm{erfc}(bw)}{w^n}dw, \qquad T = 1/\sqrt{t}\,,$$

is computed in **(2.15)** or Chapter 3, Folder 21.

__Special Cases__

__$a = 0$ for__

(2.16.10) $$I_{22}(0,b,c,t) = \int_0^t \mathrm{erfc}\left(\frac{b}{\sqrt{\tau}}\right)\frac{e^{-c^2/\tau}}{\sqrt{\tau}}d\tau$$
$$= \frac{1}{T}E_{3/2}(c^2T^2) - 2I_1(c,b,T)$$

where $T = 1/\sqrt{t}$ and I_1 is computed in Folder 10a. We also have

(2.16.11) $$I_{22}(0,b,c,t) = 2\sqrt{t}e^{-c^2/t}\mathrm{erfc}(c/\sqrt{t}) - \frac{2b}{\sqrt{\pi}}E_1(X) - 4c^2I_5(c,b,T)$$

where I_5 is the I function of Folder 5. Notice also that the series for small a at $a = 0$ gives

(2.16.12) $$I_{22}(0,b,c,t) = 2e^{-X}[by_1(c,b,T) + 2\sqrt{t}y_2(c,b,T)]$$

__$a = 0$ for__

$$J_{22}(0,b,c,t) = \int_0^t \mathrm{erfc}\left(\frac{b}{\sqrt{\tau}}\right)\sqrt{\tau}e^{-c^2/\tau}d\tau\,,$$

(2.16.13) $$J_{22}(0,b,c,t) = 2\left[\frac{e^{-c^2T^2}\,erfc(bT)}{3T^3} - \frac{b}{3T^2\sqrt{\pi}}E_2(X) - \frac{2c^2}{3}I_1^c(c,b,T)\right]$$

where I_1^c is defined and computed in Chapter 3, Folder 10b(also = I_1^c of **(2.6)**),

(2.16.14) $$I_1^c(c,b,T) = \frac{\sqrt{\pi}}{T} i\,\text{erfc}(cT) - I_1(c,b,T)$$

in terms of I_1 of Folder 10a,

(2.16.15) $$I_1(c,b,T) = \frac{e^{-c^2T^2}\,\text{erf}(bT)}{T} + \frac{b}{\sqrt{\pi}} E_1(X) - c\sqrt{\pi}\,\text{erfc}(cT) + 2c^2 I_5(c,b,T)$$

where I_5 is the I function of Chapter 3, Folder 5. Also,

(2.16.16) $$J_{22}(0,b,c,t) = \frac{2}{3}\left[\sqrt{t}(t-2c^2)e^{-c^2/t}\,\text{erfc}(bT) - \frac{b}{T^2\sqrt{\pi}} E_2(X) + \frac{2bc^2}{\sqrt{\pi}} E_1(X) + 4c^4 I_5(c,b,T)\right] \qquad X = (b^2+c^2)/t, \qquad T = \frac{1}{\sqrt{t}}$$

We also have from the small a expression for $a = 0$,

(2.16.17) $$J_{22}(0,b,c,t) = 4\sqrt{t}e^{-X}[b^2 y_2(c,b,T) + 5b\sqrt{t}\,y_3(c,b,T) + 8ty_4(c,b,T)]$$

$a = 0,\ b = 0$ for

(2.16.18) $$I_{22}(0,0,c,t) = \int_0^t \frac{e^{-c^2/\tau}d\tau}{\sqrt{\tau}},$$

$$= 2\sqrt{\pi t}\ \text{ierfc}(cT), \qquad T = \frac{1}{\sqrt{t}}$$

$a = 0,\ b = 0$ for

(2.16.19) $$J_{22}(0,0,c,t) = \int_0^t \sqrt{\tau}e^{-c^2/\tau}d\tau,$$

$$= t^{3/2}E_{5/2}(c^2T^2), \qquad T = \frac{1}{\sqrt{t}}.$$

$a = 0,\ c = 0$ for

(2.16.20) $$I_{22}(0,b,0,t) = \int_0^t \frac{\text{erfc}(b/\sqrt{\tau})}{\sqrt{\tau}}d\tau$$

$$= \frac{2\,\text{erfc}(bT)}{T} - \frac{2b}{\sqrt{\pi}} E_1(b^2T^2), \qquad T = \frac{1}{\sqrt{t}}$$

b=0 for $I_{22}(a,0,c,t)$

(2.16.21) $$I_{22}(a,0,c,t) = \sqrt{t}\sum_{n=0}^{\infty} \frac{(-a\sqrt{t})^n}{\Gamma(\frac{n}{2}+1)} E_{\frac{n+3}{2}}\left(\frac{c^2}{t}\right)$$

Relationship Between $\mathbf{I_{22}(a,b,0,t)}$ and $\mathbf{J_{22}(a,b,0,t)}$

(2.16.22) $$I_{22}(a,b,0,t) = 2U(a,b,t)\sqrt{t} \; + \; \frac{2at}{\sqrt{\pi}} E_2\left(\frac{b^2}{t}\right) - \frac{2b}{\sqrt{\pi}} E_1\left(\frac{b^2}{t}\right) - 2a^2 J_{22}(a,b,0,t).$$

Asymptotic Expansion For $\mathbf{I_{22}(a,0,c,t)}$ For Large a

(2.16.23) $$I_{22}(a,0,c,t) \sim \frac{e^{-c^2/t}}{a\sqrt{\pi}} \sum_{k=0}^{N} \frac{(-1)^k (1/2)_k}{(ac)^{2k}} [e^{c^2/t}\Gamma(k,\frac{c^2}{t})] + G_N$$

where $$\left|G_N\right| \le \frac{(1/2)_{N+1}}{a\sqrt{\pi}(a^2c^2)^{N+1}} \Gamma(N+1,\frac{c^2}{t})$$

and for k=0 and k=1 we have

$$\Gamma(0,x) = \int_x^\infty \frac{e^{-w}}{w} dw = \int_1^\infty \frac{e^{-xv}}{v} dv = E_1(x) \; , \qquad \Gamma(1,x) = \int_x^\infty e^{-w} dw = e^{-x} \; .$$

Forward recurrence on

$$[e^x\Gamma(k+1,x)] = k[e^x\Gamma(k,x)] + x^k , \qquad k = 1,2... \qquad x = c^2/t$$

is numerically stable.

Computer Subroutines

$\mathbf{I_{22}(a,b,c,t)}$: SUBROUTINE INTEGI22(…)
$\mathbf{J_{22}(a,b,c,t)}$: SUBROUTINE INTEGJ22(…)
$\mathbf{S_2(a,b,c,t)}$: SUBROUTINE INTEGS2(…)

References: Chapter 3, Folder 10, Folder 22

(2.17) Function $\mathbf{I_{24}(a,b,c,t)}$, $\mathbf{I_{24}^c(a,b,c,t)}$ and Related Integrals

$$\mathbf{I_{24}(a,b,c,t)=\int_0^t \tau U(\tau)\,erf\left(\frac{c}{\sqrt{\tau}}\right)d\tau,\qquad I_{24}^c(a,b,c,t)=\int_0^t \tau U(\tau)\,erfc\left(\frac{c}{\sqrt{\tau}}\right)d\tau}$$

$$\mathbf{J_{24}(a,b,t)=\int_0^t \tau U(\tau)\,d\tau\qquad V_{24}(a,b,t)=\int_0^t V(\tau)d\tau}$$

where

$$\mathbf{U(t)=e^{a^2t+2ab}erfc(a\sqrt{t}+b/\sqrt{t})}$$

$$\mathbf{V(t)=\int_0^t U(\tau)\,d\tau=\frac{2}{a}\sqrt{\frac{t}{\pi}}e^{-b^2/t}-\left(\frac{1}{a^2}+\frac{2b}{a}\right)erfc\left(\frac{b}{\sqrt{t}}\right)+\frac{U(t)}{a^2}}$$

$$a>0,\quad b>0,\quad c>0,\quad t>0$$

Representations

For $I_{24}(a,b,c,t)$ we have

(2.17.1) $$I_{24}(a,b,c,t)=tV(t)\,\mathrm{erf}\left(\frac{c}{\sqrt{t}}\right)+\frac{c}{\sqrt{\pi}}T_0-[T_1-T_2-T_3]$$

where $V(t)$ is computed in Folder 21 and

$$T_0(a,b,c,t)=\frac{2t}{a\sqrt{\pi}}E_2(X)-2\left(\frac{1}{a}+\frac{2b}{a}\right)I_1^c(c,b,T)+\frac{1}{a^2}I_{22}(a,b,c,t)$$

$$T_1(a,b,c,t)=\frac{4}{3a\sqrt{\pi}}\left[\frac{e^{-b^2T^2}\,\mathrm{erf}(cT)}{T^3}+\frac{c}{T^2\sqrt{\pi}}E_2(X)-2b^2I_1(b,c,T)\right]$$

$$T_2(a,b,c,t)=2\left(\frac{1}{a^2}+\frac{2b}{a}\right)[J_3(c,T)-W_3(b,c,T)]=2\left(\frac{1}{a^2}+\frac{2b}{a}\right)[J_3^c(b,T)-W_3^c(b,c,T)]$$

$$T_3(a,b,c,t)=\frac{1}{a^2}I_{21}(a,b,c,t),\qquad T=\frac{1}{\sqrt{t}},\qquad X=(b^2+c^2)T^2$$

with I_1 , J_3, J_3^c, W_3, W_3^c *and* I_{22} computed in Folders 10 and 22 of Chapter 3.

For $I_{24}^c(a,b,c,t)$ we have

(2.17.2) $$I_{24}^c(a,b,c,t)=tV(t)\,\mathrm{erfc}\left(\frac{c}{\sqrt{t}}\right)-\frac{c}{\sqrt{\pi}}T_0^c-[T_1^c-T_2^c+T_3^c]$$

where

$$T_0^c=T_0$$

$$T_1^c=\frac{4}{3a\sqrt{\pi}}\left[\frac{e^{-b^2T^2}\,erfc(cT)}{T^3}-\frac{c}{T^2\sqrt{\pi}}E_2(X)-2b^2I_1^c(b,c,T)\right]$$

$$T_2^c = 2\left(\frac{1}{a^2}+\frac{2b}{a}\right)W_3^c(b,c,T)$$

$$T_3^c = \frac{1}{a^2} I_{21}^c(a,b,c,t), \qquad T = \frac{1}{\sqrt{t}}, \qquad X=(b^2+c^2)T^2$$

with I_1^c , J_3, J_3^c, W_3, W_3^c *and* I_{22}^c computed in Folders 10 and 22 of Chapter 3.

For $J_{24}(a,b,t)$ we have

(2.17.3) $$J_{24}(a,b,t) = \int_0^t \tau U(\tau)d\tau = \lim_{c\to\infty} I_{24}(a,b,c,t) = tV(t) - \int_0^t V(\tau)d\tau$$

and

(2.17.4) $$J_{24}(a,b,t) = \left(t-\frac{1}{a^2}\right)V(t) - \frac{2}{aT^3\sqrt{\pi}}E_{5/2}(b^2T^2) + \frac{4}{T^2}\left(\frac{1}{a^2}+\frac{2b}{a}\right)i^2 \operatorname{erfc}(bT)$$

$$J_{24}(a,b,t) = \left(t-\frac{1}{a^2}\right)V(t) - \frac{8}{aT^3}i^3 erfc(bT) + \frac{4}{a^2T^2}i^2 \operatorname{erfc}(bT)$$

It follows that

(2.17.5) $$V_{24}(a,b,t) = \int_0^t V(\tau)d\tau = \frac{1}{a^2}V(a,b,t) + \frac{2}{aT^3\sqrt{\pi}}E_{5/2}(b^2T^2) - \frac{4}{T^2}\left(\frac{1}{a^2}+\frac{2b}{a}\right)i^2 \operatorname{erfc}(bT)$$

$$= \frac{1}{a^2}V(a,b,t) + \frac{8}{aT^3}i^3 erfc(bT) - \frac{4}{a^2T^2}i^2 \operatorname{erfc}(bT)$$

In all of these, computation for small a results in losses of significance by small differences of large numbers or an indeterminant form for $a \to 0$ since each integral is analytic in the parameter a. Resolution of these forms for $a \to 0$ yields the power series in $a\sqrt{t}$ which converges best for $a\sqrt{t} < 1$:

(2.17.6) $$I_{24}(a,b,c,t) = J_{24}(a,b,t)\operatorname{erf}\left(\frac{c}{\sqrt{t}}\right) + \frac{8cte^{-X}}{\sqrt{\pi}}\left[b\sum_{n=0}^{\infty}(-2a\sqrt{t})^n y_{n+3} + \sqrt{t}\sum_{n=0}^{\infty}(n+2)(-2a\sqrt{t})^n y_{n+4}\right]$$

(2.17.7) $$I_{24}^c(a,b,c,t) = J_{24}(a,b,t)\operatorname{erfc}\left(\frac{c}{\sqrt{t}}\right) - \frac{8cte^{-X}}{\sqrt{\pi}}\left[b\sum_{n=0}^{\infty}(-2a\sqrt{t})^n y_{n+3} + \sqrt{t}\sum_{n=0}^{\infty}(n+2)(-2a\sqrt{t})^n y_{n+4}\right]$$

(2.17.8) $$J_{24}(a,b,t) = tV(t) - V_{24}(a,b,t)$$

(2.17.9) $$V_{24}(a,b,t) = 16t^2\sum_{n=0}^{\infty}(-2a\sqrt{t})^n i^{n+4} \operatorname{erfc}(bT)$$

where $V(t)$ and the $y_n(c,b,T)$ sequence are computed in Chapter 3, Folder 21,

(2.17.10) $$V(t) = 4t\sum_{n=0}^{\infty}(-2a\sqrt{t})^n i^{n+2} \operatorname{erfc}(bT) \qquad T = \frac{1}{\sqrt{t}}, \qquad X=(b^2+c^2)T^2$$

(2.17.11) $$y_n(c,b,T) = 2T^{n-1}e^X \int_T^\infty e^{-c^2w^2}\frac{i^n \operatorname{erfc}(bw)}{w^n}dw.$$

The iterated coerror functions are generated in subroutine DINERFC and the $y_n(c,b,T)$ sequence is genrerated in subroutine GNSEQ.

Computer Subroutines

$\mathbf{I_{24}(a,b,c,t)}$ or $\mathbf{I^c_{24}(a,b,c,t)}$: **SUBROUTINE INTEGI24(…)** with KODE=1 or KODE=2
$\mathbf{J_{24}(a,b,t)}$: **SUBROUTINE INTEGJ24(…)**
$\mathbf{V_{24}(a,b,t)}$: **SUBROUTINE INTEGV24(…)**

References: Chapter 3, Folders 10, 21, 22, 24

(2.18) Function $\mathbf{I_{25}(a,b,c,d,t)}$

$$\mathbf{I_{25}(a,b,c,d,t) = \int_0^t \; U(a,b,\tau)\,U(c,d,\tau)\,d\tau}$$

$$\mathbf{U(a,b,t) = e^{a^2t+2ab}\,erfc(a\sqrt{t} + b/\sqrt{t})}$$

$$a \ge 0, \quad b \ge 0, \quad c \ge 0, \quad d \ge 0, \quad t > 0$$

Representation

(2.18.1) $$I_{25}(a,b,c,d,t) = \frac{1}{(a^2+c^2)}\left\{a^2U(c,d,t)V(a,b,t) + \frac{c}{\sqrt{\pi}}R_1(a,b,d,t) - \frac{d}{\sqrt{\pi}}R_2(a,b,d,t)\right.$$

$$\left. -\frac{2ac^2}{\sqrt{\pi}}J_{22}(c,d,b,t) + c^2(1+2ab)I_{21}^c(c,d,b,t) + \frac{c}{\sqrt{\pi}}I_{22}(a,b,d,t) - \frac{d}{\sqrt{\pi}}J_{21}(a,b,d,t)\right\}$$

where

$$\begin{aligned} V(a,b,t) &= \frac{2}{a}\sqrt{\frac{t}{\pi}}e^{-b^2/t} - \frac{(1+2ab)}{a^2}\operatorname{erfc}\left(\frac{b}{\sqrt{t}}\right) + \frac{1}{a^2}U(a,b,t) \\ &= \frac{2\sqrt{t}}{a}ierfc\left(\frac{b}{\sqrt{t}}\right) + \frac{1}{a^2}\left[U(a,b,t) - erfc\left(\frac{b}{\sqrt{t}}\right)\right] \end{aligned}$$

$$R_1(a,b,d,t) = \frac{2at}{\sqrt{\pi}}E_2(X) - 2(1+2ab)I_1^c(d,b,T)$$

$$R_2(a,b,d,t) = \frac{2a}{\sqrt{\pi}}E_1(X) - 2(1+2ab)I_5(d,b,T)$$

$$T = \frac{1}{\sqrt{t}}, \qquad\qquad X = (b^2+d^2)/t.$$

Here, I_5 and I_1^c are computed in Folders 5 and 10. V, J_{21} I_{21}^c, I_{22} and J_{22} are the principal results of Folders 21 and 22. Notice also that I_{25} is symmetric in the pairs (a,b) and (c,d) and exchanging these pairs on the right yields an alternate form. In fact, any convex linear combination of these forms gives I_{25}; in particular, adding and dividing by 2 gives the symmetric form.

Computer Subroutine

$\mathbf{I_{25}(a,b,c,d,t)}$**: SUBROUTINE INTEGI25(…)**

Reference Chapter 3, Folder 25

(2.19) Function $\mathbf{I_{26}(a,b,c,d,t)}$

$$\boldsymbol{I_{26}(a,b,c,d,t) = \int_0^t \tau U(a,b,\tau)U(c,d,\tau)d\tau}$$

$$\boldsymbol{U(a,b,t) = e^{a^2t+2ab}\,\mathbf{erfc}(a\sqrt{t} + b/\sqrt{t})}$$

$$\boldsymbol{a \geq 0, \quad b \geq 0, \quad c \geq 0, \quad d \geq 0, \quad t > 0}$$

Series Representations

Symmetric form

(2.19.1)

$$\begin{aligned} I_{26}(a,b,c,d,t) = \frac{1}{(a^2+c^2)} \{ & tU(a,b,t)U(c,d,t) - I_{25}(a,b,c,d,t) \\ & + \frac{a}{\sqrt{\pi}} J_{22}(c,d,b,t) - \frac{b}{\sqrt{\pi}} I_{22}(c,d,b,t) \\ & + \frac{\mathrm{c}}{\sqrt{\pi}} \mathrm{J}_{22}(\mathrm{a,b,d,t}) - \frac{\mathrm{d}}{\sqrt{\pi}} \mathrm{I}_{22}(\mathrm{a,b,d,t}) \} \end{aligned}$$

where I_{22}, J_{22} and I_{25} are defined in Chapter 3, Folders 22 and 25. A more efficient computational form is also presented in terms of more fundamental integrals which eliminates redundant computation.

An alternate, non-symmetric, more complicated form is also derived in Chapter 3, Folder 26.

Computer Subroutine

$\mathbf{I_{26}(a,b,c,d,t)}$: SUBROUTINE INTEGI26(…)

Reference: Chapter 3, Folder 26

(2.20) Indefinite Integrals

$$\mathbf{J = \int \; e^{(a^2-b^2)x} erfc\left(a\sqrt{x} + \frac{c}{\sqrt{x}}\right)dx}$$

$$\mathbf{I = \int \; xe^{(a^2-b^2)x} erfc\left(a\sqrt{x} + \frac{c}{\sqrt{x}}\right)dx}$$

<u>**Representations**</u>

(2.20.1) $$J \equiv \int \; e^{(a^2-b^2)x} \operatorname{erfc}\left(a\sqrt{x} + \frac{c}{\sqrt{x}}\right)dx$$

$$= \frac{e^{(a^2-b^2)x}}{a^2-b^2} \operatorname{erfc}\left(a\sqrt{x} + \frac{c}{\sqrt{x}}\right) + \frac{e^{-2ac}}{2}\left[- f(\sqrt{x},a,b,c) + f(\sqrt{x},-a,b,-c)\right]$$

where $f(\sqrt{x},a,b,c) = \dfrac{e^{2bc}}{(a^2-b^2)}\left(1+\dfrac{a}{b}\right)\operatorname{erfc}\left(b\sqrt{x} + \dfrac{c}{\sqrt{x}}\right)$

(2.20.2)
$$I = -\frac{e^{(a^2-b^2)x}}{a^2-b^2}\left[-x + \frac{1}{a^2-b^2}\right]\operatorname{erfc}[X(\sqrt{x},a,c)]$$
$$+\frac{e^{-2ac}}{2}[F(\sqrt{x},a,b,c) - F(\sqrt{x},-a,b,-c)]$$

$$F(\sqrt{x},a,b,c) = \frac{e^{2bc}}{a^2-b^2}\left\{\frac{-\sqrt{x}}{b\sqrt{\pi}} e^{-X^2(\sqrt{x},b,c)}\left(1+\frac{a}{b}\right)\right.$$
$$\left. + \left[\left(\frac{c}{b} + \frac{1}{a^2-b^2}\right)\left(1+\frac{a}{b}\right) - \frac{a}{2b^3}\right]\operatorname{erfc}[X(\sqrt{x},b,c]\right\}$$

$$X(\sqrt{x},a,c) = a\sqrt{x} + c/\sqrt{x}$$

<u>**Computer Subroutines**</u>

I: SUBROUTINE ERFINT(…)

<u>**References:**</u> Chapter 3, Folder 12

(2.21) Functions $\mathbf{H_{23}(x)}$, $\mathbf{I_{23}(x)}$ and $\mathbf{J_{23}(x)}$

$$\mathbf{H_{23}(x)=\int_0^x e^{w^2}erfc(w)\,dw=\frac{1}{2}\int_0^{x^2} e^{v}erfc(\sqrt{v})\frac{dv}{\sqrt{v}}}$$

$$\mathbf{I_{23}(x)=\int_0^x e^{w^2}erfc(w)\,w\,dw=\frac{1}{2}\int_0^{x^2} e^{v}erfc(\sqrt{v})\,dv}$$

$$\mathbf{J_{23}(x)=\int_0^x e^{w^2}erfc(w)\,w^2dw=\frac{1}{2}\int_0^{x^2} e^{v}erfc(\sqrt{v})\sqrt{v}\,dv}$$

$$\boldsymbol{x\geq 0}$$

Representations

H_{23} is evaluated from the relations

(2.21.1)
$$H_{23}(x)=\begin{cases} S(x,0) & 0\leq x\leq 1 \\ S_2+S(x,2) & 1<x\leq 3 \\ S_4+S(x,4) & 3<x\leq 4 \\ S_4+V(x,4) & 4<x<\infty \end{cases}$$

where $S(x,x_0)$ is the Taylor expansion about x_0,

(2.21.2)
$$S(x,x_0)=\sum_{n=0}^{\infty}(-2)^n[e^{x_0^2}i^n\,\mathrm{erfc}(x_0)]\frac{(x-x_0)^{n+1}}{n+1},$$

(2.21.3)
$$S_2=\int_0^2 e^{v^2}\,\mathrm{erfc}(v)dv,\qquad S_4=\int_0^4 e^{v^2}\,\mathrm{erfc}(v)dv,$$

(2.21.4) $S_2 = 0.9753620874841564$ $\qquad$ $S_4 = 1.344468257503159.$

and $V(x,4)$ is the integral

(2.21.5)
$$V(x,4)=\int_4^x y(v)dv \qquad y(v)=e^{v^2}\mathrm{erfc}(v)=\frac{1}{v\sqrt{\pi}}{\sum_{n=0}^{\infty}}' a_{2r}T_{2r}\left(\frac{4}{v}\right),\qquad v\geq 4,$$

where $y(v)$ is a Chebyshev sum with coefficients a_{2r} (see Chapter 3, Folder 23 for numerical values). The result for $V(x,4)$ is

(2.21.6)
$$V(x,4)=\frac{1}{\sqrt{\pi}}\left[{\sum_{r=0}^{\infty}}' a_{2r}A_{2r}(x)+\ln\left(\frac{x}{4}\right)\right]$$

where $A_{2r}(x)$ is computed from the recurrence

(2.21.7)
$$A_0(x)=0,\qquad A_2(x)=1-\left(\frac{4}{x}\right)^2$$

$$A_{2r+2}(x)+A_{2r}(x)=-\frac{1}{2r(r+1)}-\left[\frac{T_{2r+2}(4/x)}{2r+2}-\frac{T_{2r}(4/x)}{2r}\right],\qquad r\geq 1.$$

Truncation of $V(x,4)$ at 18 terms suffices for errors $O(10^{-16})$. Here, the prime on the sum means to halve the first term.

$I_{23}(x)$ can be evaluated explicitly,

(2.21.8) $$I_{23}(x) = \frac{e^{x^2}\operatorname{erfc}(x) - 1}{2} + \frac{x}{\sqrt{\pi}}$$

and $J_{23}(x)$ can be expressed in terms of $H_{23}(x)$,

(2.21.9) $$J_{23}(x) = \frac{1}{2}\left[xe^{x^2}\operatorname{erfc}(x) + \frac{x^2}{\sqrt{\pi}} - H_{23}(x)\right]$$

Reduction Formula

Let

(2.21.10) $$I_\alpha(x) = \int_0^x e^{w^2}\operatorname{erfc}(w) w^\alpha dw, \qquad \alpha = 2n \text{ or } 2n+1, \quad n = 1,2, \ldots.$$

Then

(2.21.11) $$I_\alpha(x) = \frac{1}{2}x^{\alpha-1}\operatorname{erfc}(x) + \frac{1}{\sqrt{\pi}}\int_0^x w^{\alpha-1}dw - \frac{(\alpha-1)}{2}\int_0^x e^{w^2}\operatorname{erfc}(w) w^{\alpha-2}dw$$

(2.21.12) $$I_\alpha(x) = \frac{1}{2}x^{\alpha-1}\operatorname{erfc}(x) + \frac{1}{\alpha\sqrt{\pi}}x^\alpha - \frac{(\alpha-1)}{2}I_{\alpha-2}(x).$$

Notice that for $\alpha = 2$ we get the formula for $J_{23}(x)(=I_2(x))$ since $I_0(x) = H_{23}(x)$. Since $\alpha = 2n$ or $2n+1$, a repeated application of this formula ends up with $I_0(x)$ or $I_1(x)$ which translate to $H_{23}(x)$ or $I_{23}(x)$ respectively.

Numerically, all of the formulas derived so far suffer from high cancellation of significant digits when x is small which results in a loss of relative error. Consequently the power series is valuable in numerical evaluation.

Power Series

(2.21.13) $$I_\alpha(x) = \int_0^x e^{w^2}\operatorname{erfc}(w) w^\alpha dw = x^{\alpha+1}\sum_{n=0}^{\infty}\frac{(-x)^n}{\Gamma\left(\frac{n}{2}+1\right)(n+\alpha+1)}, \qquad \alpha = 0, 1, 2, \ldots$$

where $I_0(x) = H_{23}(x), I_1(x) = I_{23}(x)$ and $I_2(x) = J_{23}(x)$

Asymptotic Expansion for $H_{23}(x)$ for $x \to \infty$

(2.21.14) $$H_{23}(x) = \int_0^x e^{w^2} erfc(w) dw = \frac{1}{2}\int_0^{x^2}\frac{e^\tau erfc(\sqrt{\tau})}{\sqrt{\tau}}d\tau \sim \frac{1}{2\sqrt{\pi}}\left[\gamma + 2\ln(2x) - \sum_{n=1}^{\infty}\frac{A_n}{x^{2n}}\right]$$

where γ is the Euler constant

$$\gamma = 0.5772156649015328606 \text{ and } A_n = \frac{(-1)^n (1/2)_n}{n}.$$

Computer Subroutines

$H_{23}(x)$: DOUBLE PRECISION FUNCTION DHERFC(…)

References: Chapter 3, Folder 23

(2.22) Incomplete Bessel Function

$$\boldsymbol{I(a,b,X) = \int_X^\infty e^{-at-b/t} dt},$$
$$\boldsymbol{a > 0,\quad b > 0,\quad X > 0}$$

Representations

Small b/X expansion

(2.22.1) $$I = \frac{e^{-aX}}{a} + X\sum_{k=1}^{\infty} \frac{(-1)^k}{k!}\left(\frac{b}{X}\right)^k E_k(aX)$$

Small aX expansion

(2.22.2) $$I = 2\sqrt{\frac{b}{a}}K_1(2\sqrt{ab}) - X\sum_{k=0}^{\infty} \frac{(-1)^k}{k!}(aX)^k E_{k+2}\left(\frac{b}{X}\right)$$

A representation in terms of analytic functions suitable for a quadrature,

(2.22.3) $$I(a,b,X) = \sqrt{\frac{b}{a}}\int_{\ln(X\sqrt{a/b})}^{\infty} e^{-2\sqrt{ab}\cosh v+v}dv$$

The incomplete Gamma functions are explored in Chaudhry and Zubar where Equation 2.163 expresses I in terms of

(2.22.4) $$\Gamma(\alpha, x : b) = \int_x^\infty t^{\alpha-1}\exp(-t - b/t)dt = \frac{1}{2\pi i}\sum_{n=0}^{\infty} \frac{(-b)^n}{n!}\int_{c-i\infty}^{c+i\infty} \frac{\Gamma(s)x^{\alpha-n-s}}{s+n-\alpha}ds,$$

which for α = 1 is $I(1, b, x)$. Evaluation of the residues in the complex integral results in the small b/X expansion. Notice that a change of variables t=v/a in $I(a,b,X)$ produces

(2.22.5) $$I(a,b,X) = \frac{1}{a}I(1,ab,aX) = \frac{1}{a}\Gamma(1,aX : ab)$$

Special Cases

X=0 gives the Bessel Function of order 1

(2.22.6) $$I(a,b,0) = \int_0^\infty e^{-at-b/t}dt = 2\sqrt{\frac{b}{a}}K_1(2\sqrt{ab})$$

a=0 gives

(2.22.7) $$I(0,b,X) = \int_X^\infty e^{-b/t}dt = X\ E_2(b/X).$$

b=0 gives

(2.22.8) $$I(a,0,X) = \int_X^\infty e^{-at}dt = \frac{e^{-aX}}{a}$$

References: Chapter 3, Folder17;
Chaudhry, M. A. and Zubair, S. M. "On A Class of Incomplete Gamma Functions with Applications" Chapman & Hall/CRC, Boca Raton, 2001

3. Derivations

This chapter is divided_into sub-sections called folders. These folders contain derivations of formulae for a variety of integrals. Numerically promising formulae were implemented in FORTRAN subroutines and tested with appropriate program drivers which are described in Chapter 4. The TABLE OF CONTENTS below lists each folder, a brief description, and computer files and subroutines appropriate to that folder in the format (FILE:SUBROUTINES).

TABLE OF CONTENTS

BECKSUBS: INTEGI6
BECKDRVR: I6COMP

Folder 4: A Convolution Integral $I = \int_0^t K(t-\tau)J(\tau)d\tau$ where

$$K(t) = \frac{A}{\sqrt{t}} e^{-c_2^2/t}, \qquad J(t) = c_1\left[\frac{2}{\sqrt{\pi t}} - 2c_1 e^{c_1^2 t}\,\mathrm{erfc}(c_1\sqrt{t})\right]$$

$$A = \frac{1}{2\sqrt{\pi\alpha}}, \quad c_1 = \frac{h\sqrt{\alpha}}{k}, \quad c_2 = \frac{z}{2\sqrt{\alpha}}, \qquad h>0,\ \alpha>0,\ k>0,\ z\ge 0$$

Summary While this integral can be evaluated easily by the LaPlace Transform, the object of this analysis is to derive the result by straight forward manipulation of the convolution.

Folder 5: Series Expansions for

$$I(a,b,X) = \int_X^\infty e^{-a^2x^2}\,\mathrm{erfc}(bx)dx, \qquad J(a,b,X) = \int_X^\infty e^{-a^2x^2}\,\mathrm{erf}(bx)dx$$

$$U(a,b,X) = \int_0^X e^{-a^2x^2}\,\mathrm{erfc}(bx)dx, \qquad V(a,b,X) = \int_0^X e^{-a^2x^2}\,\mathrm{erf}(bx)dx$$

$$a>0, \quad b>0, \quad X\ge 0$$

Summary These functions seem to evolve extensively during the analysis of integrals containing error functions. Therefore, they, especially I, J, and V are analyzed thoroughly to produce computational algorithms. These functions are also designated by I_5, J_5, U_5, *and* V_5.

BECKSUBS: INTEGI5, INTEGJ5, INTEGV5
BECKDRVR: I5COMP, J5COMP, V5COMP

Folder 6: Convergent Series for

$$I_6(a,b,T) = \int_T^\infty \frac{\mathrm{erf}(aw)\,\mathrm{erf}(bw)}{w^2}dw = \frac{1}{2}\int_0^t \frac{1}{\sqrt{u}}\,\mathrm{erf}\left(\frac{a}{\sqrt{u}}\right)\mathrm{erf}\left(\frac{b}{\sqrt{u}}\right)du$$

$$a>0, \quad b>0, \quad T = 1/\sqrt{t}, \quad t>0$$

Summary Symmetric and non-symmetric series are developed in powers of $a^2/(a^2+b^2)$ and $b^2/(a^2+b^2)$ for $a \le b$ and $a > b$ respectively. Since these ratios are at most ½, the series can be evaluated to high accuracy with relatively few terms. The function

$$G(x) = \int_x^\infty \frac{\mathrm{erfc}(w)}{w} dw, \qquad x > 0,$$

is explored as part of the analysis. Folder 16 presents an algorithm for a more accurate evaluation. Folder 15 presents the results for I_6 in terms of more basic functions P and Q of Folder 11.

BECKSUBS: INTEGI6
BECKDRVR: I6COMP

Folder 7: Evaluation of

$$I_3(a,b,c,T) = \int_T^\infty e^{-c^2w^2}\,\mathrm{erf}(aw)\,\mathrm{erf}(bw)dw = \frac{1}{2}\int_0^t \frac{e^{-c^2/u}}{u^{3/2}}\mathrm{erf}\left(\frac{a}{\sqrt{u}}\right)\mathrm{erf}\left(\frac{b}{\sqrt{u}}\right)du$$

$$I_3^c(a,b,c,T) = \int_T^\infty e^{-c^2w^2}\,\mathrm{erfc(aw)erfc(bw)dw} = \frac{1}{2}\int_0^t \frac{e^{-c^2/u}}{u^{3/2}}\mathrm{erfc}\left(\frac{a}{\sqrt{u}}\right)\mathrm{erfc}\left(\frac{b}{\sqrt{u}}\right)du$$

$$a \ge 0, \quad b \ge 0, \quad c > 0, \quad T = 1/\sqrt{t}, \quad t > 0$$

<u>Summary</u> Subfolders 7a-7f present several representations for I_3, each having a place in the computational tree which defines the algorithm. The results of Folder 5 lead to the main (default) computational series. The evaluation of $I_3(a,b,c,T)$ is converted into the evaluation of $I_3^c(a,b,c,T)$ and several closed forms are derived for $I_3^c(a,0,c,T)$, $I_3^c(0,b,0,T)$, $I_3^c(a,b,0,0)$. $I_3^c(a,b,c,T)$ is also designated in some contexts as $J_3^c(a,b,c,T)$

BECKSUBS: INTEGI3
BECKDRVR: I3COMP

Folder 8: Evaluation of Integrals

$$I_4(a,b,c,T) = \int_T^\infty w^2 e^{-c^2w^2}\,\mathrm{erf}(aw)\,\mathrm{erf}(bw)dw = \frac{1}{2}\int_0^t \frac{e^{-c^2/u}}{u^{5/2}}\mathrm{erf}\left(\frac{a}{\sqrt{u}}\right)\mathrm{erf}\left(\frac{b}{\sqrt{u}}\right)du$$

$$J_4(a,b,c,T) = \int_T^\infty w e^{-c^2w^2}\,\mathrm{erf}(aw)\,\mathrm{erf}(bw)dw = \frac{1}{2}\int_0^t \frac{e^{-c^2/u}}{u^{2}}\mathrm{erf}\left(\frac{a}{\sqrt{u}}\right)\mathrm{erf}\left(\frac{b}{\sqrt{u}}\right)du$$

$$a \ge 0, \quad b \ge 0, \quad c > 0, \quad T = 1/\sqrt{t}, \quad t > 0$$

<u>Summary</u> Subfolders 8a-8d present computational forms. Folder 8a reduces I_4 to the I_3 integral of Folder 7. Folder 8b develops the quadrature method described in Folder 3 for the computation of I_4. Folder 8c reduces the computation of J_4 to the J function of Folder 5. Folder 8d develops the quadrature method of Folder 3 for J_4.

RESEARCH: I4SER, J4SER

Folder 9: Integrals Related to Folder 7

$$I_2(a,b,T)=\int_0^T \operatorname{erf}(aw)\operatorname{erf}(bw)dw=\frac{1}{2}\int_t^\infty \frac{\operatorname{erf}\left(a/\sqrt{u}\right)\operatorname{erf}\left((b/\sqrt{u}\right)}{u^{3/2}}du$$

$$I_2^c(a,b,T)=\int_T^\infty \operatorname{erfc}(aw)\operatorname{erfc}(bw)dw=\frac{1}{2}\int_0^t \frac{\operatorname{erfc}\left(a/\sqrt{u}\right)\operatorname{erfc}\left((b/\sqrt{u}\right)}{u^{3/2}}du$$

$$I_9(a,b,T)=\int_0^T w\operatorname{erf}(aw)\operatorname{erf}(bw)dw=\frac{1}{2}\int_t^\infty \frac{\operatorname{erf}\left(a/\sqrt{u}\right)\operatorname{erf}\left((b/\sqrt{u}\right)}{u^2}du$$

$$I_9^c(a,b,T)=\int_T^\infty w\operatorname{erfc}(aw)\operatorname{erfc}(bw)dw=\frac{1}{2}\int_0^t \frac{\operatorname{erfc}\left(a/\sqrt{u}\right)\operatorname{erfc}\left((b/\sqrt{u}\right)}{u^2}du$$

$$a>0,\quad b>0,\quad T=1/\sqrt{t},\quad t>0$$

Summary Closed form expressions are derived using the results of Folder 7.The power series for each integral is displayed and indeterminancies for parameters to zero are resolved for numerical purposes.

BECKSUBS: INTEGI2, INTEGI9
BECKDRVR: I2COMP, I9COMP

Folder 10: Closed Form for I_1 of Folders 1 and 2 and Related Integrals

$$I_1(a,b,T)=\int_T^\infty \frac{e^{-a^2w^2}\operatorname{erf}(bw)}{w^2}dw=\frac{1}{2}\int_0^t \frac{e^{-a^2/\tau}\operatorname{erf}(b/\sqrt{\tau})}{\sqrt{\tau}}d\tau,$$

$$W_3(a,b,T)=\int_T^\infty \frac{\operatorname{erf}(aw)\operatorname{erf}(bw)}{w^3}dw=\frac{1}{2}\int_0^t \operatorname{erf}\left(a/\sqrt{\tau}\right)\operatorname{erf}\left(b/\sqrt{\tau}\right)d\tau$$

$$I_1^c(a,b,T)=\int_T^\infty \frac{e^{-a^2w^2}\operatorname{erfc}(aw)}{w^2}dw,\qquad W_3^c(a,b,T)=\int_T^\infty \frac{\operatorname{erfc}(aw)\operatorname{erfc}(bw)}{w^3}dw$$

$$J_\nu(a,T)=\int_T^\infty \frac{\operatorname{erf}(aw)}{w^\nu}dw\ ,\ \nu>1\qquad J_\nu^c(a,T)=\int_T^\infty \frac{\operatorname{erfc}(aw)}{w^\nu}dw\ ,\ \nu\neq 1$$

$$a>0,\quad b>0,\quad T=1/\sqrt{t},\quad t>0$$

where the superscript c denotes the replacement of erf with erfc. (See Folders 3, 6, 11 for similar pairs)

Summary I_1 and I_1^c are expressed in closed form in terms of the J and I functions of Folder 5. W_3 is expressed in terms of $I_1(a,b,T)$ and $I_1(b,a,T)$. Similarly, W_3^c is

expressed in terms of $I_1^c(a,b,T)$ and $I_1^c(b,a,T)$. J_ν and J_ν^c have closed form expressions.

BECKSUBS: INTEGI1, INTEGW3
BECKDRVR: I1COMP, W3COMP

Folder 11: Evaluation of

$$P(a,b,T) = \int_T^\infty \frac{e^{-a^2w^2}\,\mathrm{erf}(bw)}{w}\,dw, \qquad P^c(\mathrm{a,b,T}) = \int_\mathrm{T}^\infty \frac{\mathrm{e}^{-a^2w^2}\,\mathrm{erfc(bw)}}{\mathrm{w}}\,\mathrm{dw}$$

$$Q(a,b,T) = \int_T^\infty e^{-a^2w^2} E_1(b^2w^2)\,dw$$

$$a > 0, \quad b > 0, \quad T \geq 0$$

<u>Summary</u> Expansions for these integrals are derived from the results of Folder 6. These integrals occur in other contexts and seem to be of basic importance, like the I and J integrals of Folder 5.

BECKSUBS: INTEGP, INTEGQ
BECKDRVR: PCOMP, QCOMP, PQUAD, QQUAD

Folder 12: Evaluation of

$$I(a,b,c,x) = \int xe^{(a^2-b^2)x}\,\mathrm{erfc}(a\sqrt{x} + c/\sqrt{x})\,dx$$

<u>Summary</u> The known formula [Beck et al, p. 426 #3]

$$J = \int e^{(a^2-b^2)x}\,\mathrm{erfc}(a\sqrt{x} + c/\sqrt{x})\,dx$$

is manipulated to get an x multiplier in the integrand.

RESEARCH: ERFINT

Folder 13: Evaluation of

$$I_{13}(a,b,T) = \int_T^\infty \frac{e^{-a^2w^2}\,\mathrm{erf}(bw)}{w^3}\,dw = \frac{1}{2}\int_0^t e^{-a^2/u}\,\mathrm{erf}\left(\frac{b}{\sqrt{u}}\right)du$$

$$I_{13}^c(a,b,T) = \int_T^\infty \frac{e^{-a^2w^2}\,erfc(bw)}{w^3}\,dw = \frac{1}{2}\int_0^t e^{-a^2/u}\,erfc\left(\frac{b}{\sqrt{u}}\right)du$$

$$a > 0, \quad b \geq 0, \quad T = 1/\sqrt{t}, \quad t > 0$$

<u>Summary</u> This integral is a generalization of those found in Folders 10 and 11. I_{13} is expressed in terms of P of Folder 11.

RESEARCH: I13COMP, INTEGI13

Folder 14: Evaluation of

$$I_{14}(a,b,c,T) = \int_T^\infty e^{-c^2w^2} \frac{\mathrm{erf}(aw)\,\mathrm{erf}(bw)}{w^2} dw = \frac{1}{2}\int_0^t e^{-c^2/u} \frac{\mathrm{erf}(a/\sqrt{u})\,\mathrm{erf}(b/\sqrt{u})}{\sqrt{u}} du$$

$$I_{14}^c(a,b,c,T) = \int_T^\infty e^{-c^2w^2} \frac{\mathrm{erfc(aw)erfc(bw)}}{w^2} dw = \frac{1}{2}\int_0^t e^{-c^2/u} \frac{\mathrm{erfc}(a/\sqrt{u})\mathrm{erfc}(b/\sqrt{u})}{\sqrt{u}} du$$

$$a > 0, \quad b \ge 0, \quad c \ge 0, \quad T = 1/\sqrt{t}\,, \quad t > 0$$

Summary I_{14} can be expressed in terms of the I_3 function of Folder 7, the P function of Folder 11, and the I_{13} function of Folder 13. I_{14}^c ia expressed similarly in terms of the same complementary functions The special case for $c = 0$ is treated in Folders 3, 6 and 15.

RESEARCH: INTEGI14, I14COMP

Folder 15: Folders 3 and 6 Revisited

$$I_6(a,b,T) = \int_T^\infty \frac{\mathrm{erf(aw)erf(bw)}}{w^2} dw = \frac{1}{2}\int_0^t \frac{\mathrm{erf}(a/\sqrt{u})\mathrm{erf}(b/\sqrt{u})}{\sqrt{u}} du$$

$$I_6^c(a,b,T) = \int_T^\infty \frac{\mathrm{erfc(aw)erfc(bw)}}{w^2} dw = \frac{1}{2}\int_0^t \frac{\mathrm{erfc}(a/\sqrt{u})\mathrm{erfc}(b/\sqrt{u})}{\sqrt{u}} du$$

$$a > 0, \quad b \ge 0, \quad T = 1/\sqrt{t}\,, \quad t > 0$$

Summary I_6 and I_6^c can be expressed exclusively in terms of the P and Q functions of Folder 11, which makes the computation more orderly. See Folder 3 also.

BECKSUBS: INTEGI6
BECKDRVR: I6COMP

Folder 16: Accurate Evaluation of

$$F(x) = \int_0^x \frac{\mathrm{erf}(w)}{w} dw, \qquad G(x) = \int_x^\infty \frac{\mathrm{erfc}(w)}{w} dw,$$

$$\int_0^x e^{-a^2w^2} \ln w dw, \qquad \int_x^\infty e^{-a^2w^2} \ln w dw,$$

$$a > 0, \quad x > 0$$

Summary The relation $F(x) = \gamma/2 + \ln(2x) + G(x)$ (γ = Euler constant) is derived and computational forms are derived for all of these integrals. In particular, algorithms for $F(x)$ and $G(x)$ are presented and the last two logarithmic integrals are expressed in terms of F and G. $G(x)$ is defined in Folder 6 and is used to compute $G_{1/2}(x)$ in Folder 18.

AMOSSUBS: DFERF, DGERFC
AMOSDRVR: DFCOMP, DGCOMP
RESEARCH: DGSCOMP

Folder 17: Notes on the Evaluation of

$$I(a,b,x)=\int_x^\infty e^{-at-b/t}dt, \qquad a>0,\quad b>0,\quad x>0$$

Summary Expansions for small and large x are derived. An expansion for $\Gamma(\alpha,x{:}b)$ (equation 2.163) of Chaudhry and Zubair is derived for integer $\alpha = 1$ which is $I(1,b,x)$ with a change of variables, $at = u$.

$I(a,b,x) = I(1,ab,ax)/a = \Gamma(1,ax{:}ab)/a$

Folder 18: Evaluation of

$$E_\nu(x)=\int_1^\infty \frac{e^{-xt}}{t^\nu}dt, \quad G_\nu(x)=\int_1^\infty \frac{E_\nu(xt)}{t^\nu}dt=\frac{-\partial E_\nu(x)}{\partial\nu}=\int_1^\infty \frac{e^{-xt}\ln t}{t^\nu}dt$$

for Integer and Half Odd Integer Orders With Application to

$$I_n(b,T)=\int_T^\infty \frac{e^{-b^2x^2}\ln x}{x^n}dx, \qquad b>0,\quad x>0,\quad n\ge 0$$

Summary The analysis and documentation is provided for computer codes to evaluate K member sequences $E_{n+\nu}(x)$, $G_{n+\nu}(x)$ where $\nu = 1$ or ½, $n \ge 0$. Stability analysis of recurrences is a key consideration in the analysis. Exponential scaling is a code option.

AMOSSUBS: DEXINT, DHEXINT, DGEXINT, DGHEXINT
AMOSDRVR: GECOMP, GHECOMP

Folder 19: Evaluation of

$$I_{19}(a,b,T)=\int_0^T \frac{erf(aw)erf(bw)}{w}dw=\frac{1}{2}\int_t^\infty \frac{erf(a/\sqrt{\tau})erf(b/\sqrt{\tau})}{\tau}d\tau$$

$$I_{19}^c(a,b,T)=\int_T^\infty \frac{erfc(aw)erfc(bw)}{w}dw=\frac{1}{2}\int_0^t \frac{erfc(a/\sqrt{\tau})erfc(b/\sqrt{\tau})}{\tau}d\tau$$

$$a>0,\quad b>0,\quad T=1/\sqrt{t}\,,\quad t>0,$$

Summary I_{19}^c is expanded into series in powers $a^2/(a^2+b^2)$ and $b^2/(a^2+b^2)$ for $a \le b$ and $a > b$ respectively with $E_{k+3/2}(x)$ and $G_{k+3/2}(x)$ functions of Folder 18 as term multipliers in the series. The $G(x)$ function of Folders 6 and 16 and the I function of Folder 5 appear in leading terms. As with previous results, each ratio is at most ½ and rapidly convergent series result. I_{19} is expressed in terms of I_{19}^c by repeated use of erf(x)=1-erfc(x) and identification of the resulting integrals in terms of the G function of Folder16

RESEARCH: I19COMP, INTEGI19

Folder 20: Evaluation of

$$I_{20}(a,b,T)=\int_T^{\infty} e^{-a^2x^2}\operatorname{erf}(bx)\ln x\,dx,$$

$$I_{20}^{c}(a,b,T)=\int_T^{\infty} e^{-a^2x^2}\operatorname{erfc}(bx)\ln x\,dx,$$

$$a>0,\quad b>0,\quad T>0$$

<u>Summary</u> Integration by parts leads to results in power series for I_{20}^{c} in $a^2/(a^2+b^2)$ and $b^2/(a^2+b^2)$ for $a \le b$ and $a > b$, respectively, along with terms involving the I function of Folder 5. As with previous results each ratio is at most ½ and rapidly convergent series result. The relation erf(x)=1-erfc(x) expresses I_{20} in terms of I_{20}^{c}. A reflexive relation for $I_{20}^{c}(a,b,T)$ in terms of $I_{20}^{c}(b,a,T)$ is derived from $I_{19}^{c}(a,b,T)$.

RESEARCH: I20COMP, INTEGI20

Folder 21: Evaluation of

$$U(a,b,t)=e^{a^2t+2ab}\operatorname{erfc}(a\sqrt{t}+b/\sqrt{t})$$

$$V(a,b,t)=\int_0^{t} U(a,b,\tau)d\tau$$

$$I_{21}(a,b,c,t)=\int_0^{t} U(a,b,\tau)\operatorname{erf}(c/\sqrt{\tau})d\tau$$

$$I_{21}^{c}(a,b,c,t)=\int_0^{t} U(a,b,\tau)\operatorname{erfc}(c/\sqrt{\tau})d\tau$$

$$J_{21}(a,b,c,t)=\int_0^{t} \frac{U(a,b,\tau)e^{-c^2/\tau}}{\tau^{3/2}}d\tau$$

$$a>0,\quad b>0,\quad c>0,\quad t>0$$

$$G_n(a,b,T)=\int_T^{\infty} e^{-a^2w^2}\frac{\mathrm{i}^n\operatorname{erfc}(bw)}{w^n}dw, \qquad a>0,\quad b>0,\quad T>0,\quad n\ge 0$$

<u>Summary</u> I_{21} I_{21}^{c} and J_{21} are reduced to integrals with analytic integrands which are suitable for polynomial type integrators like DGAUS8 or DQUAD8. A scaled recurrence for $G_n(a,b,T)$ is derived and a stability analysis leads to a code for accurate evaluation of sequences which are used in the series for small parameter a where indeterminate forms in reciprocal powers of a are resolved.

BECKSUBS: INTEGI21, INTEGJ21, GNSEQ, DVOFT, INTEGS1
BECKDRVR: I21COMP, J21COMP, GNCOMP

Folder 22: Evaluation of

$$I_{22}(a,b,c,t) = \int_0^t U(a,b,\tau)\frac{e^{-c^2/\tau}}{\sqrt{\tau}}d\tau\,,$$

$$J_{22}(a,b,c,t) = \int_0^t U(a,b,\tau)\sqrt{\tau}e^{-c^2/\tau}d\tau$$

$$U(a,b,t) = e^{a^2t+2ab}\operatorname{erfc}(a\sqrt{t} + b/\sqrt{t})$$

a>0, b>0, c>0 t>0

Summary These integrals are closely related to those of Folder 21. The same methods are used to derive the (default) quadratures and series for special combinations of parameters. The $G_n(a,b,T)$ sequence of Folder 21 is used for small a power series where indeterminate forms in reciprocal powers of a are resolved.

BECKSUBS: INTEGI22, INTEGJ22, INTEGS2
BECKDRVR: I22COMP, J22COMP

Folder 23: Evaluation of

$$H_{23}(x) = \int_0^x e^{w^2}\operatorname{erfc}(w)dw = \frac{1}{2}\int_0^{x^2} e^v\operatorname{erfc}(\sqrt{v})\frac{dv}{\sqrt{v}}$$

$$I_{23}(x) = \int_0^x e^{w^2}\operatorname{erfc}(w)w\,dw = \frac{1}{2}\int_0^{x^2} e^v\operatorname{erfc}(\sqrt{v})dv$$

$$J_{23}(x) = \int_0^x e^{w^2}\operatorname{erfc}(w)w^2dw = \frac{1}{2}\int_0^{x^2} e^v\operatorname{erfc}(\sqrt{v})\sqrt{v}dv$$

$$x \geq 0$$

Summary These integrals arise as special cases in Folder 21, are found in tables of integrals, and terminate the sequence in the reduction formula

$$\int_0^x e^{w^2}\operatorname{erfc}(w)w^\alpha dw = \frac{1}{2}x^{\alpha-1}e^{x^2}\operatorname{erfc}(x) + \frac{x^\alpha}{\alpha\sqrt{\pi}} - \frac{\alpha-1}{2}\int_0^x e^{w^2}\operatorname{erfc}(w)w^{\alpha-2}dw$$

$$\alpha = 2n \text{ or } 2n{+}1,\quad n \geq 1$$

Formulae suitable for numerical evaluation are developed. Because these formulae lose significant digits when x is small, the power series is developed for $0 \leq x \leq 1$, n ≥ 1.

AMOSSUBS: DHERFC
AMOSDRVR: HERFCOMP

Folder 24: Evaluation of

$$I_{24}(a,b,c,t) = \int_0^t \tau U(a,b,\tau)\mathrm{erf}(c/\sqrt{\tau})d\tau$$

$$I_{24}^c(a,b,c,t) = \int_0^t \tau U(a,b,\tau)\mathrm{erfc}(c/\sqrt{\tau})d\tau$$

$$J_{24}(a,b,t) = \int_0^t \tau U(a,b,\tau)d\tau \qquad V_{24}(a,b,t) = \int_0^t V(a,b,\tau)d\tau$$

where

$$U(a,b,t) = e^{a^2t+2ab}\,\mathrm{erfc}(a\sqrt{t} + b/\sqrt{t})$$

$$V(a,b,t) = \int_0^t U(a,b,\tau)d\tau = \frac{2}{a}\sqrt{\frac{t}{\pi}}e^{-b^2/t} - \left(\frac{1+2ab}{a^2}\right)\mathrm{erfc}\left(\frac{b}{\sqrt{t}}\right) + \frac{U(a,b,t)}{a^2}$$

a>0, b>0, c>0, t>0

Summary The evaluation of I_{24} and I_{24}^c proceeds by integration by parts and identification of the resulting integrals in terms of known functions and those of previous folders. $J_{24}(a,b,t)$ is computed by taking $c \to \infty$ in each of the terms for $I_{24}(a,b,c,t)$. V_{24} is computed from the relation $V_{24}(a,b,t) = tV(a,b,t) - J_{24}(a,b,t)$.

Appropriate power series expansions for small a are developed in the manner of Folders 21 and 22.

RESEARCH: I24COMP, INTEGI24, J24COMP, INTEGJ24, V24COMP, INTEGV24

Folder 25: Evaluation of

$$I_{25}(a,b,c,d,t) = \int_0^t U(a,b,\tau)U(c,d,\tau)d\tau$$

$$U(a,b,t) = e^{a^2t+2ab}\,\mathrm{erfc}(a\sqrt{t} + b/\sqrt{t})$$

$$a \ge 0,\quad b \ge 0,\quad c \ge 0,\quad d \ge 0,\quad t > 0$$

Summary Integration by parts gives integrals which can be evaluated in terms of the results of Folders 5, 10, 21 and 22.

RESEARCH: I25COMP, INTEGI25

Folder 26: Evaluation of

$$I_{26}(a,b,c,d,t) = \int_0^t \tau U(a,b,\tau)U(c,d,\tau)d\tau$$

$$U(a,b,t) = e^{a^2t+2ab}\,\mathrm{erfc}(a\sqrt{t} + b/\sqrt{t})$$

$$a \geq 0, \quad b \geq 0, \quad c \geq 0, \quad d \geq 0, \quad t > 0$$

<u>Summary</u> Integration of $I_{25}(a,b,c,d,t)$ by parts yields a symmetric form of I_{26} in terms of I_{22}, J_{22} and I_{25} of Folders 22 and 25. An alternate, non-symmetric form is also derived for I_{26}.

RESEARCH: I26COMP, INTEGI26, I26ACOMP, INTEG26A

Folder 27: Evaluation of

$$I_{27}(a,b,t) = \int_0^t \frac{U(a,b,\tau)}{\sqrt{\tau}}d\tau \quad and \quad J_{27}(a,b,t) = \int_0^t U(a,b,\tau)\sqrt{\tau}d\tau$$

$$U(a,b,t) = e^{a^2t+2ab}erfc(a\sqrt{t} + b/\sqrt{t})$$

a>0, b>0, t>0

<u>Summary</u> These integrals are special cases of $I_{22}(a,b,c,t)$ *and* $J_{22}(a,b,c,t)$ for c=0. Integration by parts gives a relation between these integrals. A change of variable by $u = a\sqrt{t} + b/\sqrt{t}$ produced a formula which was numerically good only for small t. However, a recurrence relation which led to a series produced numerically feasible computational results over a larger range for t up to $t = t_m = b/a$ where t_m is the value of t which makes u a minimum.

Folder 28: Reciprocal Relations

$$\frac{1}{\sqrt{\pi}}E_{n+1/2}(x^2) = \sum_{k=1}^{n} A(n,k)\, i^{2k\text{-}1}erfc(x) \;, \quad n \geq 1$$

$$i^{2n\text{-}1}erfc(x) = \sum_{k=1}^{n} B(n,k)[E_{k+1/2}(x^2)/\sqrt{\pi}], \quad n \geq 1,$$

$$\frac{2x}{\sqrt{\pi}}E_{n+1/2}(x^2) = \sum_{k=0}^{n} A(n+1,k+1)\, i^{2k}erfc(x) \;, \quad n \geq 0$$

$$i^{2n}erfc(x) = 2x\sum_{k=0}^{n} B(n+1,k+1)[E_{k+1/2}(x^2)/\sqrt{\pi}], \quad n \geq 0,$$

with application to $I_3^c(a,b,c,T)$ of Folder 7

<u>Summary</u> Reciprocal relations connecting the iterated co-error functions and exponential integrals of half odd order are derived. Application to the computation of $I_3^c(a,b,c,T) = \int_T^\infty e^{-c^2w^2} erfc(aw)erfc(bw)dw$ of Folder 7 is presented.

Folder 29: Evaluation of

$$Y_n(a,b,T) = \int_T^\infty e^{-a^2w^2} i^n erfc(bw)dw, \quad a > 0, \quad b \geq 0, \quad n \geq -1$$

With Applications to

$I_2^c(a,b,c,T)$ of Folder 9, $I_5(a,b,T)$ of Folder 5 and $I_3^c(a,b,c,T)$ of Folder 7

Summary Rapidly convergent series are derived for $Y_n(a,b,T)$ in terms of the iterated co-error function. These series are exploited for the computation of $I_2^c(a,b,c,T)$, $I_5(a,b,T)$, and $I_3^c(a,b,c,T)$.

BECKSUBS: INTEGI29

Folder 1

ASYMPTOTIC EXPANSIONS AI_1, BI_1 AND CI_1 FOR

$$\mathbf{I_1(a,b,t) = \int_T^\infty e^{-a^2w^2}\frac{erf(bw)}{w^2}dw = \frac{1}{2}\int_0^t \frac{e^{-a^2/u}}{\sqrt{u}} erf\left(\frac{b}{\sqrt{u}}\right)du}$$

$$\mathbf{a>0,\quad b>0,\quad T=1/\sqrt{t},\ t>0}$$

Donald E. Amos, March 2000

Summary

Three approaches are taken to develop asymptotic expansions for small t. It turns out that expansions AI_1 and BI_1 are the same.

1.1 ASYMPTOTIC EXPANSION FOR I_1 (EXPANSION AI_1)

$$2I_1 = \int_0^t \frac{e^{-a^2/u}}{\sqrt{u}} \operatorname{erf}\left(\frac{b}{\sqrt{u}}\right)du = \int_0^t \frac{e^{-a^2/u}}{\sqrt{u}}du - \int_0^t \frac{e^{-a^2/u}}{\sqrt{u}} \operatorname{erfc}\left(\frac{b}{\sqrt{u}}\right)du$$

Let $u = 1/v$ in first integral and $u = b^2/v$ in the second

$$du = -\frac{1}{v^2}dv \qquad\qquad du = -\frac{b^2}{v^2}dv$$

$$2I_1 = \int_{1/t}^\infty \frac{e^{-a^2v}}{v^{3/2}}dv - b\int_{b^2/t}^\infty \frac{e^{-\gamma^2 v}}{v^{3/2}}\operatorname{erfc}(\sqrt{v})dv, \quad \gamma^2 = \frac{a^2}{b^2}$$

$$\beta t = w$$

$$\int_\alpha^\infty \frac{e^{-\beta t}}{t^n}dt = \beta^{n-1}\int_{\alpha\beta}^\infty \frac{e^{-w}}{w^n}dw = \frac{(\beta^{n-1}\alpha^{n-1})}{\alpha^{n-1}}\int_{\alpha\beta}^\infty \frac{e^{-w}}{w^n}dw = \frac{1}{\alpha^{n-1}}E_n(\alpha\beta)$$

$$\operatorname{erfc}(z) = \frac{e^{-z^2}}{z\sqrt{\pi}}\sum_{k=0}^N (-1)^k \frac{(1/2)_k}{z^{2k}} + r_N \quad \text{where} \quad |r_N| \le \frac{(1/2)_{N+1}}{z^{2n+2}}\cdot\frac{e^{-z^2}}{|z|\sqrt{\pi}}$$

First Integral: $\displaystyle\int_{1/t}^\infty \frac{e^{-a^2v}}{v^{3/2}}dv$ Let $u = e^{-a^2v}$ $\quad d\mathrm{v} = \dfrac{dv}{v^{3/2}}$

$$du = -a^2e^{-a^2v}dv \qquad \mathrm{v} = -2v^{-1/2}$$

$$= \frac{-2}{\sqrt{v}} e^{-a^2 v} \Bigg|_{1/t}^{\infty} - 2a^2 \int_{1/t}^{\infty} \frac{e^{-a^2 v}}{\sqrt{v}} dv \;=\; 2\sqrt{t} e^{-a^2/t} - 2a^2 \int_{1/t}^{\infty} \frac{e^{-a^2 v}}{\sqrt{v}} dv$$

Let v = w^2/a^2 in the last integral, then dv = $(2w/a^2)dw$ and we get

$$= 2\sqrt{t} e^{-a^2/t} - 2a\sqrt{\pi}\, erfc(a/\sqrt{t}) = \sqrt{\pi t}\left[\frac{2e^{-a^2/t}}{\sqrt{\pi}} - \frac{2a}{\sqrt{t}} erfc(a/\sqrt{t})\right] = 2\sqrt{\pi t}\; ierfc(a/\sqrt{t})$$

<u>Second Integral</u>: $\quad b\int_{b^2/t}^{\infty} \frac{e^{-\gamma^2 v}}{v^{3/2}} \operatorname{erfc}(\sqrt{v}) dv$

$$\sim \frac{b}{\sqrt{\pi}} \sum_{k=0}^{N} C_k \int_{b^2/t}^{\infty} \frac{e^{-(1+\gamma^2)v}}{v^{k+2}} dv + R_N, \qquad \gamma^2 = \frac{a^2}{b^2}$$

$$\sim \frac{b}{\sqrt{\pi}} \sum_{k=0}^{N} C_k \left(\frac{t}{b^2}\right)^{k+1} E_{k+2}\left(\frac{a^2+b^2}{t}\right) + R_N$$

$$C_k = (-1)^k (1/2)_k, \quad k = 0,1,\ldots \quad |R_N| \le \frac{b}{\sqrt{\pi}} |C_{N+1}| \left(\frac{t}{b^2}\right)^{N+2} E_{N+3}\left(\frac{a^2+b^2}{t}\right)$$

(Formula A) $\quad 2I_1 \sim 2\sqrt{t} e^{-a^2/t} - 2a\sqrt{\pi}\, \operatorname{erfc}(a/\sqrt{t}) - \frac{b}{\sqrt{\pi}} \sum_{k=0}^{N} C_k \left(\frac{t}{b^2}\right)^{k+1} E_{k+2}\left(\frac{a^2+b^2}{t}\right) + R_N$

$$2I_1 \sim 2\sqrt{t} e^{-a^2/t} - 2a\sqrt{\pi} \frac{\sqrt{t}}{a} \frac{e^{-a^2/t}}{\sqrt{\pi}} \sum_{k=0}^{M} C_k \left(\frac{t}{a^2}\right)^k + r_M$$

$$- \frac{b}{\sqrt{\pi}} \sum_{k=0}^{N} C_k \left(\frac{t}{b^2}\right)^k E_{k+2}\left(\frac{a^2+b^2}{t}\right) + R_N, \qquad |r_M| \le 2\sqrt{t}\, |C_{M+1}| \left(\frac{t}{a^2}\right)^{M+1} e^{-a^2/t}$$

(Formula B) $\quad 2I_1 \sim -2\sqrt{t} e^{-a^2/t} \sum_{k=1}^{M} C_k \left(\frac{t}{a^2}\right)^k + r_M$

$$- \frac{b}{\sqrt{\pi}} \sum_{k=0}^{N} C_k \left(\frac{t}{b^2}\right)^{k+1} E_{k+2}\left(\frac{a^2+b^2}{t}\right) + R_N$$

and the errors are less than the magnitude of the next term. To get the least error, stop summing at least one term less than the minimum term if error criterion (e.g. $|r_M|<10^{-6}$ is not met before the minimum). If the minimum term is greater than the error criterion, the *t* must be decreased to meet this error criterion. The minimum term is the best one can do for that value of *t*.

1.2 ASYMPTOTIC EXPANSION FOR I_1 (EXPANSION BI_1)

$$2I_1 = 2\sqrt{t}e^{-a^2/t} - 2a\sqrt{\pi}\,\mathrm{erfc}(a/\sqrt{t}) - \int_o^t \frac{e^{-a^2/u}}{\sqrt{u}}\mathrm{erfc}\left(\frac{b}{\sqrt{u}}\right)du$$

(A&S, page 302, 7.4.2) $$\mathrm{erfc}\frac{b}{\sqrt{u}} = 2\sqrt{\frac{u}{\pi}}e^{-b^2/u}\int_o^\infty e^{-uw^2-2bw}dw$$

Take the second integral:

$$\int_o^t \frac{e^{-a^2/u}}{\sqrt{u}}\mathrm{erfc}\left(\frac{b}{\sqrt{u}}\right)du = \frac{2}{\sqrt{\pi}}\int_o^t \frac{e^{-a^2/u}}{\sqrt{u}}\sqrt{u}e^{-b^2/u}du\int_o^\infty e^{-uw^2-2bw}dw$$

$$= \frac{2}{\sqrt{\pi}}\int_o^t e^{-(a^2+b^2)/u}du\int_o^\infty e^{-uw^2-2bw}dw = \frac{2}{\sqrt{\pi}}\int_o^\infty e^{-2bw}dw\int_o^t e^{-(a^2+b^2)/u-uw^2}du$$

Let $u = \dfrac{\sqrt{a^2+b^2}}{w}v$

$$= \frac{2}{\sqrt{\pi}}\int_o^\infty e^{-2bw}dw\int_o^{tw/\sqrt{a^2+b^2}} e^{-w\sqrt{a^2+b^2}\left(\frac{1}{v}+v\right)}\frac{\sqrt{a^2+b^2}}{w}dv$$

Take $v = e^s$ and exchange the orders of integration in the (w,s) plane to the right of the curve $s=\ln(tw/\sqrt{a^2+b^2})$

$$= \frac{2}{\sqrt{\pi}}\sqrt{a^2+b^2}\int_o^\infty \frac{e^{-2bw}}{w}\int_{-\infty}^{\ln(wt/\sqrt{a^2+b^2})} e^{-2w\sqrt{a^2+b^2}\cosh s}e^s\,ds\,dw$$

$$= \frac{2\sqrt{a^2+b^2}}{\sqrt{\pi}}\int_{-\infty}^\infty e^s ds\int_{e^s t^{-1}\sqrt{a^2+b^2}}^\infty e^{-2w(b+\sqrt{a^2+b^2}\cosh s)}\frac{dw}{w}$$

$$\text{Second Integral} = \frac{2\sqrt{a^2+b^2}}{\sqrt{\pi}}\int_{-\infty}^\infty e^s E_1\left(\frac{2e^s\sqrt{a^2+b^2}}{t}\left[b+\sqrt{a^2+b^2}\cosh s\right]\right)ds$$

Let $e^s = v$ $\quad = \dfrac{2\sqrt{a^2+b^2}}{\sqrt{\pi}} \int_o^\infty E_1\left(\dfrac{a^2+b^2}{t}\left[v^2 + \dfrac{2b}{\sqrt{a^2+b^2}} v + 1\right]\right) dv$

$$= \frac{2\sqrt{a^2+b^2}}{\sqrt{\pi}} \int_o^\infty E_1\left(\frac{a^2+b^2}{t}\left[\left(v + \frac{b}{\sqrt{a^2+b^2}}\right)^2 + \frac{a^2}{a^2+b^2}\right]\right) dv$$

Let $w = v + \dfrac{b}{\sqrt{a^2+b^2}}$ $\qquad dv = dw$

$$= \frac{2\sqrt{a^2+b^2}}{\sqrt{\pi}} \int_{\frac{b}{\sqrt{a^2+b^2}}}^\infty E_1\left(\frac{a^2+b^2}{t}\left[w^2 + \frac{a^2}{a^2+b^2}\right]\right) dw$$

$$= \frac{2\sqrt{a^2+b^2}}{\sqrt{\pi}} \int_{b/\sqrt{a^2+b^2}}^\infty E_1\left(Xw^2 + a^2/t\right) dw, \quad X = \frac{a^2+b^2}{t}$$

This integral is suitable by quadrature since $E_1(z) \sim e^{-z}/z$ and $E_1(Xw^2+a^2/t) \sim e^{-a^2/t} \dfrac{e^{-Xw^2}}{Xw^2 + a^2/t}$. For the scale of integration $\sigma = \sqrt{2/X}$ or $2\sqrt{2/X}$. For the asymptotic expansion, we integrate by parts with $c = a^2/t$

$\displaystyle\int_d^\infty \frac{E_1(Xw^2+C)}{2Xw}(2Xw)dw.$ $\qquad$ Let $u = 1/w$, $\qquad dv = E_1(Xw^2+C)\,(2Xw)dw$,

$= \dfrac{E_2(Xd^2+C)}{2Xd} - \dfrac{1}{2X}\displaystyle\int_d^\infty \frac{E_2(Xw^2+C)}{w^2}dw$ $\qquad du = -\dfrac{dw}{w^2}$, $\quad v = -E_2(Xw^2+C)$

$$\int_d^\infty \frac{E_1(Xw^2+C)}{w}dw = \frac{E_2(Xd^2+C)}{2Xd} - \frac{1}{2X}\int_d^\infty \frac{E_2(Xw^2+C)}{w^2}dw$$

Let $\quad u = 1/(2\mathrm{X}w^3)$ $\qquad dv = E_2(Xw^2+C)(2Xw)dw$

$\quad du = -3/(2Xw^4)$ $\qquad v = -E_3(Xw^2+C)$

$$= \frac{E_2(Xd^2+C)}{2Xd} - \frac{1}{2X}\left[\frac{E_3(Xd^2+C)}{2Xd^3} - \frac{3}{2X}\int_d^\infty \frac{E_3(Xw^2+C)}{w^4}dw\right]$$

$$= \frac{E_2(Xd^2+C)}{2Xd} - \frac{E_3(Xd^2+C)}{(2X)^2 d^3} + \frac{3}{(2X)^2}\int_d^\infty \frac{E_3(Xw^2+C)}{w^4}dw$$

Let $\quad u = 1/(2Xw^5) \qquad dv = E_3(Xw^2+C)(2Xw)dw$

$$du = -\frac{5}{2Xw^6}dw \qquad v = -E_4(Xw^2+C)$$

$$= \frac{E_2(Xd^2+C)}{2Xd} - \frac{E_3(Xd^2+C)}{(2X)^2 d^3} + \frac{3}{(2X)^2}\left[\frac{E_4(Xd^2+C)}{2Xd^5} - \frac{5}{2X}\int_\alpha^\infty \frac{E_4(Xw^2+C)}{w^6}dw\right]$$

$$= \frac{E_2(Xd^2+C)}{2Xd} - \frac{E_3(Xd^2+C)}{(2X)^2 d^3} + \frac{3}{(2X)^3}\frac{E_4(Xd+C)}{d^5} - \frac{1\cdot 3\cdot 5}{(2X)^3}\int_d^\infty \frac{E_4(Xw^2+C)}{w^6}dw$$

Generalize

$$= \frac{2\sqrt{a^2+b^2}}{\sqrt{\pi}}\left[\frac{E_2(Xd^2+C)}{2Xd} + \sum_{k=2}^{N} D_{k-1}\frac{E_{k+1}(Xd^2+C)}{(2X)^k d^{2k-1}} + R_N\right]$$

$$D_k = (-1)^k 1\cdot 3\cdot 5 ... (2k-1), \quad X = \frac{a^2+b^2}{t}, \quad d = \frac{b}{\sqrt{a^2+b^2}}, \quad C = \frac{a^2}{t}$$

$$= \frac{1}{\sqrt{\pi}}\left[\frac{t}{b}E_2\left(\frac{a^2+b^2}{t}\right) + \sum_{k=2}^{N} C_{k-1}\left(\frac{t}{a^2+b^2}\right)^k \frac{(a^2+b^2)^k}{(b^2)^{k-1/2}}E_{k+1}\left(\frac{a^2+b^2}{t}\right)\right] + R_N$$

$$= \frac{b}{\sqrt{\pi}}\left[\sum_{k=0}^{N} C_k\left(\frac{t}{b^2}\right)^{k+1} E_{k+2}\left(\frac{a^2+b^2}{t}\right)\right] + R_N = \text{Expansion of AI}_1$$

1.3 ASYMPTOTIC EXPANSION FOR I_1 (EXPANSION CI_1)

From (BI_1) second integral $= 2\dfrac{\sqrt{a^2+b^2}}{\sqrt{\pi}}\displaystyle\int_{\frac{b}{\sqrt{a^2+b^2}}}^\infty E_1\left(Xw^2 + a^2/t\right)dw, \quad X = \dfrac{a^2+b^2}{t}$

For a^2/t large, we use the asymptotic expansion for $E_1(z)$:

$$E_1(z) \sim e^{-z}\sum_{k=0}^{N}\frac{(-1)^k k!}{z^{k+1}}$$

$$\text{Second integral} = \frac{2\sqrt{a^2+b^2}}{\sqrt{\pi}}\sum_{k=0}^{N} C_k\int_{\frac{b}{\sqrt{a^2+b^2}}}^\infty \frac{e^{-Xw^2-a^2/t}}{\left(Xw^2+a^2/t\right)^{k+1}}dw + R_N$$

Let $u = Xw^2 + \dfrac{a^2}{t}, \quad w = \sqrt{\dfrac{u - a^2/t}{X}} \qquad dw = \dfrac{1}{2\sqrt{X}}\left(u - a^2/t\right)^{-1/2} du \qquad C_k = (-1)^k k!$

$$= \frac{2\sqrt{a^2+b^2}}{2\sqrt{X}\sqrt{\pi}} \sum_{k=0}^{N} C_k \int_{(a^2+b^2)/t}^{\infty} \frac{e^{-u}}{u^{k+1}} \frac{du}{\sqrt{u - a^2/t}} + R_N$$

$$= \frac{\sqrt{t}}{\sqrt{\pi}} \sum_{k=0}^{N} C_k \int_{(a^2+b^2)/t}^{\infty} \frac{e^{-u}}{u^{k+1}} \frac{du}{\sqrt{u - a^2/t}} + R_N$$

$$\left(u - a^2/t\right)^{-1/2} = u^{-1/2}\left(1 - a^2/tu\right)^{-1/2} = u^{-1/2}\, {}_2F_1\!\left(\frac{1}{2}, 1; 1; \frac{a^2}{tu}\right)$$

$$\text{Integral} = \sqrt{\frac{t}{\pi}} \sum_{k=0}^{N} C_k \sum_{n=0}^{\infty} \frac{(1/2)_n}{n!} \left(\frac{a^2}{t}\right)^n \int_{(a^2+b^2)/t}^{\infty} \frac{e^{-u}}{u^{k+n+3/2}} du$$

$$= \sqrt{\frac{t}{\pi}} \sum_{k=0}^{N} C_k \sum_{n=0}^{\infty} \frac{(1/2)_n}{n!} \left(\frac{a^2}{t}\right)^n \left(\frac{t}{a^2+b^2}\right)^{k+n+1/2} E_{k+n+3/2}\!\left(\frac{a^2+b^2}{t}\right) + R_N$$

$$= \sqrt{\frac{t}{\pi}} \sum_{k=0}^{N} C_k \left(\frac{t}{a^2+b^2}\right)^{k+1/2} \sum_{n=0}^{\infty} \frac{(1/2)_n}{n!} \left(\frac{a^2}{a^2+b^2}\right)^n E_{k+n+3/2}\!\left(\frac{a^2+b^2}{t}\right) + R_N$$

using $\displaystyle\int_{\alpha}^{\infty} \frac{e^{-\beta t}}{t^n} dt = \frac{1}{\alpha^{n-1}} E_n(\alpha\beta), \quad \left(\frac{1}{2}\right)_n = \frac{1}{2}\cdot\frac{3}{2}\cdots\left(\frac{1}{2} + n - 1\right)$

Folder 2

CONVERGENT EXPANSIONS (I) AND (II) FOR I_1

$$\mathbf{I_1(a,b,t)} = \int_T^\infty e^{-a^2w^2}\frac{\operatorname{erf}(bw)}{w^2}dw = \frac{1}{2}\int_0^t \frac{e^{-a^2/u}}{\sqrt{u}}\operatorname{erf}\left(\frac{b}{\sqrt{u}}\right)du$$

$$a>0,\quad b>0,\quad T=1/\sqrt{t},\quad t>0$$

Donald E. Amos, May 2000

Summary

(I) $$I_1(a,b,t) = \sqrt{\pi t}\ \operatorname{ierfc}\left(\frac{a}{\sqrt{t}}\right) - \sqrt{t}\ \operatorname{erfc}(\sqrt{X}) + \frac{b}{\sqrt{\pi}}E_1(X) - \sqrt{\frac{Xt}{\pi}}\sum_{k=1}^{\infty}\frac{(-1/2)_k}{k!}\left(\frac{a^2}{a^2+b^2}\right)^k E_{k+1/2}(X),$$

(II) $$I_1(a,b,t) = \sqrt{\pi t}\ \operatorname{erf}\left(\frac{b}{\sqrt{t}}\right)\operatorname{ierfc}\left(\frac{a}{\sqrt{t}}\right) + \frac{b}{\sqrt{\pi}}E_1(X) - \frac{ab}{\sqrt{\pi X t}}\sum_{k=0}^{\infty}\frac{(1/2)_k}{k!}\left(\frac{b^2}{a^2+b^2}\right)^k E_{k+3/2}(X)$$

where

$$X = \left(a^2+b^2\right)/t.$$

Use (I) for $a \le b$ where $\dfrac{a^2}{a^2+b^2} \le \dfrac{1}{2}$ and (II) for $a > b$ where $\dfrac{b^2}{a^2+b^2} \le \dfrac{1}{2}$.

Analysis

I_1 is defined by $$I_1 = \frac{1}{2}\int_o^t \frac{e^{-a^2/u}}{\sqrt{u}}\operatorname{erf}\left(\frac{b}{\sqrt{u}}\right)du.$$

Replacing erf(x) by 1-erfc(x) we get 2 terms

$$2I_1 = \int_o^t \frac{e^{-a^2/u}}{\sqrt{u}}du - \int_o^t \frac{e^{-a^2/u}}{\sqrt{u}}\operatorname{erfc}\left(\frac{b}{\sqrt{u}}\right)du.$$

In Folder 1, the first integral is evaluated in the derivation of an asymptotic form called AI_1 and the second integral is converted to an integral over E_1 in the derivation of a second asymptotic expansion called BI_1:

$$2I_1 = 2\sqrt{t}e^{-a^2/t} - 2a\sqrt{\pi}\operatorname{erfc}(a/\sqrt{t}) - 2\sqrt{\frac{a^2+b^2}{\pi}}\int_{\frac{b}{\sqrt{a^2+b^2}}}^{\infty} E_1\left(Xw^2+\frac{a^2}{t}\right)dw, \qquad X = \frac{a^2+b^2}{t}.$$

A series expansion for this integral is developed and this representation for I_1 is called Expansion (I).

2.1 CONVERGENT EXPANSION (I) FOR I_1

We start with the second integral from BI_1 in **Part 1 (Folder 1)**.

$$S = 2\sqrt{\frac{a^2+b^2}{\pi}}\int_{b/\sqrt{a^2+b^2}}^{\infty} E_1\left(Xw^2+\frac{a^2}{t}\right)dw, \qquad X = \frac{a^2+b^2}{t}$$

Let $v = Xw^2+a^2/t$, $\qquad w = \frac{1}{\sqrt{X}}\sqrt{v-a^2/t}, \qquad dw = \frac{1}{2\sqrt{X}}\left(v-\frac{a^2}{t}\right)^{-1/2}dv$

$$S = 2\sqrt{\frac{a^2+b^2}{\pi}}\int_X^{\infty}\frac{E_1(v)}{2\sqrt{Xv}}\left(1-\frac{a^2}{tv}\right)^{-1/2}dv$$

Now, using formula (4) below, $\qquad \left(1-\frac{a^2}{tv}\right)^{-1/2} = \sum_{k=0}^{\infty}\frac{(1/2)_k}{k!}\left(\frac{a^2}{tv}\right)^k$

$$S = \sqrt{\frac{t}{\pi}}\left\{\int_X^{\infty}\frac{E_1(v)}{\sqrt{v}}dv+\sum_{k=1}^{\infty}\frac{(1/2)_k}{k!}\left(\frac{a^2}{t}\right)^k\int_X^{\infty}\frac{E_1(v)}{v^{k+1/2}}dv\right\}$$

Formulas (5) and (6) below give

$$S = \sqrt{\frac{t}{\pi}}\left\{2\sqrt{\pi}\,\mathrm{erfc}\left(\sqrt{X}\right)-2\sqrt{X}E_1(X)+\sum_{k=1}^{\infty}\frac{(1/2)_k}{k!}\left(\frac{a^2}{t}\right)^k\frac{1}{X^{k-1/2}}\frac{\left[E_1(X)-E_{k+1/2}(X)\right]}{(k-1/2)}\right\}$$

Now, $\qquad \frac{(1/2)_k}{k!(k-1/2)} = \frac{\frac{1}{2}\cdot\frac{3}{2}\;\cdots\left(k-\frac{3}{2}\right)\left(k-\frac{1}{2}\right)}{k!(k-1/2)} = (-2)\frac{\left(-\frac{1}{2}\right)\left[\frac{1}{2}\;\cdots\left(k-\frac{3}{2}\right)\right]}{k!} = -2\frac{\left(-\frac{1}{2}\right)_k}{k!}$

$$S = \sqrt{\frac{t}{\pi}}\left\{2\sqrt{\pi}\,\mathrm{erfc}\left(\sqrt{X}\right)-2\sqrt{X}E_1(X)-2\sum_{k=1}^{\infty}\frac{(-1/2)_k}{k!}\left(\frac{a^2}{a^2+b^2}\right)^k\sqrt{X}E_1(X)\right.$$
$$\left.+2\sum_{k=1}^{\infty}\frac{(-1/2)_k}{k!}\left(\frac{a^2}{a^2+b^2}\right)^k\sqrt{X}E_{k+1/2}(X)\right\}$$

Using (4)

$$S = 2\sqrt{t}\,\mathrm{erfc}\left(\sqrt{X}\right)-2\sqrt{\frac{Xt}{\pi}}\left[1-\frac{a^2}{a^2+b^2}\right]^{1/2}E_1(X)+2\sqrt{\frac{Xt}{\pi}}\sum_{k=1}^{\infty}\frac{(-1/2)_k}{k!}\left(\frac{a^2}{a^2+b^2}\right)^k E_{k+1/2}(X)$$

or

$$S = 2\sqrt{t}\,\mathrm{erfc}\left(\sqrt{X}\right)-\frac{2b}{\sqrt{\pi}}E_1(X)+2\sqrt{\frac{Xt}{\pi}}\sum_{k=1}^{\infty}\frac{(-1/2)_k}{k!}\left(\frac{a^2}{a^2+b^2}\right)^k E_{k+1/2}(X) \quad \text{since } X=(a^2+b^2)/t.$$

List of Formulae:

1) $$E_{1/2}(z) = \int_1^\infty \frac{e^{-zw}}{\sqrt{w}}dw \overset{w = v^2/z}{=} \int_{\sqrt{z}}^\infty e^{-v^z}\frac{\sqrt{z}}{v}\frac{2vdv}{z} = \sqrt{\frac{\pi}{z}}\,\mathrm{erfc}\left(\sqrt{z}\right)$$

2) $$E_{k+1/2}(z) = \frac{1}{k-1/2}\left[e^{-z} - zE_{k-1/2}(z)\right], \quad k = 1,2,...$$

3) $(\alpha)_k = \alpha(\alpha+1)\ldots(\alpha+\mathrm{k}-1)$

4) $$(1+z)^\alpha = {}_2F_1(-\alpha,b;b;-z) = \sum_{k=0}^\infty \frac{(-\alpha)_k}{k!}(-z)^k, \qquad |z|<1$$

5) $$\int_X^\infty \frac{E_1(v)}{\sqrt{v}}dv = \int_X^\infty \frac{dv}{\sqrt{v}}\int_v^\infty \frac{e^{-w}}{w}dw$$

Now exchange the orders of integration in the (w,v) plane to the right of the line $v=w$ and to the right of the quadrant with a corner (X,X). Then (5) becomes

$$= \int_X^\infty \frac{dw}{w}\int_X^w \frac{e^{-w}}{\sqrt{v}}dv$$

$$= \int_X^\infty \frac{e^{-w}}{w}\left[2\sqrt{v}\right]_X^w dw = 2\int_X^\infty \frac{e^{-w}}{w}\left[\sqrt{w}-\sqrt{X}\right]dw$$

$$= 2\int_X^\infty \frac{e^{-w}}{\sqrt{w}}dw - 2\sqrt{X}\int_X^\infty \frac{e^{-w}}{w}dw$$

$$= 2\sqrt{X}\int_1^\infty \frac{e^{-Xv}}{\sqrt{v}}dv - 2\sqrt{X}E_1(X)$$

From (1),

$$= 2\sqrt{\pi}\,\mathrm{erfc}\left(\sqrt{X}\right) - 2\sqrt{X}E_1(X)$$

6) $$\int_X^\infty \frac{E_1(v)}{v^{k+1/2}}dv \overset{v = Xw}{=} \int_1^\infty \frac{E_1(Xw)Xdw}{X^{k+1/2}w^{k+1/2}} = \frac{1}{X^{k-1/2}}\int_1^\infty \frac{E_1(Xw)}{w^{k+1/2}}dw$$

$$= \frac{1}{X^{k-1/2}}\int_1^\infty \frac{dw}{w^{k+1/2}}\int_1^\infty \frac{e^{-Xwv}}{v}dv = \int_1^\infty \frac{dv}{v}E_{k+1/2}(Xv)\cdot\frac{1}{X^{k-1/2}}$$

Using (2) above we get

$$\int_X^\infty \frac{E_1(v)}{v^{k+1/2}}dv = \frac{1}{k-1/2}\left[E_1(X) - X\int_1^\infty E_{k-1/2}(Xv)dv\right]\cdot\frac{1}{X^{k-1/2}}$$

$$\int_X^\infty \frac{E_1(v)}{v^{k+1/2}}dv = \frac{1}{(k-1/2)}\left[E_1(X) + X\int_1^\infty E'_{k+1/2}(Xv)dv\right]\cdot\frac{1}{X^{k-1/2}}$$

$$= \frac{1}{(k-1/2)}\left[E_1(X) + \int_X^\infty E'_{k+1/2}(w)dw\right]\cdot\frac{1}{X^{k-1/2}}$$

$$= \frac{1}{(k-1/2)}\left[E_1(X) - E_{k+1/2}(X)\right]\cdot\frac{1}{X^{k-1/2}}$$

<u>Notes on Convergence</u>:

$$\frac{(-1/2)_k}{k!} = \frac{\Gamma(k-1/2)}{\Gamma(-1/2)\Gamma(k+1)} \sim \frac{k^{-3/2}}{\Gamma(-1/2)} = \frac{k^{-3/2}}{-2\sqrt{\pi}}$$

$E_{k+1/2}(X) \sim \dfrac{e^{-X}}{X+k+1/2}$. Therefore the convergence is like a power series with coefficients $O(k^{-5/2})$, though for large *X*, the convergence is more like a power series with coefficients $O(k^{-3/2})$ since it takes many terms to make $(X+k+1/2)$ contribute to the convergence.

<u>Notes on Numerical Evaluation of</u> $\mathbf{E_{k+1/2}(X)}$:

While conceptually, $E_{k+1/2}(X)$ can be evaluated by forward recurrence on (1) and (2), the process is numerically unstable when *X* is much larger than 5 or 10. The numerically stable way is to find $E_{K+1/2}(X)$ where *K* is close to *X* and recur away from *K* in a backward and forward direction. This is the subject of the next comment.

<u>A Recipe for Numerical Generation of</u> $E_{k-1/2}(z)$

First we derive a 3-term homogeneous relation for $E_{k-1/2}(z)$ by eliminating e^{-z} in (2): Set indices *k* and *k*+1 in formula (2) and subtract,

$$(k-1/2)E_{k+1/2} = e^{-z} - zE_{k-1/2}(z)$$

$$(k+1/2)E_{k+3/2} = e^{-z} - zE_{k+1/2}(z)$$

$$(k+1/2)E_{k+3/2}(z) + (z-k+1/2)E_{k+1/2}(z) - zE_{k-1/2}(z) = 0$$

or, shifting indices down by 1 gives

7) $$(k-1/2)E_{k+1/2}(z)+(z-k+3/2)E_{k-1/2}(z)-zE_{k-3/2}(z) \;=\; 0$$

It is the change of sign in $(z-k+3/2)$ which makes the computation unstable.

Procedure:

Select index $N \;=\; [z+3/2]$ = integer part of $z+3/2$. Generate a y_k sequence by backward recurrence from $k = N$ starting with $y_N = 1$ and $y_{N+1} = 0$. Then generate a w_k sequence by backward recurrence starting with $w_N = 0$, $w_{N+1} = 1$. Notice that because the recurrence is homogeneous, these sequences are solutions of (7) when multiplied by constants and summed:

$$r_k = C_N y_k + C_{N+1} w_k \quad k = 1,2,\ldots \mathrm{N},\; N+1$$

Now, for $k = 1$ and $k = 2$, $r_1 = E_{1/2}(z)$, $r_2 = E_{3/2}(z)$ from formula (2):

$$C_N y_1 + C_{N+1} w_1 \;=\; E_{1/2}(z)$$

$$C_N y_2 + C_{N+1} w_2 \;=\; E_{3/2}(z)$$

This gets C_N and $C_{N+1.}$ Furthermore, each r_k is $E_{k-1/2}(z)$ since $r_1 = E_{1/2}(z)$, $r_2 = E_{3/2}(z)$, then r_3 and $E_{5/2}(z)$ satisfy the same equation. Hence, $r_3 = E_{5/2}(z)$. By induction, we get $r_k = E_{k-1/2}(z)$ for $k = 1,2\ldots,N-1$.

For $k = N$-2,

$$(N-3/2)E_{N-1/2}(z)-(z-N+5/2)E_{N-3/2}(z)-zE_{N-5/2}(z) \;=\; 0$$

$$(N-3/2)C_N-(z-N+5/2)r_{N-1}-zr_{N-2} \;=\; 0$$

Thus, $C_N \;=\; E_{N-1/2}$ since $r_N = 1\cdot C_N + (0)\cdot C_{N+1} = C_N$.

For $k = N$-1, we get

$$(N-1/2)C_{N+1}-(z-N+3/2)r_N-zr_{N-1} \;=\; 0$$

and since $r_{N+1} = 0\cdot C_N + 1\cdot C_{N+1}$, then $C_{N+1} = E_{N+1/2}(z)$. Thus, we can use C_N and C_{N+1} as starting values for forward recurrence on either (2) or (7) for $k > N+1$.

Numerical Results:

Comparing the above process with quadrature on $\int_1^\infty \frac{e^{-zt}}{t^k}dt$ verified the theory. C_N and C_{N+1} were values of E. However for $z > 15$, the denominator $D = y_1w_2\text{-}w_1y_2$ in the solution for C_N and C_{N+1} lost significance because the sequences y_k and w_k looked nearly the same for small indices, especially for $k = 1$ and 2. The long recurrence lost the effect of the initial condition at $k = N$ and $k = N+1$. For $z = 30$, all significance was lost. On the other hand, the quadrature worked fine with $\sigma = 3/(z+k)$ with 5-10 steps, being nearly constant at 8 steps. All computations were done in double precision arithmetic.

It appears that the best procedure is to evaluate $E_{N-1/2}(z)$ by quadrature and recur backward and forward with (2) from $k = N$. However, the theory for codes EXINT or DEXINT does not require the order to be an integer. Consequently, DEXINT was modified to make a new code DHEXINT for half odd integer orders.

2.2 CONVERGENT EXPANSION (II) FOR I_1

$$2\,I_1 = \int_0^t \frac{e^{-a^2/u}}{\sqrt{u}}\,\mathrm{erf}\left(\frac{b}{\sqrt{u}}\right)du = \frac{2}{\sqrt{\pi}}\int_0^t \frac{e^{-a^2/u}}{\sqrt{u}}\int_0^{b/\sqrt{u}} e^{-v^2}dv$$

Exchange the orders of integration under the curve $v = b/\sqrt{u}$ or $u = b^2/v^2$

$$= \frac{2}{\sqrt{\pi}}\int_0^{b/\sqrt{t}} e^{-v^2}dv\int_0^t \frac{e^{-a^2/u}}{\sqrt{u}}du + \frac{2}{\sqrt{\pi}}\int_{b/\sqrt{t}}^\infty e^{-v^2}dv\int_0^{b^2/v^2}\frac{e^{-a^2/u}}{\sqrt{u}}du$$

Let $u = t/w$ in first integral

$$= \mathrm{erf}\left(\frac{b}{\sqrt{t}}\right)\int_1^\infty \frac{e^{-(a^2/t)w}}{w^2}\frac{\sqrt{wt}}{\sqrt{t}}dw + \frac{2}{\sqrt{\pi}}\int_{b/\sqrt{t}}^\infty e^{-v^2}dv\int_0^{b^2/v^2}\frac{e^{-a^2/u}}{\sqrt{u}}du$$

Let $u = b^2/(v^2w)$ in second integral

$$= \sqrt{t}\,\mathrm{erf}\left(\frac{b}{\sqrt{t}}\right)E_{3/2}\left(\frac{a^2}{t}\right) + \frac{2}{\sqrt{\pi}}\int_{b/\sqrt{t}}^\infty e^{-v^2}dv\int_1^\infty \frac{e^{-(a^2v^2/b^2)w}}{b}\frac{b^2v\sqrt{w}}{v^2w^2}dw$$

$$= \sqrt{t}\,\mathrm{erf}\left(\frac{b}{\sqrt{t}}\right)E_{3/2}\left(\frac{a^2}{t}\right) + \frac{2b}{\sqrt{\pi}}\int_{b/\sqrt{t}}^\infty \frac{e^{-v^2}}{v}dv\int_1^\infty \frac{e^{-(a^2v^2/b^2)w}}{w^{3/2}}dw$$

$$= \sqrt{t}\,\mathrm{erf}\left(\frac{b}{\sqrt{t}}\right)E_{3/2}\left(\frac{a^2}{t}\right) + \frac{2b}{\sqrt{\pi}}\int_{b/\sqrt{t}}^\infty \frac{e^{-v^2}}{v}E_{3/2}\left(\frac{a^2v^2}{b^2}\right)dv$$

$$2I_1 = \sqrt{t}\,\mathrm{erf}\left(\frac{b}{\sqrt{t}}\right)E_{3/2}\left(\frac{a^2}{t}\right)+\frac{2b}{\sqrt{\pi}}\int_1^\infty \frac{dw}{w^{3/2}}\int_{b/\sqrt{t}}^\infty \frac{e^{-v^2}}{v}e^{-[(a^2w)/b^2]v^2}dv$$

Let $v=\dfrac{b}{\sqrt{t}}\sqrt{u}$

$$= \sqrt{t}\,\mathrm{erf}\left(\frac{b}{\sqrt{t}}\right)E_{3/2}\left(\frac{a^2}{t}\right)+\frac{2b}{\sqrt{\pi}}\int_1^\infty \frac{dw}{w^{3/2}}\int_1^\infty \frac{e^{-(1+a^2w/b^2)(b^2/t)u}}{(b/\sqrt{t})\sqrt{u}}\frac{b}{2\sqrt{t}}\frac{du}{\sqrt{u}}$$

$$= \sqrt{t}\,\mathrm{erf}\left(\frac{b}{\sqrt{t}}\right)E_{3/2}\left(\frac{a^2}{t}\right)+\frac{b}{\sqrt{\pi}}\int_1^\infty \frac{dw}{w^{3/2}}E_1\left(\frac{b^2+a^2w}{t}\right)$$

Let $(b^2+a^2w)/t = s$

$$= \sqrt{t}\,\mathrm{erf}\left(\frac{b}{\sqrt{t}}\right)E_{3/2}\left(\frac{a^2}{t}\right)+\frac{b}{\sqrt{\pi}}\frac{a}{\sqrt{t}}\int_X^\infty \frac{E_1(s)ds}{(s-b^2/t)^{3/2}}, \qquad X=\frac{a^2+b^2}{t}$$

Now, this integral is of the same form as that in Convergent Expansion (I),

$$\int_x^\infty \frac{E_1(s)}{\sqrt{s-a^2/t}}ds$$

where 3/2 replaces 1/2 in the denominator and b^2/t replaces a^2/t.

Thus,

$$\int_X^\infty \frac{E_1(s)}{\sqrt{s-b^2/t}}ds = \int_X^\infty \frac{E_1(s)}{s^{3/2}\left(1-b^2/ts\right)^{3/2}}ds$$

$$= \sum_{k=0}^\infty \frac{(3/2)_k}{k!}\left(\frac{b^2}{t}\right)^k\int_X^\infty \frac{E_1(s)}{s^{k+3/2}}ds$$

and using (6) of expansion (I),

$$\int_X^\infty \frac{E_1(s)}{s^{k+3/2}}ds = \frac{1}{(k+1/2)}\left[E_1(X)-E_{k+3/2}(X)\right]\frac{1}{X^{k+1/2}}$$

we get

$$\int_X^{\infty} \frac{E_1(s)}{\sqrt{s-b^2/t}}\,ds = \sum_{k=0}^{\infty} \frac{(3/2)_k}{k!}\left(\frac{b^2}{t}\right)^k \frac{E_1(X)}{(k+1/2)X^{k+1/2}} - \sum_{k=0}^{\infty} \frac{(3/2)_k}{k!}\left(\frac{b^2}{t}\right)^k \frac{E_{k+3/2}(X)}{(k+1/2)X^{k+1/2}}$$

Now, $(3/2)_k = 2(1/2)(3/2)\ldots(k\text{-}1/2)(k+1/2) = 2(1/2)_k(k+1/2)$

$$= \frac{2}{\sqrt{X}}\sum_{k=0}^{\infty} \frac{(1/2)_k}{k!}\left(\frac{b^2}{a^2+b^2}\right)^k E_1(X) - \sum_{k=0}^{\infty} \frac{2}{\sqrt{X}}\frac{(1/2)_k}{k!}\left(\frac{b^2}{a^2+b^2}\right)^k E_{k+3/2}(X)$$

$$= \frac{2E_1(x)}{\sqrt{X}}\left(1-\frac{b^2}{a^2+b^2}\right)^{-1/2} - \frac{2}{\sqrt{X}}\sum_{k=0}^{\infty} \frac{(1/2)_k}{k!}\left(\frac{b^2}{a^2+b^2}\right)^k E_{k+3/2}(X)$$

$$= \frac{2\sqrt{t}}{a}E_1(X) - \frac{2}{\sqrt{X}}\sum_{k=0}^{\infty} \frac{(1/2)_k}{k!}\left(\frac{b^2}{a^2+b^2}\right)^k E_{k+3/2}(X)$$

The final expansions are

(I) $$I_1(a,b,t) = \sqrt{\pi t}\ \mathrm{ierfc}\left(\frac{a}{\sqrt{t}}\right) - \sqrt{t}\ \mathrm{erfc}(\sqrt{X}) + \frac{b}{\sqrt{\pi}}E_1(X) - \sqrt{\frac{Xt}{\pi}}\sum_{k=1}^{\infty}\frac{(-1/2)_k}{k!}\left(\frac{a^2}{a^2+b^2}\right)^k E_{k+1/2}(X),$$

(II) $$I_1(a,b,t) = \sqrt{\pi t}\ \mathrm{erf}\left(\frac{b}{\sqrt{t}}\right)\mathrm{ierfc}\left(\frac{a}{\sqrt{t}}\right) + \frac{b}{\sqrt{\pi}}E_1(X) - \frac{ab}{\sqrt{\pi X t}}\sum_{k=0}^{\infty}\frac{(1/2)_k}{k!}\left(\frac{b^2}{a^2+b^2}\right)^k E_{k+3/2}(X)$$

where

$$X = \left(a^2+b^2\right)/t$$

and we have used the relations

$$E_{3/2}(x) = 2\sqrt{\pi}\ ierfc(x)\ , \qquad ierfc(x) = -x\ erfc(x) + \frac{e^{-x^2}}{\sqrt{\pi}} \sim \frac{e^{-x^2}}{2x^2\sqrt{\pi}} \quad \text{for } x \to \infty .$$

Comments on Expansions (I) and (II)

Note that (I) converges for $a^2/(a^2+b^2) \le 1/2$ very nicely where $b^2/(a^2+b^2) > 1/2$ and (II) does not converge as well. On the other hand, where $a^2/(a^2+b^2) > 1/2$ and (I) does not converge very well, (II) converges very well with $b^2/(a^2+b^2) \le 1/2$. Thus, (I) and (II) complement one another and the problem of numerical evaluation is solved. The worse case for $a^2/(a^2+b^2) = 1/2$ or $b^2/(a^2+b^2) = 1/2$ converges like $2^{-k}k^{-3/2}$ or $2^{-k}k^{-1/2}$ and for $k = 20$ we get $O(1/2^{20}) \sim 0((10^{-3})^2) = O(10^{-6})$ with corresponding truncation errors (relative errors). For 40 terms we get $0(10^{-12})$, i.e. (I) converges well for $a \le b$ and (II) converges well for $a > b$.

The relations were written in terms of ierfc(*) because the subtraction in the recurrence form loses significant digits as the argument gets larger. For example at x=10, the asymptotic form shows that we have lost about 2 significant digits. Subroutine DINERFC computes accurately and takes this difference into account by using the full asymptotic expansion for x>7.

Folder 3

Quadrature For

$$I_6(a,b,T) = \int_T^{\infty} \frac{\mathrm{erf}(aw)\mathrm{erf}(bw)}{w^2}\,dw = \frac{1}{2}\int_0^t \frac{1}{\sqrt{u}}\,\mathrm{erf}\left(\frac{a}{\sqrt{u}}\right)\mathrm{erf}\left(\frac{b}{\sqrt{u}}\right)du$$

$$a>0,\quad b>0,\quad T=\frac{1}{\sqrt{t}},\quad t>0$$

Donald E. Amos, June 2000

Summary

A change of variable converts the integral to a form which contains no singularities and whose integrand is analytic

$$I = \int_{1/\sqrt{t}}^{\infty} \frac{erf(aw)erf(bw)}{w^2}\,dw$$

The idea is to replace each erf by 1 when the argument is 6 or larger. In the first development, a value of $W_M = \max(6/a, 6/b)$ is used and for $w > W_M$ each erf function is replaced by 1.

A refinement of this procedure is developed which replaces the erf by 1 as soon as aw or bw exceeds 6. This refinement makes the procedure more robust for large t and a or b large where a unit step function is approximated near $w = 0$. In this case, the quadrature is confined to an area where it can address the sharp changes in the integrand.

It should be noted also that the integrand should be computed as

$$\frac{\mathrm{erf}(aw)}{w}\cdot\frac{\mathrm{erf}(bw)}{w}$$

since each is $O(a)$ or $O(b)$ and squaring the denominator can make use of only half of the underflow range of the machine.

3.1 Quadrature For $I_6(a,b,T)$

$$I_6(a,b,T) = \int_{1/\sqrt{t}}^{\infty} \frac{erf(aw)erf(bw)}{w^2}\,dw$$

and the integrand is analytic for all w.

Since for large z, $\mathrm{erf}(z) = 1$ with an error $\dfrac{e^{-z^2}}{z\sqrt{\pi}}$, it follows that for $z \ge 6$, the error is uniformly bounded by $O(10^{-16})$ which is the accuracy of double precision arithmetic.

Now if $aw \geq 6$ and $bw \geq 6$ then the tail of the integral can be computed as

$$Q = \int_w^\infty \frac{dw}{w^2} = \frac{1}{w}$$

with an accuracy of 10^{-16} or better.

Procedure:

If $\quad T = \dfrac{1}{\sqrt{t}} > W_M = \max\left(\dfrac{6}{a}, \dfrac{6}{b}\right)$, then

$$I = \int_T^\infty \frac{dw}{w^2} = \frac{1}{T}$$

If $\quad T \leq W_M$, then

$$I = \int_T^{W_M} \frac{\mathrm{erf}(aw)\mathrm{erf}(bw)}{w^2} dw + \frac{1}{W_M}$$

where the integral on (T, W_M) is computed by DGAUS8. Here all functions are analytic and DGAUS8 should have no problem attaining an accuracy up to 10^{-15}.

For $t \to 0$, $I \to \dfrac{1}{T}$. This quadrature was checked for $10^{-3} \leq a, b \leq 10^3$, $10^{-3} \leq t \leq 10^4$.

3.2 Quadrature Refinement For I_6(a,b,T)

$$I_6(a,b,T) = \int_T^\infty \frac{erf(aw)erf(bw)}{w^2} dw$$

For large z, erf(z) = 1 with an error $\dfrac{e^{-z^2}}{z\sqrt{\pi}}$.

Therefore, for $z \geq 6$, we replace the erf function with 1 and obtain an error which is uniformly on the order of $O(10^{-16})$, which is sufficient for double precision arithmetic.
In a previous development, the erf functions were replaced by 1 for all w greater than the max of $6/a$ and $6/b$. In this development we replace an erf function with 1 as soon as the argument is 6 or larger. This is especially advantageous when a and b are widely separated, since one can use shorter length quadratures. It also has the advantage of confining a quadrature to the area of a unit step if a or b is large.

DGAUS8 is very robust and capable of handling very rapid changes in an integrand.

Let $W_m = \min\left(\frac{6}{a}, \frac{6}{b}\right)$, $\quad W_M = \max\left(\frac{6}{a}, \frac{6}{b}\right)$

and let $\quad x = a \quad \text{if } W_m = \frac{6}{b} \text{ or } (b \geq a)$

i.e. $\quad x = \min(a,b)$

$x = b \quad \text{if } W_m = \frac{6}{a} \text{ or } (b < a)$

Case I $\quad \mathrm{T} \leq \mathrm{W_m}$, $\quad \mathrm{I} = \int_{\mathrm{T}}^{\mathrm{W_m}} \frac{\mathrm{erf(aw)erf(bw)}}{\mathrm{w}^2}\mathrm{dw} + \int_{\mathrm{W_m}}^{\mathrm{W_M}} \frac{\mathrm{erf(xw)}}{\mathrm{w}^2}\mathrm{dw} + \int_{\mathrm{W_M}}^{\infty} \frac{\mathrm{dw}}{\mathrm{w}^2}$

Case II $\quad W_m < \mathrm{T} \leq \mathrm{W_M}$, $\quad \mathrm{I} = \int_{\mathrm{T}}^{\mathrm{W_M}} \frac{\mathrm{erf(xw)}}{\mathrm{w}^2}\mathrm{dw} + \int_{\mathrm{W_M}}^{\infty} \frac{\mathrm{dw}}{\mathrm{w}^2}$

Case III $\quad \mathrm{W_M} < \mathrm{T} < \infty$, $\quad \mathrm{I} = \int_{\mathrm{T}}^{\infty} \frac{\mathrm{dw}}{\mathrm{w}^2}$

Now, we evaluate the last two integrals.

Let $\quad P = \int_L^U \frac{\operatorname{erf}(xw)}{w^2} dw$ and integrate by parts

$$u = \operatorname{erf}(xw), \quad dv = \frac{1}{w^2} dw$$

$$du = \frac{2x}{\sqrt{\pi}} e^{-x^2 w^2} dw, \quad v = -\frac{1}{w}$$

$$P(x,L,U) = \left. \frac{-\operatorname{erf}(xw)}{w} \right|_L^U + \frac{2x}{\sqrt{\pi}} \int_L^U \frac{e^{-x^2 w^2}}{w} dw$$

$$= \frac{\operatorname{erf}(xL)}{L} - \frac{\operatorname{erf}(xU)}{U} + \frac{2x}{\sqrt{\pi}} \left[\int_L^\infty \frac{e^{-x^2 w^2}}{w} dw - \int_U^\infty \frac{e^{-x^2 w^2}}{w} dw \right]$$

$$P(x,L,U) = \frac{\operatorname{erf}(xL)}{L} - \frac{\operatorname{erf}(xU)}{U} + \frac{x}{\sqrt{\pi}} \left[E_1(x^2 L^2) - E_1(x^2 U^2) \right]$$

Procedure:

Case I: $\quad T \le W_m, \quad I = \int_T^{W_m} \frac{\mathrm{erf}(aw)\mathrm{erf}(bw)}{w^2}\,dw + P(x, W_m, W_M) + \frac{1}{W_M}$

Case II: $\quad W_m < T \le W_M, \quad I = P(x, T, W_M) + \frac{1}{W_M}$

Case III: $\quad W_M < T < \infty, \quad I = \frac{1}{T}$

where $\quad W_m = \min\left(\frac{6}{a}, \frac{6}{b}\right), \quad W_M = \max\left(\frac{6}{a}, \frac{6}{b}\right)$

$x = \min(a,b)$

and

$$\frac{1}{W_M} + P(x, L, W_M) = \frac{\mathrm{erf}(xL)}{L} + \frac{\mathrm{erf}\,c(xW_M)}{W_M} + \frac{x}{\sqrt{\pi}}\left[E_1(x^2L^2) - E_1(x^2W_M{}^2)\right]$$

Folder 4

Evaluation of a Convolution Integral

$$\mathbf{I} = \int_0^t \mathbf{K(t-\tau)\,J(\tau)\,d\tau}$$

$$\mathbf{K(t)} = \frac{\mathbf{A}}{\sqrt{\mathbf{t}}}\mathbf{e}^{-c_2^2/t}, \quad \mathbf{J(t)} = \mathbf{c_1}\left[\frac{\mathbf{2}}{\sqrt{\pi \mathbf{t}}} - 2c_1 e^{c_1^2 t}\,\mathbf{erfc}(c_1\sqrt{\mathbf{t}})\right]$$

$$\mathbf{A} = \frac{1}{2\sqrt{\pi\alpha}}, \quad \mathbf{c_1} = \frac{\mathbf{h}\sqrt{\alpha}}{\mathbf{k}}, \quad \mathbf{c_2} = \frac{\mathbf{z}}{2\sqrt{\alpha}}$$

Donald E. Amos, April 2000

Summary

I(t) can be evaluated by means of the Laplace transform. It is however an instructive exercise to proceed by direct manipulation of the convolution with integrals from mathematical handbooks.

Derivation

Represent $K(t)$ by a Fourier Transform:

[2, p. 15 (11)] $$\int_0^\infty e^{-ax^2}\cos xy\,dx = \frac{\sqrt{\pi}}{2}\frac{1}{\sqrt{a}}e^{-y^2/(4a)}$$

or $$\int_{-\infty}^{+\infty} e^{-ax^2+ixy}\,dx = \frac{\sqrt{\pi}}{\sqrt{a}}e^{-y^2/(4a)}$$

Then $a = t,\ y = 2c_2$ *and* $K(t) = \frac{A}{\sqrt{\pi}}\int_{-\infty}^{+\infty} e^{-tx^2+2c_2 ix}\,dx.$

Represent $J(t)$ by a Laplace Transform:

[3, p. 238 (5) / 2, p. 146 (22)] $$\int_0^\infty e^{-pw}we^{-w^2/(4a)}\,dw = 2a - 2p\sqrt{\pi}a^{3/2}e^{ap^2}\,\mathrm{erfc}(p\sqrt{a})$$

Then $p = 2c_1$, $a = t/4$, $w = y$

$$\int_0^\infty e^{-2c_1 y}ye^{-y^2/t}\,dy = \frac{\sqrt{\pi}}{4}t^{3/2}\left[\frac{2}{\sqrt{\pi t}} - 2c_1 e^{c_1^2 t}\,\mathrm{erfc}(c_1\sqrt{t})\right]$$

and

$$J(t) = \frac{4c_1}{\sqrt{\pi}}\frac{1}{t^{3/2}}\int_0^\infty ye^{-2c_1 y - y^2/t}\,dy$$

Now,

$$I = \frac{4Ac_1}{\pi}\int_0^t \frac{1}{\tau^{3/2}}\int_{-\infty}^{+\infty} e^{-(t-\tau)x^2+2c_2ix}dx\int_0^\infty ye^{-2c_1y-y^2/\tau}dyd\tau$$

$$= \frac{4Ac_1}{\pi}\int_{-\infty}^{+\infty} e^{-tx^2+2c_2ix}dx\int_0^\infty ye^{-2c_1y}dy\int_0^t \frac{e^{-y^2/\tau+\tau x^2}}{\tau^{3/2}}d\tau$$

Let $\tau = \frac{1}{\nu^2}$, $\quad d\tau = \frac{-2}{\nu^3}d\nu$: $\int_0^t \frac{e^{-y^2/\tau+x^2\tau}}{\tau^{3/2}}d\tau = 2\int_{1/\sqrt{t}}^\infty e^{-y^2v^2+x^2/v^2}dv$

[1, p. 304 (7.4.34)] $\quad \int e^{-a^2v^2+b^2/v^2}dv = \frac{-\sqrt{\pi}}{4a}e^{-a^2v^2+b^2/v^2}\left[w\left(\frac{b}{v}+iav\right)+w\left(\frac{-b}{v}+iav\right)\right]$

[1, p. 297 (7.1.3)] $\quad$ where $w(z) = e^{-z^2}$ erfc($-iz$)

Let $a = y$, $b = x$, then

$$\int_0^t \frac{e^{-y^2/\tau+x^2\tau}}{\tau^{3/2}}d\tau = \frac{-\pi}{2y}\left\{e^{-y^2v^2+x^2/v^2}\left[w\left(\frac{x}{v}+iyv\right)+w\left(\frac{-x}{v}+iyv\right)\right]\right\}_{v=1/\sqrt{t}}^{\infty}$$

At ∞, $w(z) \to w(\pm 0+iyv) \sim e^{y^2v^2}$ erfc(yv) $\sim \frac{1}{\sqrt{\pi}yv} \to 0$

$$= \frac{\sqrt{\pi}}{2y}\left\{e^{-y^2/t+tx^2}\left[w\left(x\sqrt{t}+\frac{iy}{\sqrt{t}}\right)+w\left(-x\sqrt{t}+\frac{iy}{\sqrt{t}}\right)\right]\right\}$$

Now

$$w(a+ib) = e^{-(a^2-b^2+2abi)}\operatorname{erfc}(b-ia), \quad w(-a+ib) = e^{-(a^2-b^2-2abi)}\operatorname{erfc}(b+ia)$$

and $w(a+ib)$ and $w(-a+ib)$ are complex conjugates. Then

$$\int_0^t \frac{e^{-y^2/\tau+x^2\tau}}{\tau^{3/2}}d\tau = \frac{\sqrt{\pi}}{2y}\left\{e^{-y^2/t+tx^2}\cdot 2\operatorname{Re}\left[w\left(x\sqrt{t}+\frac{iy}{\sqrt{t}}\right)\right]\right\}$$

Now, [1, p. 302 (7.4.13)] $\quad \operatorname{Re}[w(a+bi)] = \frac{1}{\pi}\int_{-\infty}^{+\infty}\frac{be^{-w^2}dw}{(a-w)^2+b^2}$

Let $a = x\sqrt{t}$, $b = y/\sqrt{t}$

$$\int_0^t \frac{e^{-y^2/\tau + x^2\tau}}{\tau^{3/2}} d\tau = \frac{1}{y\sqrt{\pi}} \frac{e^{-y^2/t + tx^2}}{t} \int_{-\infty}^{+\infty} \frac{\left(y/\sqrt{t}\right) e^{-w^2} dw}{\left(x - w/\sqrt{t}\right)^2 + y^2/t^2}$$

Then

$$I = \frac{4Ac_1}{(\pi t)^{3/2}} \int_0^\infty y e^{-2c_1 y - y^2/t} dy \int_{-\infty}^{+\infty} e^{-w^2} dw \int_{-\infty}^{+\infty} \frac{e^{2ic_2 x} dx}{\left(x - w/\sqrt{t}\right)^2 + y^2/t^2}$$

Let $x - \frac{w}{\sqrt{t}} = u$

$$I = \frac{4Ac_1}{(\pi t)^{3/2}} \int_0^\infty y e^{-2c_1 y - y^2/t} dy \int_{-\infty}^{+\infty} e^{-w^2 + 2c_2 wi/\sqrt{t}} dw \int_{-\infty}^{+\infty} \frac{e^{2c_2 ui}}{u^2 + y^2/t^2} du$$

Notice that

$$\int_{-\infty}^{+\infty} \frac{e^{aiz}}{z^2 + b^2} dz = (2\pi i \cdot \text{Residue @ } z = ib) = \frac{2\pi i}{2ib} e^{-ab} = \frac{\pi}{b} e^{-ab}, \quad a > 0, \; b > 0.$$

$$I = \frac{4Ac_1}{(\pi t)^{3/2}} \int_0^\infty y e^{-2c_1 y - y^2/t} \int_{-\infty}^{+\infty} e^{-w^2 + 2c_2 iw/\sqrt{t}} \cdot \frac{\pi e^{-2c_2 y/t}}{(y/t)} dw dy$$

$$= \frac{4Ac_1}{\sqrt{\pi t}} \int_0^\infty e^{-(2c_1 + 2c_2/t)y} e^{-y^2/t} dy \cdot \int_{-\infty}^{+\infty} e^{-w^2 + 2c_2 wi/\sqrt{t}} dw$$

[2, p. 146 (21), $b = 0$] $$\int_0^\infty e^{-py} e^{-y^2/4a} dy = \sqrt{\pi a} e^{ap^2} \operatorname{erfc}(p\sqrt{a})$$

and

[2, p. 15 (11)] $$\int_{-\infty}^{+\infty} e^{-aw^2 + iyw} dw = \frac{\sqrt{\pi}}{\sqrt{a}} e^{-y^2/(4a)}$$

Let $p = 2c_1 + 2c_2/t$, $a = t/4$

$$\int_0^\infty e^{-py - y^2/t} dy = \frac{\sqrt{\pi t}}{2} e^{t\left(c_1^2 + 2c_1 c_2/t + c_2^2/t^2\right)} \operatorname{erfc}\left(c_1\sqrt{t} + \frac{c_2}{\sqrt{t}}\right)$$

Let $y = 2c_2/\sqrt{t}$, $a = 1$

$$\int_{-\infty}^{\infty} e^{-w^2 + 2c_2 wi/\sqrt{t}} dw \;=\; \sqrt{\pi} e^{-c_2^2/t}$$

Then

$$I \;=\; 2Ac_1\sqrt{\pi} e^{c_1^2 t + 2c_1 c_2} \operatorname{erfc}\left(c_1\sqrt{t} + \frac{c_2}{\sqrt{t}} \right)$$

$$c_1^2 \;=\; \frac{h^2\alpha}{k^2}, \quad 2c_1c_2 \;=\; 2\frac{h\sqrt{\alpha}}{k}\cdot\frac{z}{2\sqrt{\alpha}} \;=\; \frac{hz}{k}$$

$$c_1^2 t + 2c_1c_2 \;=\; \frac{h^2\alpha t}{k^2} + \frac{hz}{k}, \quad 2Ac_1\sqrt{\pi} \;=\; 2\cdot\frac{1}{2\sqrt{\pi\alpha}}\cdot\frac{h\sqrt{\alpha}}{k}\cdot\sqrt{\pi} \;=\; \frac{h}{k}$$

$$I \;=\; \frac{h}{k} e^{\left(h^2\alpha t/k^2 + hz/k\right)} \operatorname{erfc}\left(\frac{h\sqrt{\alpha t}}{k} + \frac{z}{2\sqrt{\alpha t}} \right)$$

QED.

References

[1] Abramowitz , M. and Stegun, I. A.
Handbook of Mathematical Functions
National Bureau of Standards, AMS 55,
US department of Commerce, 1964

[2] Erdelyi, A. Magnus, W. Oberhettinger, F. Tricomi, F. G.
Tables of Integral Transforms, Vol. 1
The Bateman Project
McGraw-Hill, 1954

[3] Gradshteyn, I. S. and Ryzhik, I. M.
Tables *of Integrals, Series and Products*
Acadenic Press, 1965

Folder 5

$$\mathbf{I(a,b,x) = \int_x^\infty e^{-a^2w^2}\,erfc(bw)dw, \quad J(a,b,x) = \int_x^\infty e^{-a^2w^2}\,erf(bw)dw}$$

$$\mathbf{U(a,b,x) = \int_0^x e^{-a^2w^2}\,erfc(bw)dw, \quad V(a,b,x) = \int_0^x e^{-a^2w^2}\,erf(bw)dw}$$

$$\mathbf{a>0, \quad b>0, \quad x>0}$$

Donald E. Amos, July 2000

Summary

Reference: See Θ and H Functions, Chapter 6, Beck et al.

A series expansion is developed for

$$I(a,b,x) = \frac{1}{2\sqrt{\pi}\sqrt{a^2+b^2}} G(a,b,x), \quad G(a,b,x) = \sum_{k=0}^{\infty} \frac{(1/2)_k}{k!}\left(\frac{a^2}{a^2+b^2}\right)^k E_{k+3/2}(d^2x^2) \quad a \le b$$

$$d^2 = a^2 + b^2$$

which converges well for $a \le b$. A companion relation

$$I(a,b,x) = \frac{\sqrt{\pi}}{2a}\operatorname{erfc}(ax)\operatorname{erfc}(bx) - \frac{b}{a} I(b,a,x) \qquad a > b$$

is used for $a > b$ since $I(b,a,x)$ contains the factor $[b^2/(a^2+b^2)]^k$ and the convergence is good for $a > b$. $G(a,b,x)$ is further broken down into

$$\frac{G(a,b,x)}{\sqrt{a^2+b^2}} = 2\frac{e^{-d^2x^2}}{a}\tan^{-1}\frac{a}{b} - dx^2 S(a,b,x),$$

$$S(a,b,x) = \sum_{k=0}^{\infty} \frac{(1/2)_k}{k!(k+1/2)}\left(\frac{a^2}{a^2+b^2}\right)^k E_{k+1/2}(d^2x^2)$$

to break out the dominant behavior when x is small. These formulae are manipulated to obtain forms which are suitable for computation. Auxiliary results for Laplace transforms are obtained:

$$\int_0^\infty e^{-pt}\frac{\operatorname{erfc}(\sqrt{\alpha t})}{\sqrt{t}}dt, \qquad \int_0^\infty e^{-pt}\frac{\operatorname{erf}(\sqrt{\alpha t})}{\sqrt{t}}dt$$

$$\int_0^\infty e^{-pt}\sqrt{t}\operatorname{erfc}(\sqrt{\alpha t})dt, \qquad \int_0^\infty e^{-pt}\sqrt{t}\operatorname{erf}(\sqrt{\alpha t})dt$$

as well as series for $i^n\operatorname{erfc}(z)$, $n = -1,0,1$.

Folder 5a

Series Expansion and computation of $\int_x^\infty e^{-a^2w^2}\,\mathrm{erfc}(bw)dw$, $a > 0,\ b > 0$

The integral $\int_x^\infty e^{-a^2w^2}\,\mathrm{erfc}(bw)dw$ and its companion $\int_x^\infty e^{-a^2w^2}\,\mathrm{erf}(bw)dw$ occur in Chapter 6 of Beck et al. for the solution of a heat conduction problem. Since

$$\int_x^\infty e^{-a^2w^2}\,\mathrm{erf}(bw)dw = \frac{\sqrt{\pi}}{2a}\mathrm{erfc}(ax) - \int_x^\infty e^{-a^2w^2}\,\mathrm{erfc}(bw)dw$$

we concentrate on the coerror function for this section.

Let, $I = \int_x^\infty e^{-a^2v^2}\,\mathrm{erfc}(bv)dv$ and make the substitution

$$\mathrm{erfc}(bx) = \frac{2}{\sqrt{\pi}}\int_{bx}^\infty e^{-v^2}dv = \frac{2bx}{\sqrt{\pi}}\int_1^\infty e^{-b^2x^2w^2}dw$$

Then,

$$I = \frac{2b}{\sqrt{\pi}}\int_1^\infty dw\int_x^\infty e^{-(b^2w^2+a^2)v^2}vdv$$

$$= \frac{b}{\sqrt{\pi}}\int_1^\infty \frac{e^{-(b^2w^2+a^2)x^2}}{b^2w^2+a^2}dw$$

Let $(b^2w^2+a^2)x^2 = s$, $2b^2wx^2dw = ds$, $w = \frac{1}{xb}\sqrt{s-a^2x^2}$

$$I = \frac{x}{2\sqrt{\pi}}\int_{(a^2+b^2)x^2}^\infty e^{-s}\Big/\left(s^{3/2}\sqrt{1-\frac{a^2x^2}{s}}\right)ds$$

$$\left(1-\frac{a^2x^2}{s}\right)^{-1/2} = \sum_{k=0}^\infty \frac{(1/2)_k}{k!}\frac{(a^2x^2)^k}{s^k}$$

$$I = \frac{x}{2\sqrt{\pi}}\sum_{k=0}^\infty \frac{(1/2)_k}{k!}(a^2x^2)^k\int_{(a^2+b^2)x^2}^\infty \frac{e^{-s}}{s^{k+3/2}}ds$$

$$I = \frac{1}{2\sqrt{\pi}} \sum_{k=0}^{\infty} \frac{(1/2)_k}{k!} \left(\frac{a^2}{a^2+b^2} \right)^k E_{k+3/2}[(a^2+b^2)x^2] \frac{1}{\sqrt{a^2+b^2}}$$

and finally,

$$I(a,b,x) = \frac{1}{2\sqrt{\pi}\sqrt{a^2+b^2}} \sum_{k=0}^{\infty} \frac{(1/2)_k}{k!} \left(\frac{a^2}{a^2+b^2} \right)^k E_{k+3/2}[(a^2+b^2)x^2]$$

Now, this series converges very well for $\frac{a^2}{a^2+b^2} \le \frac{1}{2}$ or $a \le b$. Now we integrate by parts to obtain a companion series which converges well for $a > b$:

$$u = \operatorname{erfc}(bw) \qquad dv = e^{-a^2w^2}\,dw$$

$$u = \frac{-2b}{\sqrt{\pi}} e^{-b^2w^2} \qquad v = -\frac{\sqrt{\pi}}{2a} \operatorname{erfc}(aw)$$

$$I(a,b,x) = -\frac{\sqrt{\pi}}{2a} \operatorname{erfc}(ax)\operatorname{erfc}(bx) \Bigg|_x^{\infty} - \frac{b}{a}\int_x^{\infty} e^{-b^2w^2} \operatorname{erfc}(aw)\,dw$$

$$= \frac{\sqrt{\pi}}{2a} \operatorname{erfc}(ax)\operatorname{erfc}(bx) - \frac{b}{a} I(b,a,x)$$

Thus, $I(b,a,X)$ has the factor $\left(\frac{b^2}{a^2+b^2} \right)^k$ in the series and this converges well for $a > b$, with the worst case for $a = b$ where the convergence is like $O(2^{-k}k^{-3/2})$ ultimately since $E_\nu(x) \sim e^{-x}/(x+\nu)$ for large $\nu = k+3/2$. In practice, ν has to be large compared to x for this to apply, so the convergence is more like $O(2^{-k}k^{-1/2})$ for most cases and $E_\nu(x)$ behaves like a scale factor.

SPECIAL CASES

A Laplace Transform:

Set x = 0 in I and we get

$$\int_0^{\infty} e^{-a^2x^2} \operatorname{erfc}(bx)\,dx = \frac{1}{2\sqrt{\pi}\sqrt{a^2+b^2}} \sum_{k=0}^{\infty} \frac{(1/2)_k}{k!} \left(\frac{a^2}{a^2+b^2} \right)^k E_{k+3/2}(0).$$

Now, $E_{k+3/2}(0) = \frac{1}{k+1/2}$ and with $C_k = (1/2)_k/k!$ we have

$$(1-z)^{-1/2} = \sum_{k=0}^{\infty} C_k z^k, \qquad z^{-1/2}(1-z)^{-1/2} = \sum_{k=0}^{\infty} C_k z^{k-1/2}$$

and $\displaystyle \frac{1}{\sqrt{z}}\int_0^z \frac{dt}{\sqrt{t}\sqrt{1-t}} = \sum_{k=0}^{\infty} C_k \frac{z^k}{k+1/2}$. Then

$$I = \frac{1}{2\sqrt{\pi}}\frac{1}{\sqrt{a^2+b^2}}\frac{1}{\sqrt{z}}\int_0^z \frac{dt}{\sqrt{t(1-t)}}, \qquad z = \frac{a^2}{a^2+b^2}$$

$$= \frac{1}{2a\sqrt{\pi}}\int_0^z \frac{dt}{\sqrt{t(1-t)}} = \frac{1}{2a\sqrt{\pi}} \cdot 2\sin^{-1}\frac{a}{\sqrt{a^2+b^2}}$$

$$= \frac{1}{a\sqrt{\pi}}\tan^{-1}\left(\frac{a}{b}\right)$$

A change of variables $x^2 = t$, gives

$$\int_0^{\infty} e^{-a^2 t}\frac{\operatorname{erfc}(b\sqrt{t})}{\sqrt{t}}dt = \frac{2}{a\sqrt{\pi}}\tan^{-1}\left(\frac{a}{b}\right)$$

Let $a = \sqrt{p}$ and $b = \sqrt{\alpha}$, then

$$\int_0^{\infty} e^{-pt}\frac{\operatorname{erfc}(\sqrt{\alpha t})}{\sqrt{t}}dt = \frac{2}{\sqrt{\pi p}}\tan^{-1}\sqrt{\frac{p}{\alpha}}$$

also $\displaystyle \int_0^{\infty} e^{-pt}\sqrt{t}\operatorname{erfc}(\sqrt{\alpha\,\mathrm{t}})dt = \frac{-2}{\sqrt{\pi}}\frac{d}{dp}\left(\frac{1}{\sqrt{p}}\tan^{-1}\sqrt{\frac{p}{\alpha}}\right) = \frac{1}{\sqrt{\pi}}\frac{1}{p}\left[\frac{1}{\sqrt{p}}\tan^{-1}\left(\sqrt{\frac{p}{\alpha}}\right) - \frac{\sqrt{\alpha}}{p+\alpha}\right]$

Representation for i^nerfc(z), n = -1,0,1

Take $b = 0$ in $I(a,b,x)$: $(C_k = (1/2)_k/k!)$

$$\frac{\sqrt{\pi}}{2a}\operatorname{erfc}(ax) = \frac{1}{2\sqrt{\pi}}\frac{1}{a}\sum_{k=0}^{\infty} C_k E_{k+3/2}(a^2x^2)$$

or $\displaystyle \operatorname{erfc}(z) = \frac{1}{\pi}\sum_{k=0}^{\infty} C_k E_{k+3/2}(z^2)$

Differentiate wrt z to get

$$\frac{2}{\sqrt{\pi}} e^{-z^2} = \frac{1}{\pi}\sum_{k=0}^{\infty} C_k E_{k+1/2}(z^2)\cdot(2z) \qquad \text{or} \qquad \frac{e^{-z^2}}{z} = \frac{1}{\sqrt{\pi}}\sum_{k=0}^{\infty} C_k E_{k+1/2}(z^2)\,.$$

Now, integrate the expansion for erfc(z) to get

$$i^1\,\mathrm{erfc}(z) = \frac{1}{\pi}\sum_{k=0}^{\infty} C_k \int_z^{\infty} E_{k+3/2}(x^2)dx$$

$$= \frac{1}{\pi}\sum_{k=0}^{\infty} C_k \int_{z^2}^{\infty} \frac{E_{k+3/2}(u)}{2\sqrt{u}}du$$

$$= \frac{z}{2\pi}\sum_{k=0}^{\infty} C_k \left[\frac{E_{1/2}(z^2) - E_{k+3/2}(z^2)}{k+1}\right]$$

$$= \frac{1}{\sqrt{\pi}}\mathrm{erfc}(z) - \frac{z}{2\pi}\sum_{k=0}^{\infty} \frac{(1/2)_k}{k!(k+1)} E_{k+3/2}(z^2)$$

since $E_{1/2}(z^2) = \dfrac{\sqrt{\pi}}{z}\mathrm{erfc}(z)$ and $\displaystyle\sum_{k=0}^{\infty} \frac{C_k}{k+1} = \int_0^1 \frac{dz}{\sqrt{1-z}} = 2$ using the series above for $(1\text{-}z)^{-1/2}$.

<u>Computation of</u> $I(a,b,x) = \int_x^{\infty} e^{-a^2w^2}\,\mathrm{erfc}(bw)dw\,, \quad a > 0,\ b \geq 0$

In the previous sections, we derived the relations

$$I(a,b,x) = \begin{cases} \dfrac{1}{2\sqrt{\pi}}\dfrac{1}{\sqrt{a^2+b^2}}G(a,b,x) & a \leq b \\ \dfrac{\sqrt{\pi}}{2a}\mathrm{erfc}(ax)\,\mathrm{erfc}(bx) - \dfrac{b}{2a\sqrt{\pi}}\dfrac{1}{\sqrt{a^2+b^2}}G(b,a,x) & a > b \end{cases}$$

where $G(a,b,x) = \displaystyle\sum_{k=0}^{\infty} C_k \left(\frac{a^2}{a^2+b^2}\right)^k E_{k+3/2}(d^2x^2)$ $\qquad d^2 = a^2 + b^2$, $\quad C_k = (1/2)_k / k!$

The following manipulations of G result in two benefits which are valuable later on. The first benefit is an acceleration of the convergence of the series and the second is a separation of dominant behavior for extremes in a,b and x.

We apply the relation $(k+1/2)E_{k+3/2}(z) = e^{-z}\text{-}zE_{k+1/2}(z)$ to obtain

$$G(a,b,x) = e^{-d^2x^2}\sum_{k=0}^{\infty} \frac{C_k}{k+1/2}\left(\frac{a^2}{a^2+b^2}\right)^k - d^2x^2\sum_{k=0}^{\infty} \frac{C_k}{k+1/2}\left(\frac{a^2}{a^2+b^2}\right)^k E_{k+1/2}(d^2x^2)$$

Now, from Folder 6, we have

$$\sum_{k=0}^{\infty} \frac{C_k z^{k+x+1/2}}{x+k+1/2} = \int_0^z t^{x-1/2}(1-t)^{-1/2}\,dt\,.$$

With $x = 0$, we get $\sum_{k=0}^{\infty} \frac{C_k}{k+1/2} z^k = \frac{2}{\sqrt{z}}\sin^{-1}\sqrt{z}$ and with $z = a^2/(a^2+b^2)$ we get

$$\frac{G(a,b,x)}{\sqrt{a^2+b^2}} = \frac{2e^{-d^2x^2}}{a}\tan^{-1}\frac{a}{b} - dx^2 S(a,b,x)\,, \qquad d^2 = a^2+b^2$$

$$S(a,b,x) = \sum_{k=0}^{\infty} S_k\left(\frac{a^2}{a^2+b^2}\right)^k E_{k+1/2}(d^2x^2)\,, \qquad S_k = C_k/(k+1/2)$$

In the special cases, we showed that for $x = 0$,

$$I(a,b,0) = \frac{1}{a\sqrt{\pi}}\tan^{-1}\frac{a}{b}$$

so that with the coefficient of G/d we have the dominant behavior for small and large x. However, for $a > b$, we need an expression for small x. Then,

$$\frac{\sqrt{\pi}}{2a}\operatorname{erfc}(ax)\operatorname{erfc}(bx) - \frac{e^{-d^2x^2}}{a\sqrt{\pi}}\tan^{-1}\frac{b}{a} =$$

$$\frac{1}{a\sqrt{\pi}}\left\{\tan^{-1}\frac{a}{b} + \phi(d^2x^2)\tan^{-1}\frac{b}{a}\right\} - \frac{\sqrt{\pi}}{2a}\left[\operatorname{erf}(ax) + \operatorname{erf}(bx) - \operatorname{erf}(ax)\operatorname{erf}(bx)\right]$$

using $\tan^{-1}z+\tan^{-1}(1/z) = \pi/2$ where $\phi(z) \equiv 2e^{-z/2}\sinh(z/2) = 1-e^{-z}$.

In this form, $\phi(z)$ does not lose significant digits as $z \to 0$. Then,

$$
I(a,b,x) = \begin{cases} \dfrac{1}{2\sqrt{\pi}} \dfrac{G(a,b,x)}{\sqrt{a^2+b^2}} & a \le b \\[2ex] \dfrac{1}{a\sqrt{\pi}}\left[\tan^{-1}\dfrac{a}{b} + \phi(d^2x^2)\tan^{-1}\dfrac{b}{a}\right] - \dfrac{\sqrt{\pi}}{2a}[\mathrm{erf}(ax) + & \\ \quad \mathrm{erf}(bx) - \mathrm{erf}(ax)\,\mathrm{erf}(bx)] + \dfrac{bdx^2}{2a\sqrt{\pi}} S(b,a,x) \qquad d^2x^2 \le 3 & \\ & a > b \\ \dfrac{\sqrt{\pi}}{2a}\mathrm{erfc}(ax)\,\mathrm{erfc}(bx) - \dfrac{b}{2a\sqrt{\pi}} \dfrac{G(b,a,x)}{\sqrt{a^2+b^2}} \qquad d^2x^2 > 3 & \end{cases}
$$

where

$$
G(a,b,x) = \sum_{k=0}^{\infty} C_k \left(\frac{a^2}{a^2+b^2} \right)^k E_{k+3/2}(d^2x^2)
$$

$$
S(a,b,x) = \sum_{k=0}^{\infty} S_k \left(\frac{a^2}{a^2+b^2} \right)^k E_{k+1/2}(d^2x^2)
$$

$$
C_k = \left(\frac{1}{2} \right)_k / k!, \qquad S_k = C_k / (k+1/2), \qquad d^2 = a^2 + b^2
$$

Folder 5b

Computation of $J(a,b,x) = \int_x^\infty e^{-a^2w^2}\,\mathrm{erf}(bw)dw$, $\qquad a > 0,\ b \geq 0$

Notice that $J(a,b,x) = I(a,0,x) - I(a,b,x) = \dfrac{\sqrt{\pi}}{2a}\mathrm{erfc}(ax) - I(a,b,x)$

Then, with $d^2 = a^2+b^2$,

$$J(a,b,x) = \begin{cases} \dfrac{\sqrt{\pi}}{2a}\mathrm{erfc}(ax) - \dfrac{e^{-d^2x^2}}{a\sqrt{\pi}}\tan^{-1}\dfrac{a}{b} + \dfrac{dx^2}{2\sqrt{\pi}}S(a,b,x) & 0 < a \leq b \\ \\ \dfrac{\sqrt{\pi}}{2a}\mathrm{erfc}(ax)\,\mathrm{erf}(bx) + \dfrac{e^{-d^2x^2}}{a\sqrt{\pi}}\tan^{-1}\dfrac{b}{a} - \dfrac{bdx^2}{2a\sqrt{\pi}}S(b,a,x) & a > b \geq 0 \end{cases}$$

The expression for $a \leq b$ does not show the dominant behavior as $x \to 0$ since $\dfrac{\sqrt{\pi}}{2a} - \dfrac{1}{a\sqrt{\pi}}\tan^{-1}\dfrac{a}{b} = \dfrac{1}{a\sqrt{\pi}}\tan^{-1}\dfrac{b}{a}$. Therefore we use erfc($ax$) = 1-erf($ax$) and get ($d^2 = a^2+b^2$)

$$J(a,b,x) = \begin{cases} \dfrac{\phi(d^2x^2)}{a\sqrt{\pi}}\tan^{-1}\dfrac{a}{b} + \dfrac{1}{a\sqrt{\pi}}\tan^{-1}\dfrac{b}{a} - \dfrac{\sqrt{\pi}}{2a}\mathrm{erf}(ax) & \\ \qquad\qquad + \dfrac{dx^2}{2\sqrt{\pi}}S(a,b,x) & d^2x^2 \leq 3 \qquad 0 < a \leq b \\ \\ \dfrac{\sqrt{\pi}}{2a}\mathrm{erfc}(ax) - \dfrac{1}{2\sqrt{\pi}}\dfrac{G(a,b,x)}{\sqrt{a^2+b^2}} & d^2x^2 > 3 \\ \\ \dfrac{\sqrt{\pi}}{2a}\mathrm{erfc}(ax)\,\mathrm{erf}(bx) + \dfrac{b}{2a\sqrt{\pi}}\dfrac{G(b,a,x)}{\sqrt{a^2+b^2}} & a > b \geq 0 \end{cases}$$

where $\phi(z) = 2e^{-z/2}\sinh(z/2) = 1-e^{-z}$ and $d^2 = a^2+b^2$.

From this we get the result

$$J(a,b,0) = \int_0^\infty e^{-a^2x^2}\,\mathrm{erf}(bx)dx = \frac{1}{a\sqrt{\pi}}\tan^{-1}\frac{b}{a}$$

or the Laplace Transform

$$\int_0^\infty e^{-pt}\,\frac{\mathrm{erf}(\sqrt{\alpha t})}{\sqrt{t}}\,dt = \frac{2}{\sqrt{\pi p}}\tan^{-1}\sqrt{\frac{\alpha}{p}}$$

We also have

$$\int_0^\infty e^{-pt}\sqrt{t}\,\mathrm{erf}(\sqrt{\alpha t})dt = \frac{1}{p\sqrt{\pi}}\left[\frac{1}{\sqrt{p}}\tan^{-1}\sqrt{\frac{\alpha}{p}} + \frac{\sqrt{\alpha}}{p+\alpha}\right]$$

by differentiation wrt p. A check computation:

Since $\int_0^\infty e^{-pt}\sqrt{t}\,\mathrm{erfc}(\sqrt{\alpha t})dt = \frac{1}{p\sqrt{\pi}}\left[\frac{1}{\sqrt{p}}\tan^{-1}\sqrt{\frac{p}{\alpha}} - \frac{\sqrt{\alpha}}{p+\alpha}\right]$

we add to get

$$\int_0^\infty e^{-pt}\sqrt{t}dt = \frac{1}{p\sqrt{\pi}}\cdot\frac{1}{\sqrt{p}}\left[\tan^{-1}\sqrt{\frac{p}{\alpha}} + \tan^{-1}\sqrt{\frac{\alpha}{p}}\right] = \frac{\sqrt{\pi}}{2p^{3/2}} = \frac{\Gamma(3/2)}{p^{3/2}} = \frac{1/2\Gamma(1/2)}{p^{3/2}} = \frac{\sqrt{\pi}}{2p^{3/2}}$$

since $\mathrm{erf}(z)+\mathrm{erfc}(z) = 1$ and $\tan^{-1}z+\tan^{-1}(1/z) = \pi/2$.

$J(a,b,x)$ can also be expressed by

$$J(a,b,x) = \int_0^\infty e^{-a^2w^2}\,\mathrm{erf}(bw)dw - \int_0^x e^{-a^2w^2}\,\mathrm{erf}(bw)dw = \frac{1}{a\sqrt{\pi}}\tan^{-1}\frac{b}{a} - V(a,b,x)$$

where V is treated separately. It turns out that for small a,b and x, the power series development of V outperforms other formulae in terms of accuracy. Therefore, for $ax\le 1$ and $bx\le 1$, $J(a,b,x)$ was computed from the relation above with the power series. The formulae in terms of S and G were applied for $ax>1$ or $bx>1$.

Folder 5c

Computation of U(a,b,x) = $\int_0^x e^{-a^2w^2}$ erfc(bw)dw , $a \geq 0,\ b \geq 0$

Notice $U(a,b,x) = I(a,b,0) - I(a,b,x) = \frac{1}{a\sqrt{\pi}}\tan^{-1}\frac{a}{b} - I(a,b,x)$

then,

$$U(a,b,x) = \begin{cases} \dfrac{(1-e^{-d^2x^2})}{a\sqrt{\pi}}\tan^{-1}\dfrac{a}{b} + \dfrac{dx^2}{2\sqrt{\pi}}S(a,b,x) & a \leq b \\ \\ \dfrac{1}{a\sqrt{\pi}}\tan^{-1}\dfrac{a}{b} - \dfrac{\sqrt{\pi}}{2a}\operatorname{erfc}(ax)\operatorname{erfc}(bx) + \dfrac{b}{2a\sqrt{\pi}}\dfrac{G(b,a,x)}{\sqrt{a^2+b^2}} & a > b \end{cases}$$

In order to prevent losses of significance when x is small, we compute $1-e^{-z} = 2e^{-z/2}\sinh(z/2) \equiv \phi(z)$. We also need to modify the expression for $a>b$ for small x because erfc(z)$\to$1 and the first two terms combine to $\left(\tan^{-1}\frac{a}{b} - \frac{\pi}{2}\right)/a\sqrt{\pi} = -\tan^{-1}\frac{b}{a}\cdot\frac{1}{a\sqrt{\pi}}$ which loses significant digits when b is small. We use the S form of G to get

$$\frac{1}{a\sqrt{\pi}}\left\{\tan^{-1}\frac{a}{b} - \frac{\pi}{2}\left[1-\operatorname{erf}(ax)-\operatorname{erf}(bx)+\operatorname{erf}(ax)\operatorname{erf}(bx)\right] + e^{-d^2x^2}\tan^{-1}\frac{b}{a}\right\} - \frac{bdx^2}{2a\sqrt{\pi}}S(b,a,x)$$

$$= -\frac{\phi(d^2x^2)}{a\sqrt{\pi}}\tan^{-1}\frac{b}{a} + \frac{\sqrt{\pi}}{2a}\left[\operatorname{erf}(ax)+\operatorname{erf}(bx)-\operatorname{erf}(ax)\operatorname{erf}(bx)\right] - \frac{bdx^2}{2a\sqrt{\pi}}S(b,a,x)$$

using $\tan^{-1}z + \tan^{-1}(1/z) = \pi/2$.

Then the final answer, which is borne out by numerical experiments using a quadrature on the original integral, is

$$U(a,b,x)=\begin{cases}\dfrac{\phi(d^2x^2)}{a\sqrt{\pi}}\tan^{-1}\dfrac{a}{b}+\dfrac{dx^2}{2\sqrt{\pi}}S(a,b,x) & d^2x^2\le 3 \quad a\le b\\[2ex] \dfrac{1}{a\sqrt{\pi}}\tan^{-1}\dfrac{a}{b}-\dfrac{1}{2\sqrt{\pi}}\dfrac{G(a,b,x)}{\sqrt{a^2+b^2}} & d^2x^2>3 \\[2ex] -\dfrac{\phi(d^2x^2)}{a\sqrt{\pi}}\tan^{-1}\dfrac{b}{a}+\dfrac{\sqrt{\pi}}{2a}\left[\operatorname{erf}(ax)+\operatorname{erf}(bx)-\operatorname{erf}(ax)\operatorname{erf}(bx)\right] & \\ \qquad -\dfrac{bdx^2}{2a\sqrt{\pi}}S(b,a,x) & d^2x^2\le 3 \quad a>b\\[2ex] \dfrac{1}{a\sqrt{\pi}}\tan^{-1}\dfrac{a}{b}-\dfrac{\sqrt{\pi}}{2a}\operatorname{erfc}(ax)\operatorname{erfc}(bx) & \\ \qquad +\dfrac{b}{2a\sqrt{\pi}}\dfrac{G(b,a,x)}{\sqrt{a^2+b^2}} & d^2x^2>3\end{cases}$$

where $\phi(z) = 2e^{-z/2}\sinh(z/2)$ and

$$S(a,b,x)=\sum_{k=0}^{\infty} S_k\left(\frac{a^2}{a^2+b^2}\right)^k E_{k+1/2}(d^2x^2)$$

$$G(a,b,x)=\sum_{k=0}^{\infty} C_k\left(\frac{a^2}{a^2+b^2}\right)^k E_{k+3/2}(d^2x^2)$$

$C_k = (1/2)_k/k!$, $\quad S_k = C_k/(k+1/2)$, $\quad d^2 = a^2+b^2$.

Folder 5d

Computation of $V(a,b,x)=\int_0^x e^{-a^2w^2}\,\mathrm{erf}(bw)\,dw$, $a>0$, $b>0$

Notice $V(a,b,x)=\dfrac{\sqrt{\pi}}{2a}\mathrm{erf}(ax)-U(a,b,x)$

Then

$$V(a,b,x)=\begin{cases}\dfrac{\sqrt{\pi}}{2a}\mathrm{erf}(ax)-\dfrac{\phi(d^2x^2)}{a\sqrt{\pi}}\tan^{-1}\dfrac{a}{b}-\dfrac{dx^2}{2\sqrt{\pi}}S(a,b,x) & d^2x^2\le 3 \\ & \qquad a\le b \\ \dfrac{\sqrt{\pi}}{2a}\mathrm{erf}(ax)-\dfrac{1}{a\sqrt{\pi}}\tan^{-1}\dfrac{a}{b}+\dfrac{1}{2\sqrt{\pi}}\dfrac{G(a,b,x)}{d} & d^2x^2>3 \\ \\ \dfrac{\phi(d^2x^2)}{a\sqrt{\pi}}\tan^{-1}\dfrac{b}{a}-\dfrac{\sqrt{\pi}}{2a}[\mathrm{erfc}(ax)\,\mathrm{erf}(bx)]+\dfrac{bdx^2}{2a\sqrt{\pi}}S(b,a,x) & d^2x^2\le 3 \\ & \qquad a>b \\ \dfrac{-1}{a\sqrt{\pi}}\tan^{-1}\dfrac{a}{b}+\dfrac{\sqrt{\pi}}{2a}[1-\mathrm{erfc}(ax)\,\mathrm{erf}(bx)]-\dfrac{b}{2a\sqrt{\pi}}\dfrac{G(b,a,x)}{d} & d^2x^2>3\end{cases}$$

Now we use $\tan^{-1}z+\tan^{-1}(1/z)=\pi/2$ and $\mathrm{erf}(z)=1-\mathrm{erfc}(z)$ to eliminate losses of significance in subtraction:

$$V(a,b,x)=\begin{cases}\dfrac{\sqrt{\pi}}{2a}\mathrm{erf}(ax)-\dfrac{\phi(d^2x^2)}{a\sqrt{\pi}}\tan^{-1}\dfrac{a}{b}-\dfrac{dx^2}{2\sqrt{\pi}}S(a,b,x) & d^2x^2\le 3 \\ & \qquad a\le b \\ \dfrac{1}{a\sqrt{\pi}}\tan^{-1}\dfrac{b}{a}-\dfrac{\sqrt{\pi}}{2a}\mathrm{erfc}(ax)+\dfrac{1}{2\sqrt{\pi}}\dfrac{G(a,b,x)}{d} & d^2x^2>3 \\ \\ \dfrac{\phi(d^2x^2)}{a\sqrt{\pi}}\tan^{-1}\dfrac{b}{a}-\dfrac{\sqrt{\pi}}{2a}\mathrm{erfc}(ax)\,\mathrm{erf}(bx)+\dfrac{bdx^2}{2a\sqrt{\pi}}S(b,a,x) & d^2x^2\le 3 \\ & \qquad a>b \\ \dfrac{1}{a\sqrt{\pi}}\tan^{-1}\dfrac{b}{a}-\dfrac{\sqrt{\pi}}{2a}\mathrm{erfc}(ax)\,\mathrm{erf}(bx)-\dfrac{b}{2a\sqrt{\pi}}\dfrac{G(b,a,x)}{d} & d^2x^2>3\end{cases}$$

Series development of V for small a,b,x. Numerical experiments on the power series compared with the formulae in terms of S and G showed the power series to be more accurate for $ax \le 1$ and $bx \le 1$. By writing the power series for $\exp(-a^2x^2)$ and $\mathrm{erf}(bx)$, multiplying and integrating, we get

$$V(a,b,x) = \frac{bx^2}{\sqrt{\pi}} \sum_{k=0}^{\infty} \frac{1}{k+1} P_k(a,b,x)$$

where

$$P_k(a,b,x) = \sum_{m=0}^{k} \frac{(-b^2x^2)^m}{(2m+1)m!} \frac{(-a^2x^2)^{k-m}}{(k-m)!}$$

and the powers of x have been distributed through P_k to use the variables ax and bx. The factor $(2m+1)m!$ comes from $\mathrm{erf}(bx)$ while $(k-m)!$ comes from $\exp(-a^2x^2)$.

In programming, the test for $ax \le 1$ and $bx \le 1$ was applied first and the formulae in terms of S and G were then applied if this test failed.

21 coefficients for $\exp(-a^2x^2)$ and $\mathrm{erf}(bx)$ were stored in data statements.

<u>Special Cases $a \to 0$</u>

$$V(0,b,x) = \int_0^x \mathrm{erf}(bx)dx = x\,\mathrm{erf}(bx) - \frac{1}{b\sqrt{\pi}}(1 - e^{-b^2x^2})$$

Take $a \le b$ case and let $a \to 0$, $d^2 = a^2 + b^2 \to b^2$

$$V(0,b,x) = -\frac{\phi(b^2x^2)}{b\sqrt{\pi}} - \frac{bx^2}{2\sqrt{\pi}} \cdot 2E_{1/2}(b^2x^2) + \lim_{a \to 0} \frac{\sqrt{\pi}}{2a}\mathrm{erf}(ax)$$

$$= -\frac{1}{b\sqrt{\pi}}(1 - e^{-b^2x^2}) - \frac{bx^2}{\sqrt{\pi}} \cdot \frac{\sqrt{\pi}\,\mathrm{erfc}(bx)}{bx} + x$$

$$= x - x\,\mathrm{erfc}(bx) - \frac{1}{b\sqrt{\pi}}(1 - e^{-b^2x^2})$$

$$= x\,\mathrm{erf}(bx) - \frac{1}{b\sqrt{\pi}}(1 - e^{-b^2x^2})$$

Also from second case for $a \le b$

$$
\begin{aligned}
V(0,b,x) &= x - \frac{1}{b\sqrt{\pi}} + \frac{1}{2\sqrt{\pi}} \frac{E_{3/2}(b^2x^2)}{b} \\
&= x - \frac{1}{b\sqrt{\pi}} + \frac{1}{2b\sqrt{\pi}} \cdot 2\sqrt{\pi}\, i\,\mathrm{erfc}(bx) \\
&= x - \frac{1}{b\sqrt{\pi}} + \frac{1}{b}\left[\frac{1}{\sqrt{\pi}} e^{-b^2x^2} - bx\,\mathrm{erfc}(bx)\right] \\
&= x - \frac{1}{b\sqrt{\pi}}(1 - e^{-b^2x^2}) - x(1 - \mathrm{erf}(bx)) \\
&= x\,\mathrm{erf}(bx) - \frac{1}{b\sqrt{\pi}}(1 - e^{-b^2x^2})
\end{aligned}
$$

Folder 5e

Asymptotics for I, J, U, V for Large x

We start with the asymptotics for I(a,b,x) by replacing the erfc(bw) integrand with its asymptotic expansion and relating all other functions to I(a,b,x). Thus, for $b \ge a \quad or \; a \le b$ and $x \to \infty$ we have

$$erfc(x) = \frac{e^{-x^2}}{x\sqrt{\pi}} \sum_{k=0}^{N} \frac{(-1)^k (1/2)_k}{x^{2k}} + R_N \quad , \qquad |R_N| \le \frac{e^{-x^2}}{x\sqrt{\pi}} \frac{(1/2)_{N+1}}{x^{2N+2}}$$

$$I(a,b,x) = \int_x^\infty e^{-a^2 w^2} erfc(bw)dw = \frac{1}{b\sqrt{\pi}} \sum_{k=0}^{N} \frac{(-1)^k (1/2)_k}{b^{2k}} \int_x^\infty \frac{e^{-(a^2+b^2)w^2}}{w^{2k+1}} dw + \int_x^\infty e^{-a^2 w^2} R_N(bw)dw \,, \quad \text{b>a}$$

and with the substitution $w = x\sqrt{v}$ we get

$$I(a,b,x) = \frac{1}{2b\sqrt{\pi}} \sum_{k=0}^{N} \frac{(-1)^k (1/2)_k}{(bx)^{2k}} E_{k+1}\left[x^2(a^2+b^2)\right] + W_N \;, \quad |W_N| \le \frac{1}{2b\sqrt{\pi}} \frac{(1/2)_{N+1}}{(bx)^{2N+2}} E_{N+2}\left[x^2(a^2+b^2)\right]$$

$$a \le b \,.$$

For $x \to \infty$ and a>b we use the reflexive relation

$$I(a,b,x) = \frac{\sqrt{\pi}}{2a} \operatorname{erfc}(ax)\operatorname{erfc}(bx) - \frac{b}{a} I(b,a,x) \,, \qquad \text{a>b}$$

since I(b,a,x) is the form just considered where the second parameter is larger than the first. Explicitly

$$I(a,b,x) = \frac{\sqrt{\pi}}{2a} \operatorname{erfc}(ax)\operatorname{erfc}(bx) - \frac{b}{2a^2\sqrt{\pi}} \sum_{k=0}^{N} \frac{(-1)^k (1/2)_k}{(ax)^{2k}} E_{k+1}\left[x^2(a^2+b^2)\right] + W_N \;, \qquad \text{a>b}$$

$$|W_N| \le \frac{b}{2a^2\sqrt{\pi}} \frac{(1/2)_{N+1}}{(ax)^{2N+2}} E_{N+2}\left[x^2(a^2+b^2)\right]$$

Now, we use relations

$$\mathrm{J}(\mathrm{a},\mathrm{b},x) = \int_x^\infty \mathrm{e}^{-\mathrm{a}^2 w^2} \operatorname{erf}(\mathrm{b}\,w)\,\mathrm{d}\,w = \frac{\sqrt{\pi}}{2a} erfc(ax) - I(a,b,x) \,,$$

$$\mathrm{U}(\mathrm{a},\mathrm{b},x) = \int_0^x \mathrm{e}^{-\mathrm{a}^2 w^2} \operatorname{erfc}(\mathrm{b}\,w)\,\mathrm{d}\,w, = I(a,b,0) - I(a,b,x) = \frac{1}{a\sqrt{\pi}} \tan^{-1}\frac{a}{b} - I(a.b.x)$$

$$\mathrm{V}(\mathrm{a},\mathrm{b},x) = \int_0^x \mathrm{e}^{-\mathrm{a}^2 w^2} \operatorname{erf}(\mathrm{b}\,w)\,\mathrm{d}\,w = \frac{\sqrt{\pi}}{2a} erf(ax) - U(a,b,x) = \frac{1}{a\sqrt{\pi}} \tan^{-1}\frac{b}{a} - J(a,b,x)$$

to get the expansions for J, U, and V.

APPENDIX

Inequalities

$$I(a,b,x) = \int_x^\infty e^{-a^2w^2}\,\mathrm{erfc}(bw)dw, \qquad a>0,\ b>0$$

We derived the relation

$$W \equiv \frac{\sqrt{\pi}}{2}\mathrm{erfc}(ax)\,\mathrm{erfc}(bx) = aI(a,b,x) + bI(b,a,x)$$

Divide W by $a+b$. Then the right side is a convex linear combination of $I(a,b,x)$ and $I(b,a,x)$. Therefore

$$\min[I(a,b,x), I(b,a,x)] \le \frac{W}{a+b} \le \max[I(a,b,x), I(b,a,x)]$$

Divide W by $\sqrt{a^2+b^2}$ and we have the dot product of two vectors, one of which has length 1. By the Cauchy inequality, we have

$$\left(\frac{W}{\sqrt{a^2+b^2}}\right)^2 \le I^2(a,b,x) + I^2(b,a,x)$$

Notice that if $b=a$, then the max and min are the same and $I(a,a,x) = \dfrac{W}{2a} = \dfrac{\sqrt{\pi}}{4a}\mathrm{erfc}^2(ax)$.

We have similar results for J(a,b,x):

$$\bar{W} \equiv \frac{\sqrt{\pi}}{2}\left[1 - \mathrm{erf}(ax)\,\mathrm{erf}(bx)\right] = aJ(a,b,x) + bJ(b,a,x)$$

$$\min[J(a,b,x),\ J(b,a,x)] \le \frac{\bar{W}}{a+b} \le \max[J(a,b,x),\ J(b,a,x)]$$

$$\left(\frac{\bar{W}}{\sqrt{a^2+b^2}}\right)^2 \le J^2(a,b,x) + J^2(b,a,x)$$

If a = b then the max and min are equal and

$$J(a,a,x) = \frac{\bar{W}}{2a} = \frac{\sqrt{\pi}}{4a}\left[1 - \mathrm{erf}^2(ax)\right]$$

Reference: See Θ and H Functions, Chapter 6, Beck et al.

Folder 6

Convergent Series for

$$I(a,b,T)=\int_T^\infty \frac{\mathrm{erf}(aw)\mathrm{erf}(bw)}{w^2}dw=\frac{1}{2}\int_0^t \frac{1}{\sqrt{u}}\mathrm{erf}\left(\frac{a}{\sqrt{u}}\right)\mathrm{erf}\left(\frac{b}{\sqrt{u}}\right)du\,,\; T=\frac{1}{\sqrt{t}}$$

a>0, b>0, t>0

Donald E. Amos, July 2000

Summary:

In Folder 3, a quadrature procedure was developed for I. In this folder, we develop two series representations for *I*, one symmetric in *a* and *b* and one not symmetric in *a* and *b*. The *symmetric* form is

$$I(a,b,T) \;=\; \frac{\mathrm{erf}(aT)\,\mathrm{erf}(bT)}{T}+\frac{aE_1(a^2T^2)}{\sqrt{\pi}}\mathrm{erf}(bT)$$

$$+\frac{bE_1(b^2T^2)}{\sqrt{\pi}}\mathrm{erf}(aT)+\frac{2}{\sqrt{\pi}}\left[a\ln\left(\frac{b+\sqrt{a^2+b^2}}{a}\right)+b\ln\left(\frac{a+\sqrt{a^2+b^2}}{b}\right)\right]\mathrm{erfc}(\sqrt{X})$$

$$-\frac{2abT}{\pi}\left[\sum_{k=0}^{\infty}\left(\frac{a^2}{a^2+b^2}\right)^k\frac{E_{k+1}(X)}{2k+1}+\sum_{k=0}^{\infty}\left(\frac{b^2}{a^2+b^2}\right)^k\frac{E_{k+1}(X)}{2k+1}\right]$$

where $T=1/\sqrt{t}$, $X=(a^2+b^2)/t=T^2(a^2+b^2)$.

For the special case, $t\to\infty$, $X\to 0$, $T\to 0$, we get

$$I(a,b,0)=\frac{2}{\sqrt{\pi}}\left[a\ln\left(\frac{b+\sqrt{a^2+b^2}}{a}\right)+b\ln\left(\frac{a+\sqrt{a^2+b^2}}{b}\right)\right]$$

The *non-symmetric* form for $a\le b$ is

$$I(a,b,T)=\mathrm{erf}(aT)\left[\frac{\mathrm{erf}(bT)}{T}+\frac{b}{\sqrt{\pi}}E_1(b^2T^2)\right]+\frac{a}{\sqrt{\pi}}E_1(a^2T^2)$$

$$-\frac{2a}{\sqrt{\pi}}\int_{\sqrt{X}}^{\infty}\frac{\mathrm{erfc}(x)}{x}dx+\frac{2}{\sqrt{\pi}}\left[-a\ln\left(\frac{2\sqrt{a^2+b^2}}{b+\sqrt{a^2+b^2}}\right)+b\ln\left(\frac{a+\sqrt{a^2+b^2}}{b}\right)\right]\mathrm{erfc}(\sqrt{X})$$

$$-\frac{2abT}{\pi}E_1(X)+\frac{a\sqrt{X}}{\pi}\sum_{k=1}^{\infty}\frac{(1/2)_k}{k!}\left(\frac{a^2}{a^2+b^2}\right)^k\frac{E_{k+1/2}(X)}{k}$$

$$-\frac{2abT}{\pi}\sum_{k=1}^{\infty}\left(\frac{a^2}{a^2+b^2}\right)^k\frac{E_{k+1}(X)}{2k+1}, \qquad T=1/\sqrt{t},\quad X=(a^2+b^2)/t=T^2(a^2+b^2)$$

where $$\int_{\sqrt{X}}^{\infty}\frac{\operatorname{erfc}(x)}{x}dx=\frac{1}{2}E_1(X)-e^{-X}\ln 2+\frac{X}{2\pi}\sum_{k=0}^{\infty}\frac{(1/2)_k}{k!}\frac{E_{k+1/2}(X)}{(k+1/2)^2}.$$

(See Folder 16 for a fast numerical algorithm for this integral.)

The symmetric form is computationally efficient for a and b relatively close together. When a and b are widely separated, one of the series in $a^2/(a^2+b^2)$ or $b^2/(a^2+b^2)$ can be slowly convergent. Therefore the non-symmetric form was developed for overall computational use. The main impediment to computation is the integral on $(\sqrt{X},\infty)$ of $\operatorname{erfc}(x)/x$ due to the slow convergence of the series: $0(1/k^{5/2})$ for practical purposes since the factor $E_{k+1/2}(X) \sim e^{-X}/(X+k+1/2)$ acts more like scale factor for small k. The series for this integral was accelerated by subtracting off 4 terms of the asymptotic expansion for E and adding back the closed form sum corresponding to each subtraction.

Since $I(a,b,T)$ is symmetric in a and b, this non-symmetric development for $a \le b$ can also be used for $a > b$ by exchanging a and b.

Check computations were made by comparing these formulae with the refined quadrature described in Folder 3.

General Development

We start by integrating $I(a,b,T)$ by parts (see Folder 3).

Let $u=\operatorname{erf}(aw)$ $\qquad dv=\frac{\operatorname{erf}(bw)}{w^2}dw$

$$du=\frac{2a}{\sqrt{\pi}}e^{-a^2w^2}dw \qquad v=-\left[\frac{\operatorname{erf}(bw)}{w}+\frac{b}{\sqrt{\pi}}E_1(b^2w^2)\right]$$

Then, $$I(a,b,T) = \operatorname{erf}(aT)\left[\frac{\operatorname{erf}(bT)}{T}+\frac{b}{\sqrt{\pi}}E_1(b^2T^2)\right]+$$

$$+\frac{2a}{\sqrt{\pi}}\int_T^{\infty}\frac{e^{-a^2w^2}\operatorname{erf}(bw)}{w}dw+\frac{2ab}{\pi}\int_T^{\infty}e^{-a^2w^2}E_1(b^2w^2)dw$$

This is the form we develop into the symmetric and non-symmetric cases.

Symmetric Case

Replace erf(bw) in the first integral by 1-erfc(bw):

$$\int_T^\infty e^{-a^2w^2}\,\frac{\mathrm{erf}(bw)}{w}dw = \int_T^\infty \frac{e^{-a^2w^2}}{w}dw - \int_T^\infty e^{-a^2w^2}\,\frac{\mathrm{erfc}(bw)}{w}dw$$

Integrate by parts: $u = \mathrm{erfc}(bw)$ $\qquad dv = \dfrac{e^{-a^2w^2}}{w}dw$

$$du = \frac{-2b}{\sqrt{\pi}}e^{-b^2w^2} \qquad v = -\frac{1}{2}E_1(a^2w^2)$$

Then,

$$\int_T^\infty e^{-a^2w^2}\,\frac{\mathrm{erf}(bw)}{w}dw = \frac{1}{2}E_1(a^2T^2)\,\mathrm{erf}(bT) + \frac{b}{\sqrt{\pi}}\int_T^\infty e^{-b^2w^2}E_1(a^2w^2)dw$$

This leads to the symmetric form:

$$I(a,b,T) = \frac{\mathrm{erf}(aT)\,\mathrm{erf}(bT)}{T} + \frac{a}{\sqrt{\pi}}E_1(a^2T^2)\,\mathrm{erf}(bT)$$

$$+\frac{b}{\sqrt{\pi}}E_1(b^2T^2)\,\mathrm{erf}(aT) + \frac{2ab}{\pi}\int_T^\infty e^{-a^2w^2}E_1(b^2w^2)dw + \frac{2ab}{\pi}\int_T^\infty e^{-b^2w^2}E_1(a^2w^2)dw$$

We now develop the series form for each of these integrals.

We start the series development by using the definition of E_1 and exchanging the orders of integration:

$$E_1(z) = \int_1^\infty \frac{e^{-zu}}{u}du, \qquad \int_T^\infty e^{-a^2w^2}E_1(b^2w^2)dw = \int_1^\infty \frac{du}{u}\int_T^\infty e^{-(a^2+b^2u)w^2}dw$$

and $$\int_T^\infty e^{-a^2w^2}E_1(b^2w^2)dw = \frac{\sqrt{\pi}}{2}\int_1^\infty \frac{\mathrm{erfc}(T\sqrt{a^2+b^2u})}{u\sqrt{a^2+b^2u}}du\,.$$

Let $\quad s=\sqrt{a^2+b^2u}, \qquad\qquad u=(s^2-a^2)/b^2$

and $\quad \int_T^\infty e^{-a^2w^2}E_1(b^2w^2)dw=\sqrt{\pi}\int_{\sqrt{a^2+b^2}}^\infty \dfrac{\operatorname{erfc}(Ts)}{s^2-a^2}ds=\sqrt{\pi}\sum_{k=0}^\infty a^{2k}\int_{\sqrt{a^2+b^2}}^\infty \dfrac{\operatorname{erfc}(Ts)}{s^{2k+2}}ds$

Integrate by parts:

$$u=\operatorname{erfc}(Ts) \qquad\qquad dv=s^{-2k-2}ds$$

$$du=\frac{-2T}{\sqrt{\pi}}e^{-T^2s^2} \qquad\qquad v=\frac{-1}{(2k+1)s^{2k+1}}$$

$$\int_{\sqrt{a^2+b^2}}^\infty \frac{\operatorname{erfc}(Ts)}{s^{2k+2}}ds=\frac{\operatorname{erfc}(\sqrt{X})}{(2k+1)(a^2+b^2)^{k+1/2}}-\frac{2T}{(2k+1)\sqrt{\pi}}\int_{\sqrt{a^2+b^2}}^\infty \frac{e^{-T^2s^2}}{s^{2k+1}}ds$$

$$=\frac{\operatorname{erfc}(\sqrt{X})}{(2k+1)(a^2+b^2)^{k+1/2}}-\frac{T}{(2k+1)\sqrt{\pi}}\frac{E_{k+1}(X)}{(a^2+b^2)^k}, \qquad k\ge 0$$

where $X=T^2(a^2+b^2)=(a^2+b^2)/t$.

Now, $\quad \int_T^\infty e^{-a^2w^2}E_1(b^2w^2)dw=\sqrt{\pi}\sum_{k=0}^\infty \dfrac{1}{(2k+1)}\left(\dfrac{a^2}{a^2+b^2}\right)^k\left[\dfrac{\operatorname{erfc}\sqrt{X}}{\sqrt{a^2+b^2}}-\dfrac{T}{\sqrt{\pi}}E_{k+1}(X)\right]$

Now, $\quad \sum_{k=0}^\infty \dfrac{z^k}{2k+1}=\dfrac{1}{2\sqrt{z}}\ln\left(\dfrac{1+\sqrt{z}}{1-\sqrt{z}}\right)$

and with $z=a^2/(a^2+b^2)$ we have

$$\int_T^\infty e^{-a^2w^2}E_1(b^2w^2)dw=\frac{\sqrt{\pi}}{2a}\operatorname{erfc}(\sqrt{X})\ln\left(\frac{a+\sqrt{a^2+b^2}}{-a+\sqrt{a^2+b^2}}\right)-T\sum_{k=0}^\infty \frac{1}{(2k+1)}\left(\frac{a^2}{a^2+b^2}\right)^k E_{k+1}(X)$$

$$=\frac{\sqrt{\pi}}{a}\ln\left(\frac{a+\sqrt{a^2+b^2}}{b}\right)\operatorname{erfc}(\sqrt{X})-T\sum_{k=0}^\infty \left(\frac{a^2}{a^2+b^2}\right)^k\frac{E_{k+1}(X)}{2k+1}$$

Now, exchanging a and b we get the second integral in the symmetric form. Thus,

$$I(a,b,T)=\frac{\operatorname{erf}(aT)\operatorname{erf}(bT)}{T}+\frac{a}{\sqrt{\pi}}E_1(a^2T^2)\operatorname{erf}(bT)\ +$$

$$+\frac{b}{\sqrt{\pi}}E_1(b^2T^2)\operatorname{erf}(aT)+\frac{2}{\sqrt{\pi}}\left[a\ln\left(\frac{b+\sqrt{a^2+b^2}}{a}\right)+b\ln\left(\frac{a+\sqrt{a^2+b^2}}{b}\right)\right]\operatorname{erfc}(\sqrt{X})$$

$$-\frac{2abT}{\pi}\left[\sum_{k=0}^{\infty}\left(\frac{a^2}{a^2+b^2}\right)^k\frac{E_{k+1}(X)}{2k+1}+\sum_{k=0}^{\infty}\left(\frac{b^2}{a^2+b^2}\right)^k\frac{E_{k+1}(X)}{2k+1}\right]$$

Now, for $t\to\infty,\ T\to 0,\ X\to 0$ and we get a closed form for

$$I(a,b,0)=\int_0^\infty \frac{\operatorname{erf}(aw)\operatorname{erf}(bw)}{w^2}dw=\frac{2}{\sqrt{\pi}}\left[a\ln\left(\frac{b+\sqrt{a^2+b^2}}{a}\right)+b\ln\left(\frac{a+\sqrt{a^2+b^2}}{b}\right)\right]$$

Non-Symmetric Case

In the General Development Section, we derived the formula

$$I(a,b,T)=\operatorname{erf}(aT)\left[\frac{\operatorname{erf}(bT)}{T}+\frac{b}{\sqrt{\pi}}E_1(b^2T^2)\right]$$

$$+\frac{2a}{\sqrt{\pi}}\int_T^\infty e^{-a^2w^2}\frac{\operatorname{erf}(bw)}{w}dw+\frac{2ab}{\pi}\int_T^\infty e^{-a^2w^2}E_1(b^2w^2)dw$$

In the section on the Symmetric Case, a series for the second integral was derived which converged well for $a\le b$. We therefore concentrate on the first integral, and exchange a and b in the final formula to get the $a>b$ case. Now, replace erf(bw) = 1-erfc(bw) to get

$$\int_T^\infty e^{-a^2w^2}\frac{\operatorname{erf}(bw)}{w}dw=\int_T^\infty \frac{e^{-a^2w^2}}{w}dw-\int_T^\infty \frac{\operatorname{erfc}(bw)}{w}e^{-a^2w^2}dw$$

$$=\frac{1}{2}E_1(a^2T^2)-\frac{2b}{\sqrt{\pi}}\int_1^\infty du\int_T^\infty e^{-(a^2+b^2u^2)w^2}dw$$

$$=\frac{1}{2}E_1(a^2T^2)-b\int_1^\infty \frac{\operatorname{erfc}(T\sqrt{a^2+b^2u^2})}{\sqrt{a^2+b^2u^2}}du$$

Let $\quad s=\sqrt{a^2+b^2u^2}, \qquad u=(\sqrt{s^2-a^2})/b$

$$\int_1^\infty \frac{\operatorname{erfc}(T\sqrt{a^2+b^2u^2})}{\sqrt{a^2+b^2u^2}}du=\frac{1}{b}\int_{\sqrt{a^2+b^2}}^\infty \frac{\operatorname{erfc}(Ts)}{\sqrt{s^2-a^2}}ds=\frac{1}{b}\sum_{k=0}^\infty C_k(a^2)^k\int_{\sqrt{a^2+b^2}}^\infty \frac{\operatorname{erfc}(Ts)}{s^{2k+1}}ds$$

Now this last integral was integrated in the symmetric case with $2k+2$ in place of $2k+1$. Here $C_k = (1/2)_k/k!$

Therefore if we replace k by k-1/2 in the previous formula, we get

$$\int_{\sqrt{a^2+b^2}}^{\infty} \frac{\operatorname{erfc}(Ts)}{s^{2k+1}} ds = \frac{\operatorname{erfc}(\sqrt{X})}{2k(a^2+b^2)^k} - \frac{T}{2k\sqrt{\pi}} \frac{E_{k+1/2}(X)}{(a^2+b^2)^{k-1/2}} \qquad k \geq 1$$

Then,

$$\int_T^{\infty} e^{-a^2w^2} \frac{\operatorname{erf}(bw)}{w} dw = \frac{1}{2} E_1(a^2T^2) - \int_{\sqrt{X}}^{\infty} \frac{\operatorname{erfc}(x)}{x} dx$$

$$-\frac{1}{2}\sum_{k=1}^{\infty} \frac{(1/2)_k}{k \cdot k!} \left(\frac{a^2}{a^2+b^2}\right)^k \left[\operatorname{erfc}(\sqrt{X}) - \frac{\sqrt{X}}{\sqrt{\pi}} E_{k+1/2}(X)\right] \qquad X = T^2(a^2+b^2)$$

Now, $\displaystyle\sum_{k=1}^{\infty} \frac{(1/2)_k}{k \cdot k!} z^k = 2\ln\left(\frac{2}{1+\sqrt{1-z}}\right)$ and with $z = a^2/(a^2+b^2)$, we get

$$\int_T^{\infty} e^{-a^2w^2} \frac{\operatorname{erf}(bw)}{w} dw = \frac{1}{2} E_1(a^2T^2) - \int_{\sqrt{X}}^{\infty} \frac{\operatorname{erfc}(x)}{x} dx$$

$$-\ln\left(\frac{2\sqrt{a^2+b^2}}{b+\sqrt{a^2+b^2}}\right)\operatorname{erfc}(\sqrt{X}) + \frac{\sqrt{X}}{2\sqrt{\pi}} \sum_{k=1}^{\infty} C_k \left(\frac{a^2}{a^2+b^2}\right)^k \frac{E_{k+1/2}(X)}{k}$$

Thus, the non-symmetric form for $a \leq b$ is

$$I(a,b,T) = \operatorname{erf}(aT)\left[\frac{\operatorname{erf}(bT)}{T} + \frac{b}{\sqrt{\pi}} E_1(b^2T^2)\right] + \frac{a}{\sqrt{\pi}} E_1(a^2T^2)$$

$$+\frac{2}{\sqrt{\pi}}\left[b\ln\left(\frac{a+\sqrt{a^2+b^2}}{b}\right) - a\ln\left(\frac{2\sqrt{a^2+b^2}}{b+\sqrt{a^2+b^2}}\right)\right]\operatorname{erfc}(\sqrt{X})$$

$$-\frac{2a}{\sqrt{\pi}}\int_{\sqrt{X}}^{\infty} \frac{\operatorname{erfc}(x)}{x} dx + \frac{a\sqrt{X}}{\pi} \sum_{k=1}^{\infty} C_k \left(\frac{a^2}{a^2+b^2}\right)^k \frac{E_{k+1/2}(X)}{k}$$

$$-\frac{2abT}{\pi}\sum_{k=0}^{\infty} \left(\frac{a^2}{a^2+b^2}\right)^k \frac{E_{k+1}(X)}{2k+1}, \qquad C_k = (1/2)_k/k!$$

Computation of $\int_{\sqrt{x}}^{\infty}$ erfc(x)/xdx (See Folder 16 also)

Now we concentrate on the integral of erfc(x)/x on $(\sqrt{X},\infty)$. In the Special Cases Section of Folder 5, we obtained the result

$$\operatorname{erfc}(x) = \frac{1}{\pi}\sum_{k=0}^{\infty} C_k E_{k+3/2}(x^2), \qquad C_k = (1/2)_k/k!$$

We mean to exploit this to compute the integral of erfc(x)/x:

$$\int_{\sqrt{X}}^{\infty} \frac{\operatorname{erfc}(x)}{x}dx = \frac{1}{\pi}\sum_{k=0}^{\infty} C_k \int_{\sqrt{X}}^{\infty} \frac{E_{k+3/2}(x^2)}{x}dx$$

Let $x^2 = u$ and

$$= \frac{1}{2\pi}\sum_{k=0}^{\infty} C_k \int_{X}^{\infty} \frac{E_{k+3/2}(u)}{u}du$$

Now, replace $E_{k+3/2}(u) = [e^{-u} - uE_{k+1/2}(u)]/(k+1/2)$ and integrate

$$\int_{X}^{\infty} \frac{E_{k+3/2}(u)}{u}du = \frac{1}{k+1/2}\left[\int_{X}^{\infty} \frac{e^{-u}}{u}du - \int_{X}^{\infty} E_{k+1/2}(u)du\right] = \frac{1}{k+1/2}\left[E_1(X) - E_{k+3/2}(X)\right]$$

Then

$$\int_{\sqrt{X}}^{\infty} \frac{\operatorname{erfc}(x)}{x}dx = \frac{1}{2\pi}E_1(X)\sum_{k=0}^{\infty} \frac{C_k}{(k+1/2)} - \frac{1}{2\pi}\sum_{k=0}^{\infty} \frac{C_k}{(k+1/2)}E_{k+3/2}(X)$$

Now we replace $E_{k+3/2}(X)$ by the expression above to get

$$\int_{\sqrt{X}}^{\infty} \frac{\operatorname{erfc}(x)}{x}dx = \frac{1}{2}E_1(X) - (\ln 2)e^{-X} + \frac{X}{2\pi}\sum_{k=0}^{\infty} \frac{C_k}{(k+1/2)^2}E_{k+1/2}(X)$$

since $\sum_{k=0}^{\infty} \frac{C_k}{k+1/2} = \pi$ and $\sum_{k=0}^{\infty} \frac{C_k}{(k+1/2)^2} = 2\pi\ln 2$. These are developed later in the section on "Closed Form Sums".

Now, $C_k = O(k^{-1/2})$ and the coefficient $C_k/(k+1/2)^2 = O(k^{-5/2})$. Since the first term of the asymptotic expansion for $E_{k+1/2}(z)$ is $e^{-z}/(z+k+1/2)$, the effect of k is delayed by a large z and the useful convergence is more like $O(k^{-5/2})$. To accelerate the convergence, we subtract off 4 terms of the asymptotic expansion of $E_{k+1/2}(x)$ for large k and add a closed form sum for each term subtracted. Thus, for $n = k+1/2$ (A&S, p. 231)

$$E_n(X) = e^{-X}\left[t_{1k} + t_{2k} + t_{3k} + t_{4k}\right] + O\left(\frac{e^{-X}}{n^5}\right)$$

and we want to sum the expressions

$$\sum_{k=0}^{\infty} \frac{C_k}{(k+1/2)^2} t_{ik} = S_i(X), \qquad i = 1,2,3,4$$

where $t_{1k} = \dfrac{1}{X+n}$, $\quad t_{2k} = \dfrac{n}{(X+n)^3}$, $\quad t_{3k} = \dfrac{n(n-2X)}{(X+n)^5}$, $\quad t_{4k} = \dfrac{n(6X^2 - 8nX + n^2)}{(X+n)^7}$,

with $n = k$+1/2. Then, the sum takes the form

$$\left\{\sum_{k=0}^{\infty} \frac{C_k}{(k+1/2)^2}\left[e^X E_{k+1/2}(X) - t_{1k} - t_{2k} - t_{3k} - t_{4k}\right] + \sum_{i=1}^{4} S_i(x)\right\} e^{-X}$$

While the handbooks (A&S, p. 231) seem to indicate that n should be an integer, the expressions seem to work well for n=k+1/2.

<u>Closed Form Sums $S_i(X)$</u>

We start by manipulating the power series for $(1-z)^{-1/2} = \sum_{k=0}^{\infty} C_k z^k$ to get the multipliers.

Thus, $\displaystyle\sum_{k=0}^{\infty} C_k z^{k+x-1/2} = z^{x-1/2}(1-z)^{-1/2}, \qquad C_k = (1/2)_k / k!\ , \quad k \ge 0$

and $\displaystyle\sum_{k=0}^{\infty} \frac{C_k z^{k+x+1/2}}{x+k+1/2} = \int_0^z t^{x-1/2}(1-t)^{-1/2}\,dt$

and for z = 1, define $F(x)$ by

$$F(x) = \sum_{k=0}^{\infty} \frac{C_k}{x+k+1/2} = \int_0^1 t^{x-1/2}(1-t)^{-1/2}\,dt = \frac{\Gamma(x+1/2)\Gamma(1/2)}{\Gamma(x+1)}$$

Let $\displaystyle f(x) = \frac{\Gamma(x+1/2)}{\Gamma(x+1)}, \qquad F(x) = \sqrt{\pi} f(x)$

Now differentiate $F(x)$ to get

$$-F'(x)=\sum_{k=0}^{\infty}\frac{C_k}{(x+k+1/2)^2}=-\sqrt{\pi}f'(x),\qquad \ln f(x)=\ln\Gamma(x+1/2)-\ln\Gamma(x+1)$$

$$f'(x)=f(x)[\psi(x+1/2)-\psi(x+1)]$$

at $x = 0$ we get

$$F(0)=\sum_{k=0}^{\infty}\frac{C_k}{(k+1/2)}=\sqrt{\pi}\frac{\Gamma(1/2)}{\Gamma(1)}=\pi$$

$$-F'(0)=\sum_{k=0}^{\infty}\frac{C_k}{(k+1/2)^2}=-\sqrt{\pi}\cdot\sqrt{\pi}\left[\psi\left(\frac{1}{2}\right)-\psi(1)\right]=-\pi[-\gamma-2\ln 2+\gamma]=2\pi\ln 2$$

Now, from above, with $x = 0$, divide by z and integrate

$$\sum_{k=0}^{\infty}C_k\frac{z^{k+1/2}}{(k+1/2)^2}=\int_0^z\frac{1}{v}\int_0^v t^{-1/2}(1-t)^{-1/2}\,dtdv=2\int_0^z\frac{\sin^{-1}\sqrt{v}}{v}dv$$

$$\sum_{k=0}^{\infty}\frac{C_k}{(k+1/2)^2}z^{y+k-1/2}=2z^{y-1}\int_0^z\frac{\sin^{-1}\sqrt{v}}{v}dv$$

$$\sum_{k=0}^{\infty}\frac{C_k}{(k+1/2)^2}\frac{1}{y+k+1/2}=2\int_0^1 z^{y-1}\int_0^z\frac{\sin^{-1}\sqrt{v}}{v}dvdz$$

Let $G(y)=\sum_{k=0}^{\infty}\frac{G_k}{y+k+1/2}=2\int_0^1 z^{y-1}\int_0^z\frac{\sin^{-1}\sqrt{v}}{v}dvdz\equiv S_1(y)$ corresponding to the sum on t_{1k}

where $G_k = C_k/(k+1/2)^2 = (1/2)_k/[k!(k+1/2)^2]$.

Now exchange the order of summation on the unit triangle to get

$$G(y)=2\int_0^1\frac{\sin^{-1}\sqrt{v}}{v}\int_v^1 z^{y-1}dzdv=2\int_0^1\frac{\sin^{-1}\sqrt{v}}{v}\left[\frac{1}{y}-\frac{v^y}{y}\right]dv$$

$$=\frac{2}{y}\int_0^1\frac{\sin^{-1}\sqrt{v}}{v}dv-\frac{2}{y}\int_0^1 v^{y-1}\sin^{-1}\sqrt{v}dv$$

Let $v=\sin^2\theta$, $dv=2\sin\theta\cos\theta d\theta$

$$= \frac{4}{y}\int_0^{\pi/2} \theta\frac{\cos\theta}{\sin\theta}d\theta - \frac{4}{y}\int_0^{\pi/2} \theta\sin^{2y-1}(\theta)\cos\theta d\theta$$

Now integrate each by parts to eliminate θ as a multiplying factor:

$$= \frac{4}{y}\left[\theta\ln\sin\theta\Big|_0^{\pi/2} - \int_0^{\pi/2} \ln\sin\theta d\theta\right] - \frac{4}{y}\left[\frac{\theta\sin^{2y}\theta}{2y}\Big|_0^{\pi/2} - \int_0^{\pi/2} \frac{\sin^{2y}\theta}{2y}d\theta\right]$$

Now these integrals are known:

$$\int_0^{\pi/2} \ln\sin\theta d\theta = -\frac{\pi}{2}\ln 2, \qquad \int_0^{\pi/2} \sin^{2y}\theta d\theta = \frac{\sqrt{\pi}}{2}\frac{\Gamma(y+1/2)}{\Gamma(y+1)}$$

and

$$G(y) = \frac{2\pi\ln 2}{y} - \frac{\pi}{y^2} + \frac{\sqrt{\pi}}{y^2}\frac{\Gamma(y+1/2)}{\Gamma(y+1)} = \frac{2\pi\ln 2}{y} - \frac{\pi}{y^2} + \sqrt{\pi}\frac{f(y)}{y^2}.$$

Sum $\underline{S_1(y)} = G(y)$, corresponding to the term t_{1k}.

To compute $S_2(y)$ we need $\sum_{k=0}^{\infty} G_k \frac{(k+1/2)}{(y+k+1/2)^3}$ corresponding to the term t_{2k} where $G_k = C_k/(k+1/2)^2$. Notice that

$$-F'(0) = \sum_{k=0}^{\infty} G_k, \quad yG(y) = \sum_{k=0}^{\infty}\frac{yG_k}{y+k+1/2} = \sum_{k=0}^{\infty} G_k\left[1 - \frac{k+1/2}{y+k+1/2}\right]$$

and

$$-F'(0) - yG(y) = \sum_{k=0}^{\infty} G_k \frac{(k+1/2)}{y+k+1/2}$$

Differentiation yields

$$[yG(y)]' = \sum_{k=0}^{\infty} G_k \frac{(k+1/2)}{(y+k+1/2)^2}, \qquad -[yG(y)]'' = \sum_{k=0}^{\infty} G_k \frac{2(k+1/2)}{(y+k+1/2)^3}$$

Let $R(y) = -[yG(y)]''/2$ then $S_2(y) = R(y)$.

To compute, $R(y)$ explicitly, with $G(y) = \frac{2\pi\ln 2}{y} - \frac{\pi}{y^2} + \frac{\sqrt{\pi}}{y^2}f(y)$ is not hard, but in the development to follow, we will need $[yG(y)]^{(n)}$ for n=2, 4, and 6. Therefore we develop a closed form for $[yG(y)]^{(n)}$ for all positive n. Now, using the Leibniz rule for differentiation we have

$$[yG(y)]^{(n)} = -\pi\left(\frac{1}{y}\right)^{(n)} + n!\sqrt{\pi}\sum_{k=0}^{n}\frac{f^{(k)}(y)}{k!}\frac{(1/y)^{(n-k)}}{(n-k)!}$$

and with $\left(\frac{1}{y}\right)^{(n)} = \frac{(-1)^n n!}{y^{n+1}}, \quad n \geq 0$

we get

$$[yG(y)]^{(n)} = \pi \frac{(-1)^{n+1} n!}{y^{n+1}} + \frac{n!\sqrt{\pi}}{y} \sum_{k=0}^{n} \frac{f^{(k)}(y)}{k!} \frac{(-1)^{n-k}}{y^{n-k}}.$$

Then for n=2, we get

$$S_2(y) = R(y) = -[yG(y)]''/2 = \frac{\pi}{y^3} - \frac{\sqrt{\pi}}{y}\left[\frac{f(y)}{y^2} - \frac{f'(y)}{y} + \frac{f''(y)}{2}\right]$$

To get $S_3(y)$ we need $\sum_{k=0}^{\infty} G_k \frac{(k+1/2)^2 - 2y(k+1/2)}{(y+k+1/2)^5}$ corresponding to term t_{3k}.

Notice that

$$\frac{R''(y)}{3\cdot 4} = \sum_{k=0}^{\infty} G_k \frac{(k+1/2)}{(y+k+1/2)^5}, \quad \frac{F^{(4)}(y)}{4!} = \sum_{k=0}^{\infty} G_k \frac{(k+1/2)^2}{(y+k+1/2)^5}$$

and

$$S_3(y) = \frac{F^{(4)}(y)}{4!} - \frac{2yR''(y)}{3\cdot 4}, \quad F^{(4)}(y) = \sqrt{\pi} f^{(4)}(y), \quad R''(y) = -\frac{1}{2}[yG(y)]^{(4)}.$$

Using the general formula for $[yG(y)]^{(n)}$ with n=4 we have

$$\underline{S_3(y)} = -\frac{2\pi}{y^4} + \sqrt{\pi}\left[\frac{2f(y)}{y^4} - \frac{2f'(y)}{y^3} + \frac{f''(y)}{y^2} - \frac{f'''(y)}{3y} + \frac{f^{(4)}(y)}{8}\right]$$

To compute $S_4(y)$ we need $\sum_{k=0}^{\infty} G_k \frac{(k+1/2)(6y^2) - 8(k+1/2)^2 y + (k+1/2)^3}{(y+k+1/2)^7}$ corresponding to term t_{4k}.

Notice $[yF(y)]' = F(y) + yF'(y) = \sum_{k=0}^{\infty} C_k \frac{(k+1/2)}{(y+k+1/2)^2} = \sum_{k=0}^{\infty} G_k \frac{(k+1/2)^3}{(y+k+1/2)^2}$

$$\underline{S_4(y)} = 6y^2 \frac{R^{(4)}(y)}{3\cdot 4\cdot 5\cdot 6} - 8y\frac{F^{(6)}(y)}{6!} - \frac{[yF(y)]^{(6)}}{6!}$$

Using the formula for $[yG(y)]^{(n)}$ again with n=6, we have

$$R^{(4)}(y) = -\frac{1}{2}[yG(y)]^{(6)} = +\frac{360\pi}{y^7} - \frac{\sqrt{\pi}}{2y}\left[\frac{720 f(y)}{y^6} - \frac{720 f'(y)}{y^5} + \frac{360 f''(y)}{y^4} - \frac{120 f'''(y)}{y^3} + \frac{30 f^{(4)}(y)}{y^2} - \frac{6 f^{(5)}(y)}{y} + f^{(6)}(y)\right]$$

and

$$F^{(6)}(y) = \sqrt{\pi} f^{(6)}(y), \qquad [yF(y)]^{(6)} = \sqrt{\pi}\left[yf^{(6)}(y) + 6 f^{(5)}(y)\right]$$

$$\underline{S_4(y)} = \frac{6\pi}{y^5} - \frac{\sqrt{\pi}}{y}\left\{\frac{6 f(y)}{y^4} - \frac{6 f'(y)}{y^3} + \frac{3 f''(y)}{y^2} - \frac{f'''(y)}{y} + \frac{f^{(4)}(y)}{4} - \frac{yf^{(5)}(y)}{24} + \frac{y^2 f^{(6)}(y)}{48}\right\}$$

Numerical Computation

The sums $S_i(y)$ are expressed in terms of $f(y)$ and the derivatives $f^{(k)}(y)$, k = 1,6 and i = 1,4. The numerical computation of these derivatives is easy because of the relations (logarithmic differentiation)

$$f(y) = \frac{\Gamma(y+1/2)}{\Gamma(y+1)}, \qquad f'(y) = f(y)[\psi(y+1/2) - \psi(y+1)] = f(y)\phi(y)$$

and the recursive computation using the Leibniz rule

$$\left[\frac{f^{(n+1)}(y)}{(n+1)!}\right] = \frac{1}{n+1}\sum_{k=0}^{n}\left[\frac{f^{(k)}(y)}{k!}\right]\frac{\phi^{(n-k)}(y)}{(n-k)!}, \qquad n = 1,5$$

where $\phi^{(k)}(y) = \psi^{(k)}(y+1/2)-\psi^{(k)}(y+1)$. The gamma functions and derivatives $\psi^{(k)}(y)$ can be computed by the SLATEC routines DGAMLN and DPSIFN where DGAMLN computes $\ln\Gamma(y)$ and the sequence of scaled derivatives $(-1)^k\psi^k(y)/k!$ are returned from DPSIFN. Thus, the recursive computation for $f^{(k)}(y)/k!$, k = 1,6 can be done easily.

There is, however, another problem with the computation of $S_i(y)$, i = 1,4. The function $f(y)$ and $S_i(y)$ are analytic in a circle about the origin and each has a power series whose radius of convergence is ½. This means that the "singular terms" involving y^{-k} must all cancel out in the development of the power series. It also means that direct computation for small y yields large numbers of different signs which means losses of significance. Therefore, a suite of routines was written to manipulate the coefficients of power series with both positive and negative powers to include multiplication by a constant, multiplication by positive or negative powers, differentiation and integration of the power series, etc. With these routines, the power series for $S_i(y)$, i = 1,4 were developed from the power series for $f(y)$ using the recurrence above with n = 45, and y = 0:

$$f(y) = \sum_{k=0}^{\infty}\left[\frac{f^{(k)}(0)}{k!}\right]y^k \qquad |y| < \frac{1}{8}$$

The $S_i(y)$ coefficients are shown on following pages and power series was applied for $y \le 1/8$, with 36 terms. These parameters for the $S_i(y)$, $i = 1,4$ were chosen so that the last term would be $O(10^{-15})$. The accelerated series was truncated at 70 terms for errors on the order of 0.1×10^{-10}.

Checking this non-symmetric form on wide range of parameters a,b,T was done numerically. The series computation was compared with the quadrature of Folder 3 (Refined Form) with maximum errors 0.5E-11 (about 11 digits of significance).

Final Comments

These series of exponential integrals may seem to involve a lot of computation, but once the member of the sequence needed to start the recurrence is generated, each successive member is obtained by one multiplication and one division using a two-term recurrence. The same concepts apply to the scaled ψ functions $(-1)^k\psi^{(k)}(z)/k!$ Recurrences and any scaling are all done inside the routines DEXINT, DHEXINT, DPSIFN. $f(y)$ is computed by $f(y) = \exp\{\ln\Gamma(y+1/2)-\ln\Gamma(y+1)\}$ using the $\ln\Gamma$ routine from DGAMLN. The coefficients of the exponential integrals are generated by recurrence:

$$C_{k+1} = C_k(k+1/2)/(k+1), \quad Q_{k+1} = Q_k\left[(k+1/2)/(k+1)\right]\cdot\frac{a^2}{a^2+b^2}, \quad P_{k+1} = P_k\cdot\frac{a^2}{a^2+b^2}$$

Furthermore, the half odd order sequences for $E_{k+1/2}$ for the main formula and the $\mathrm{erfc}(x)/x$ integral need be generated only once since they both have the argument X.

The $S_i(y)$ sums can be arranged into a computational form:

$$S_1(y) = (2\pi\ln 2)/y + \sqrt{\pi}(f(y)-\sqrt{\pi})/y^2$$

$$S_2(y) = -\frac{\sqrt{\pi}}{2}\left\{f''(y) - 2f'(y)/y + 2[f(y)-\sqrt{\pi}]/y^2\right\}/y$$

$$S_3(y) = \sqrt{\pi}\left\{f^{(4)}(y)/8 - f^{(3)}(y)/(3y) + f^{(2)}(y)/y^2 - 2f'(y)/y^3 + 2[f(y)-\sqrt{\pi}]/y^4\right\}$$

$$S_4(y) = -\sqrt{\pi}\left\{y^2 f^{(6)}(y)/48 - yf^{(5)}(y)/24 + f^{(4)}(y)/4 - f^{(3)}(y)/y \right.$$
$$\left. + 3f''(y)/y^2 - 6f'(y)/y^3 + 6[f(y)-\sqrt{\pi}]/y^4\right\}/y$$

The symmetric form of $I(a,b,0)$ for $T = 0$ can also be derived from the non-symmetric case by grouping together all terms which do not contain T, X or $\sqrt{X}$ as multipliers since these terms go to zero as $T \to 0$, $X = T^2(a^2+b^2) \to 0$. The logarithmic behavior of $E_1(z) = -\gamma-\ln z+O(z)$ for $z \to 0$ plays a role in analyzing $E_1(a^2T^2)$ and $E_1[T^2(a^2+b^2)]$. Notice also that $\mathrm{erf}(cT) = O(T)$ for $T \to 0$ and these terms drop out also.

S COEFFICIENTS FOR THE POWER SERIES IN y, $|y| < 1/8$

```
 DATA (S(1,J),J=1,36)/
&0.818648809789096D+01,-.161116877594002D+02,0.320705147479283D+02,
&-.640456488286857D+02,0.128029936468548D+03,-.256019770882582D+03,
&0.512013108846216D+03,-.102400871131162D+04,0.204800579658222D+04,
&-.409600386006158D+04,0.819200257165949D+04,-.163840017137581D+05,
&0.327680011422340D+05,-.655360007613811D+05,0.131072000507544D+06,
&-.262144000338346D+06,0.524288000225557D+06,-.104857600015037D+07,
&0.209715200010024D+07,-.419430400006683D+07,0.838860800004455D+07,
&-.167772160000297D+08,0.335544320000198D+08,-.671088640000132D+08,
&0.134217728000009D+09,-.268435456000006D+09,0.536870912000004D+09,
&-.107374182400000D+10,0.214748364800000D+10,-.429496729600000D+10,
&0.858993459200000D+10,-.171798691840000D+11,0.343597383680000D+11,
&-.687194767360000D+11,0.137438953472000D+12,-.274877906944000D+12/

 DATA (S(2,J),J=1,36)/
&0.161116877594002D+02,-.962115442437849D+02,0.384273892972114D+03,
&-.128029936468548D+04,0.384029656323872D+04,-.107522752857705D+05,
&0.286722439167252D+05,-.737282086769598D+05,0.184320173702771D+06,
&-.450560141441272D+06,0.108134411310803D+07,-.255590408909425D+07,
&0.596377606928568D+07,-.137625600532921D+08,0.314572800406015D+08,
&-.713031680306757D+08,0.160432128023006D+09,-.358612992017142D+09,
&0.796917760012697D+09,-.176160768000936D+10,0.387553689600686D+10,
&-.848927129600501D+10,0.185220464640036D+11,-.402653184000026D+11,
&0.872415232000019D+11,-.188441690112001D+12,0.405874409472001D+12,
&-.871878361088000D+12,0.186831077376000D+13,-.399431958528000D+13,
&0.852121511526400D+13,-.181419418583040D+14,0.385516264488960D+14,
&-.817761773158400D+14,0.173173081374720D+15,-.366137372049408D+15/

 DATA (S(3,J),J=1,36)/
&0.320705147479283D+02,-.448319541800800D+03,0.320074841171370D+04,
&-.166412851073678D+05,0.716818352384702D+05,-.272386317208890D+06,
&0.946178678020984D+06,-.307200289504619D+07,0.946176297026671D+07,
&-.279347229219575D+08,0.796590107767708D+08,-.220659714563570D+09,
&0.596377602309326D+09,-.157810688203684D+10,0.409993216176385D+10,
&-.104815656975031D+11,0.264178237452628D+11,-.657457152010475D+11,
&0.161774305280859D+12,-.394012917760697D+12,0.950798385152561D+12,
&-.227512470732845D+13,0.540226355200035D+13,-.127372623872003D+14,
&0.298366009344002D+14,-.694721697546242D+14,0.160861557620736D+15,
&-.370548303462400D+15,0.849458631802880D+15,-.193857643872256D+16,
&0.440546821459149D+16,-.997202070811443D+16,0.224884487618560D+17,
&-.505376775811891D+17,0.113197470858609D+18,-.252756832504775D+18/

 DATA (S(4,J),J=1,36)/
&0.640456488286857D+02,-.192044904702822D+04,0.230417793794323D+05,
&-.179204588096175D+06,0.107520914687720D+07,-.541902333775654D+07,
&0.240845026971621D+08,-.973209905513148D+08,0.364953638173961D+09,
&-.128860164491835D+10,0.432970142630140D+10,-.139552358940382D+11,
&0.434162893360368D+11,-.131019571256367D+12,0.385037107255215D+12,
&-.110548431672484D+13,0.310917464068954D+13,-.858519502852560D+13,
&0.233178136576413D+14,-.623961440256368D+14,0.164725820227616D+15,
&-.429557127577628D+15,0.110761837854722D+16,-.282662535168002D+16,
&0.714508075008002D+16,-.179027143274005D+17,0.444919527663206D+17,
&-.109734610526536D+18,0.268737821697638D+18,-.653790229718630D+18,
&0.158071948874193D+19,-.379964830280319D+19,0.908353422388887D+19,
&-.216036305232986D+20,0.511310840066998D+20,-.120460659953743D+21/
```

Folder 7

Evaluation of

$$\mathbf{I_3(a,b,c,T) = \int_T^\infty e^{-c^2w^2} erf(aw)erf(bw)dw = \frac{1}{2}\int_0^t \frac{e^{-c^2/u}}{u^{3/2}} erf\left(\frac{a}{\sqrt{u}}\right) erf\left(\frac{b}{\sqrt{u}}\right) du}$$

$$\mathbf{I_3^c(a,b,c,T) \equiv J_3(a,b,c,T) = \int_T^\infty e^{-c^2w^2} erfc(aw)erfc(bw)dw = \frac{1}{2}\int_0^t \frac{e^{-c^2/u}}{u^{3/2}} erfc\left(\frac{a}{\sqrt{u}}\right) erfc\left(\frac{b}{\sqrt{u}}\right) du}$$

$$\mathbf{a \ge 0,\ b \ge 0,\ c>0,\ T = \frac{1}{\sqrt{t}}}$$

Donald E. Amos, October 2000, December 2005

Summary

Folder 7 is broken down into 6 subfolders: Folders 7a-7f and an APPENDIX.

Folder 7a contains the derivation of a series analogous to those of Folder 5. If we take $a = \min(a,b)$ and $b = \max(a,b)$, we have series representations in terms of exponential integrals for $a,c \le b$ and $a \le b \le c$. These two cases cover all permutations of parameters a,b and c.

Folder 7b contains the derivation of the power series for small a,b and c. This series outperforms, in terms of relative error, the evaluations in Folder 7a when aT, bT and cT are all less than 2.

Folder 7c contains a series expansion for large c satisfying the restriction

$$\frac{a^2+b^2}{c^2} \le \frac{1}{2p} \quad \text{where} \quad p = \begin{cases} 1 & \text{for} \quad c^2T^2 \le 5 \\ c^2T^2 - 4 & \text{for} \quad c^2T^2 > 5 \end{cases}$$

Folder 7d contains a quadrature procedure used previously in Folder 3 where the erf function is replaced by 1 when the argument exceeds 6.

Folder 7e contains some closed form special case evaluations for

$$J_3(a,b,c,T) = I_3^c(a,b,c,T) = \int_T^\infty e^{-c^2w^2} \operatorname{erfc}(aw)\operatorname{erfc}(bw)dw,$$

namely for $J_3(a,b,0,T)$, $J_3(0,b,0,T)$ and $J_3(a,b,0,0)$. See Folder 9 for an integral related to $I_3(a,b,c,T)$

Folder 7f combines the main expressions into representations which have better computational properties.

The APPENDIX contains some inequalities satisfied by $J_3(a,b,c,T)$ and functions obtained by permuting its parameters. These inequalities have their analogues in Folder 5 also.

Folder 7a

Series Expansion for

$$\boldsymbol{I_3(a,b,c,T)=\int_T^\infty e^{-c^2w^2}\,\mathrm{erf}(aw)\,\mathrm{erf}(bw)dw=\frac{1}{2}\int_0^t \frac{e^{-c^2/u}}{u^{3/2}}\mathrm{erf}\left(\frac{a}{\sqrt{u}}\right)\mathrm{erf}\left(\frac{b}{\sqrt{u}}\right)du,\quad T=\frac{1}{\sqrt{t}}}$$

We work with erfc functions by writing

$$\mathrm{erf}(aw)\mathrm{erf}(bw) = 1\text{-}\mathrm{erfc}(aw)\text{-}\mathrm{erfc}(bw)\text{+}\mathrm{erfc}(aw)\mathrm{erfc}(bw)$$

and using I_5 from Folder 5, we have

$$I_3(a,b,c,T)=\frac{\sqrt{\pi}}{2c}\mathrm{erfc}(cT)-I_5(c,a,T)-I_5(c,b,T)+J_3(a,b,c,T)$$

where

$$J_3(a,b,c,T)=\int_T^\infty e^{-c^2w^2}\,\mathrm{erfc}(aw)\,\mathrm{erfc}(bw)dw$$

For the analysis, we take the parameters a and b so that $a = \min(a,b)$ and $b = \max(a,b)$. Then we have 3 relations for the placement of c relative to a and b:

$c \le a \le b,\qquad a \le c \le b,\qquad a \le b \le c.$

<u>CASE I ($a,c \le b$)</u> ($c \le a \le b$ or $a \le c \le b$)

We integrate J_3 by parts:

$u = \mathrm{erfc}(aw) \qquad dv = e^{-c^2w^2}\,\mathrm{erfc}(bw)$

$du = \dfrac{-2a}{\sqrt{\pi}}e^{-a^2w^2}dw \qquad v = -I_5(c,b,w)$

and

$$J_3(a,b,c,T)=\mathrm{erfc}(aT)I_5(c,b,T)-\frac{2a}{\sqrt{\pi}}\int_T^\infty e^{-a^2w^2}I_5(c,b,w)dw$$

where we use (Folder 5)

$$I_5(c,b,w) = \frac{1}{2d\sqrt{\pi}} \sum_{k=0}^{\infty} C_k \left(\frac{c^2}{d^2}\right)^k E_{k+3/2}(d^2 w^2), \qquad d^2 = b^2 + c^2$$

$$C_k = (1/2)_k / k!$$

to obtain rapid convergence, $O(k^{-1/2} \cdot 2^{-k})$ since $c^2/d^2 \le ½$. To start Case II, we note that

$$\frac{d}{dw}[\operatorname{erfc}(aw)\operatorname{erfc}(bw)\operatorname{erfc}(cw)] = \frac{-2}{\sqrt{\pi}}[ae^{-a^2w^2} \operatorname{erfc}(bw)\operatorname{erfc}(cw)$$
$$+ be^{-b^2w^2} \operatorname{erfc}(aw)\operatorname{erfc}(cw) + ce^{-c^2w^2} \operatorname{erfc}(aw)\operatorname{erfc}(bw)]$$

and integration gives

$$\frac{\sqrt{\pi}}{2}\operatorname{erfc}(aT)\operatorname{erfc}(bT)\operatorname{erfc}(cT) = aJ_3(b,c,a,T) + bJ_3(a,c,b,T) + cJ_3(a,b,c,T)$$

which is analogous to

$$\frac{\sqrt{\pi}}{2}\operatorname{erfc}(aT)\operatorname{erfc}(bT) = aI_5(a,b,T) + bI_5(b,a,T)$$

in Folder 5.

<u>CASE II ($a \le b \le c$)</u>

$$J_3(a,b,c,T) = \frac{\sqrt{\pi}}{2c}\operatorname{erfc}(aT)\operatorname{erfc}(bT)\operatorname{erfc}(cT) - \frac{a}{c}J_3(b,c,a,T) - \frac{b}{c}J_3(a,c,b,T)$$

Notice that c is larger than a and b. This is precisely the criteria to compute $J_3(b,c,a,T)$ and $J_3(a,c,b,T)$ by Case I. This relation is analogous to

$$I_5(a,b,T) = \frac{\sqrt{\pi}}{2a}\operatorname{erfc}(aT)\operatorname{erfc}(bT) - \frac{b}{a}I_5(b,a,T)$$

in Folder 5.

To proceed, we consider Case I and develop the series for J_3:

$$J_3(a,b,c,T) = \operatorname{erfc}(aT)I_5(c,b,T) - \frac{2a}{\sqrt{\pi}}\int_T^{\infty} e^{-a^2w^2} I_5(c,b,w)dw = \operatorname{erfc}(aT)I_5(c,b,T) - S(a,b,c,T)$$

where

$$S(a,b,c,T) = \frac{2a}{\sqrt{\pi}} \int_T^\infty e^{-a^2w^2} I_5(c,b,w)dw = \frac{a}{\pi d} \sum_{k=0}^\infty C_k \left(\frac{c^2}{d^2}\right)^k R_k, \qquad d^2 = b^2+c^2$$

$$C_k = (1/2)_k/k!$$

and

$$R_k = \int_T^\infty e^{-a^2w^2} E_{k+3/2}(d^2w^2)dw \quad , \qquad k \geq 0.$$

Now we concentrate on R_k. Using the definition of $E_{k+3/2}$, we obtain

$$R_k = \int_1^\infty \frac{dv}{v^{k+3/2}} \int_T^\infty e^{-(a^2+d^2v)w^2} dw, \qquad d^2(b,c) = b^2+c^2$$

and we write $d(b,c)$ to make the dependence clear for the reparameterization in Case II. Then,

$$R_k = \frac{\sqrt{\pi}}{2} \int_1^\infty \frac{dv}{v^{k+3/2}} \frac{\mathrm{erfc}(T\sqrt{a^2+d^2v})}{\sqrt{a^2+d^2v}}$$

Now,

$$(1-z)^{-a} = \sum_{k=0}^\infty \frac{(a)_k}{k!} z^k \quad , \qquad |z| < 1$$

and with $\quad s = \sqrt{a^2+d^2v}\,, \qquad v = (s^2\text{-}a^2)/d^2, \qquad dv = \dfrac{2sds}{d^2}\,,$

$$R_k = d^{2k+1}\sqrt{\pi} \sum_{n=0}^\infty \frac{(k+3/2)_n}{n!} a^{2n} \int_X^\infty \frac{\mathrm{erfc(Ts)}}{\mathrm{s}^{2\mathrm{n}+2\mathrm{k}+3}} ds$$

$$X^2 = a^2+b^2+c^2, \qquad d^2 = b^2+c^2.$$

Now, we know

$$\int_X^\infty \frac{\mathrm{erfc}(ax)dx}{x^\nu} = \frac{1}{(\nu-1)X^{\nu-1}} \left[\mathrm{erfc}(aX) - \frac{aX}{\sqrt{\pi}} E_{\nu/2}(a^2X^2)\right], \qquad \nu \neq 1$$

$$R_k = \frac{d^{2k+1}\sqrt{\pi}}{X^{2k+2}} \sum_{n=0}^\infty \frac{B(k,n)(a/X)^{2n}}{2n+2k+2} \left[\mathrm{erfc}(TX) - \frac{TX}{\sqrt{\pi}} E_{n+k+3/2}(T^2X^2)\right]$$

where

$$B(k,n) = \frac{(k+3/2)_n}{n!} = \frac{\Gamma(n+k+3/2)}{\Gamma(k+3/2)\Gamma(n+1)}$$

or $$R_k = \frac{\sqrt{\pi}}{2X}\operatorname{erfc}(TX)\left(\frac{d}{X}\right)^{2k+1}\sum_{n=0}^{\infty}\frac{B(k,n)(a/X)^{2n}}{n+k+1}$$

$$-\frac{T}{2}\left(\frac{d}{X}\right)^{2k+1}\sum_{n=0}^{\infty}\frac{B(k,n)(a/X)^{2n}}{n+k+1}E_{n+k+3/2}(T^2X^2)$$

$X^2 = a^2+b^2+c^2$, $\quad d^2 = b^2+c^2$, $\quad B(k,n) = \dfrac{(k+3/2)_n}{n!}$

Analysis of the Series $\sum_{n=0}^{\infty} \frac{B(k,n)}{n+k+1} z^n \equiv U_k(z)$

$$U_k(z) = \sum_{n=0}^{\infty}\frac{(k+3/2)_n}{n!}\frac{z^n}{n+k+1} = \frac{1}{z^{k+1}}\int_0^z \frac{t^k\,dt}{(1-t)^{k+3/2}}. \quad \text{Let } t = zw$$

$$= \int_0^1 w^k(1-zw)^{-k-3/2}\,dw = \frac{\Gamma(1)\Gamma(k+1)}{\Gamma(k+2)}F\left(k+\frac{3}{2},\ k+1,\ k+2,\ z\right)$$

Now, the Kummer relation

$$F(a,b,c,z) = (1-z)^{c-a-b}F(c-a,\ c-b,\ c,\ z)$$

gives

$$F(k+3/2,\ k+1,\ k+2,\ z) = (1-z)^{-k-1/2}F(1/2,\ 1,\ k+2,\ z)$$

$$U_k = \frac{(1-z)^{-k-1/2}}{k+1}F(1/2, 1,\ k+2,\ z)$$

Now, for $k = 0$

$$F(3/2, 1, 2, z) = \frac{1}{\sqrt{1-z}}\cdot\frac{2}{1+\sqrt{1-z}} \qquad \text{HTF Vol. 1 \#6, p. 101}$$

and

$$F(1/2, 1, 2, z) = \frac{2}{1+\sqrt{1-z}},$$

For $k = 1$, differentiation

$$\frac{d}{dz}F(a,b,c,z) = \frac{(a)(b)}{(c)}F(a+1,\, b+1,\, c+1,\, z)$$

gives

$$\frac{(3/2)(1)}{(2)}F(5/2,\, 2,\, 3,\, z) = \frac{d}{dz}\left(\frac{2}{1+\sqrt{1-z}}\cdot\frac{1}{\sqrt{1-z}}\right)$$

and with the Kummer relation on F(5/2, 2, 3, z) we have

$$F(1/2,\, 1,\, 3,\, z) = \frac{4}{3}(1-z)^{3/2}\frac{d}{dz}\left(\frac{2}{1+\sqrt{1-z}}\cdot\frac{1}{\sqrt{1-z}}\right)$$

At this point it seems natural to use the contiguous relation in the third parameter of F to get the recurrence for U_k. But the parameters of $F(1/2, 1, k+2, z)$ are such that a quadratic transformation applies:

$$F(a,\, a+1/2,\, b,\, z) = \frac{2^{2a}}{(1+\sqrt{1-z})^{2a}}F\left(2a,\, 2a-b+1,\, b,\, \frac{1-\sqrt{1-z}}{1+\sqrt{1-z}}\right)$$

and proves to be more fruitful than the recurrence.

Let $a = 1/2$, $b = k+2$, then

$$F(1/2,\, 1,\, k+2,\, z) = \frac{2}{1+\sqrt{1-z}}F\left[1,\, -k,\, k+2,\, \frac{z}{(1+\sqrt{1-z})^2}\right]$$

and this is a simple polynomial of degree k. Let $y = z/(1+\sqrt{1-z})^2$. Then $P_k(y) = F(-k, 1, k+2, y)$ and

$$P_0 = 1, \qquad P_1 = 1-\frac{1}{3}y, \quad P_2 = 1-\frac{2}{4}y+\frac{2\cdot 1}{4\cdot 5}y^2$$

$$P_3 = 1-\frac{3}{5}y+\frac{3\cdot 2}{5\cdot 6}y^2-\frac{3\cdot 2\cdot 1}{5\cdot 6\cdot 7}y^3, \text{ etc.} \quad \text{and}$$

$$U_k(z) = \frac{(1-z)^{-k-1/2}}{k+1}\cdot\frac{2}{1+\sqrt{1-z}}P_k(y), \qquad y = \frac{z}{(1+\sqrt{1-z})^2}$$

In the application,

$z = a^2/X^2, \qquad X^2 = a^2+b^2+c^2, \qquad 1-z = (b^2+c^2)/X^2 = d^2/X^2$

Since the terms of P_k go down rapidly, a test for $|\text{term}| < 10^{-15}$ would be appropriate.

Now the sum on R_k is

$$S = \frac{a}{\pi d}\sum_{k=0}^{\infty} C_k\left(\frac{c^2}{d^2}\right)^k R_k = \frac{a}{\pi d}\left[\frac{\sqrt{\pi}}{2X}\operatorname{erfc}(TX)\sum_{k=0}^{\infty} C_k\left(\frac{c^2}{d^2}\right)^k\left(\frac{d}{X}\right)^{2k+1} U_k\left(\frac{a^2}{X^2}\right)\right.$$
$$\left.-\frac{T}{2}\sum_{k=0}^{\infty} C_k\left(\frac{c^2}{d^2}\right)^k\cdot\left(\frac{d}{X}\right)^{2k+1}\sum_{n=0}^{\infty} B(k,n)\frac{(a/X)^{2n}}{(n+k+1)}E_{n+k+3/2}(T^2X^2)\right]$$

$d^2(b,c) = b^2+c^2, \qquad X^2 = a^2+b^2+c^2, \qquad C_k = (1/2)_k/k! \qquad a,c \le b.$

Then, $S(a,b,c,T)$ becomes

$$S(a,b,c,T) = \frac{a}{\pi d}\left[\frac{d\sqrt{\pi}}{2X^2}\operatorname{erfc}(TX)\sum_{k=0}^{\infty} C_k\left(\frac{c^2}{X^2}\right)^k U_k\left(\frac{a^2}{X^2}\right)\right.$$
$$\left.-\frac{Td}{2X}\sum_{k=0}^{\infty} C_k\left(\frac{c^2}{X^2}\right)^k\sum_{n=0}^{\infty}\frac{\Gamma(n+k+3/2)(a^2/X^2)^n}{\Gamma(k+3/2)\Gamma(n+1)(n+k+1)}E_{n+k+3/2}(T^2X^2)\right]$$

$d^2(b,c) = b^2+c^2, \qquad X^2 = a^2+b^2+c^2, \qquad C_k = (1/2)_k/k! = \dfrac{\Gamma(k+1/2)}{\Gamma(1/2)\Gamma(k+1)}, \qquad a,c \le b.$

We want the series explicitly in E, so we rearrange the sum and let $m = n+k$ and sum the integer coordinates in the (k,n) plane along left sloping diagonals $n = m-k$ with the n intercept equal to m. Then S(a,b,c,T) becomes

$$S = \frac{a}{\pi d}\left[\frac{\sqrt{\pi}\operatorname{erfc}(TX)}{X+d}\sum_{k=0}^{\infty}\frac{C_k}{k+1}\left(\frac{c^2}{d^2}\right)^k P_k(y)\right.$$
$$\left.-\frac{(TX)d}{2X^2\sqrt{\pi}}\sum_{m=0}^{\infty}\frac{\Gamma(m+3/2)}{(m+1)}E_{m+3/2}(T^2X^2)\sum_{k=0}^{m}\frac{(a^2/X^2)^{m-k}(c^2/X^2)^k}{\Gamma(m-k+1)\Gamma(k+1)}\frac{\Gamma(k+1/2)}{\Gamma(k+3/2)}\right]$$

$d^2(b,c) = b^2+c^2, \qquad X^2 = a^2+b^2+c^2, \qquad y = a^2/(X+d)^2, \qquad a,c \le b$

$P_o(y) = 1, \qquad P_k(y) = F(-k, 1, k+2, y), \qquad k \ge 1.$

Finally,

$$S(a,b,c,T) = \frac{a}{\pi d}\left[\frac{\sqrt{\pi}\,\mathrm{erfc}(TX)}{X+d} \sum_{k=0}^{\infty} \frac{(1/2)_k}{(k+1)!}\left(\frac{c^2}{d^2}\right)^k P_k(y) \right.$$
$$\left. - \frac{(TX)d}{2X^2} \sum_{m=0}^{\infty} \frac{(3/2)_m}{(m+1)!} E_{m+3/2}(T^2X^2) \sum_{k=0}^{m} \binom{m}{k} \frac{(a^2/X^2)^{m-k}(c^2/X^2)^k}{2k+1} \right]$$

$d^2(b,c) = b^2+c^2$, $\quad X^2 = a^2+b^2+c^2$, $\quad y = a^2/(X+d)^2$, $\quad P_o(y) = 1$

$P_k(y) = F(-k, 1, k+2, y)$, $\quad k \geq 1$ $\quad a,c \leq b$.

Analysis of the Series $\sum_{k=0}^{\infty} \frac{C_k}{k+1}\left(\frac{c^2}{d^2}\right)^k P_k(y)$, **$C_k = (1/2)_k/k!$,$a,c \leq b$**

It is clear from the definition of $P_k(y)$ that $P_k(y)$ is bounded uniformly independent of k:

$$\left|P_k(y)\right| < 1 + y + ... = (1-y)^{-1}, \quad y = \frac{a^2}{(X+d)^2} = \frac{a^2}{X^2+d^2+2Xd} < \frac{a^2}{5a^2} = \frac{1}{5}, \quad \left|P_k(y)\right| < \frac{5}{4}$$

by replacing b with a and c with 0 in X and d where $X^2 = a^2+b^2+c^2$ and $d^2 = b^2+c^2$.

Also,

$$\frac{c^2}{d^2} = \frac{c^2}{b^2+c^2} \leq \frac{c^2}{2c^2} = \frac{1}{2}$$

since $c \leq b$ in Case I, and the convergence of the P_k sum is dominated by $O(k^{-3/2} \cdot 2^{-k})$ at worse.

Analysis of the Sum $\sum_{k=0}^{m} \binom{m}{k} \frac{(a^2/X^2)^{m-k}(c^2/X^2)^k}{2k+1}$

Notice that this binomial sum is dominated by $[(a^2+c^2)/X^2]^m$. With the restriction $a,c \leq b$, we have

$$\frac{a^2+c^2}{X^2} = 1 - \frac{b^2}{X^2} \leq 1 - \frac{b^2}{3b^2} = \frac{2}{3}$$

and the convergence of the $E_{m+3/2}$ sum is dominated by $O[m^{-1/2} \cdot (2/3)^m]$

Thus, for an error $O(10^{-14})$ we take approximately 40 terms for the P_k series and 80 terms for the $E_{k+3/2}$ series in the extreme cases. Finally,

$$I_3(a,b,c,T) = \frac{\sqrt{\pi}}{2c}\operatorname{erfc}(cT) - I_5(c,a,T) - I_5(c,b,T) + J_3(a,b,c,T).$$

We can also write $I_3(a,b,c,T)$ in terms of S, thus exposing all of the leading terms,

$$I_3(a,b,c,T) = \frac{\sqrt{\pi}}{2c}\operatorname{erfc}(cT) - I_5(c,a,T) - I_5(c,b,T) +$$

$$+\begin{cases} \operatorname{erfc}(aT)I_5(c,b,T) - S(a,b,c,T) & a,c \le b \\ \frac{\sqrt{\pi}}{2c}\operatorname{erfc}(aT)\operatorname{erfc}(bT)\operatorname{erfc}(cT) - \frac{a}{c}[\operatorname{erfc}(bT)I_5(a,c,T) & \\ & a \le b \le c \\ \qquad - S(b,c,a,T)] - \frac{b}{c}[\operatorname{erfc}(aT)I_5(b,c,T) - S(a,c,b,T)] & . \end{cases}$$

Thus, for Case I, $a,c \le b$, $[a = \min(a,b),\ b = \max(a,b)]$ and for Case II, $a \le b \le c$, $[a = \min(a,b),\ b = \max(a,b)]$. Then, separating the cases, we have

$$I_3(a,b,c,T) = \frac{\sqrt{\pi}}{2c}\operatorname{erfc}(cT) - I_5(c,a,T) - \operatorname{erf}(aT)I_5(c,b,T) - S(a,b,c,T) \quad a,c \le b$$

$$I_3(a,b,c,T) = \frac{\sqrt{\pi}}{2c}\operatorname{erfc}(cT) - I_5(c,a,T) - I_5(c,b,T)$$

$$a \le b \le c$$

$$+\frac{\sqrt{\pi}}{2c}\operatorname{erfc}(aT)\operatorname{erfc}(bT)\operatorname{erfc}(cT) - \frac{a}{c}\operatorname{erfc}(bT)I_5(a,c,T)$$
$$-\frac{b}{c}\operatorname{erfc}(aT)I_5(b,c,T) + \frac{a}{c}S(b,c,a,T) + \frac{b}{c}S(a,c,b,T).$$

Notice that there is potential for losses of significance when T is small and the analytic expressions from Folder 5 ought to be included to analytically eliminate these losses. The case for T actually equal to zero is treated in part 7b of this Folder. The analytic reductions would use the relations $\tan^{-1}x+\tan^{-1}(1/x) = \pi/2$, $\operatorname{erf}(x)+\operatorname{erfc}(x) = 1$.

Folder 7b

Power Series for Small *aT*, *bT* and *cT*

We simply use the power series for each of the terms in the product $e^{-c^2w^2}\,\mathrm{erf}(aw)\,\mathrm{erf}(bw)$ and integrate:

$$\mathrm{erf}(aw) = \frac{2aw}{\sqrt{\pi}}\sum_{k=0}^{\infty} C_k(a^2)w^{2k}; \qquad \mathrm{erf}(bw) = \frac{2bw}{\sqrt{\pi}}\sum_{k=0}^{\infty} C_k(b^2)w^{2k}$$

and $$\mathrm{erf}(aw)\,\mathrm{erf}(bw) = \frac{4abw^2}{\pi}\sum_{k=0}^{\infty} U_k(a^2,b^2)w^{2k}$$

where $U_k(a^2,b^2) = \sum_{m=0}^{k} C_m(a^2)C_{k-m}(b^2), \quad C_k(a^2) = \dfrac{(-1)^k a^{2k}}{k!(2k+1)}$

Now, $e^{-c^2w^2} = \sum_{k=0}^{\infty} D_k(c^2)w^{2k}, \qquad D_k(c^2) = \dfrac{(-1)^k c^{2k}}{k!}$

and $e^{-c^2w^2}\,\mathrm{erf}(aw)\,\mathrm{erf}(bw) = \dfrac{4ab}{\pi}\sum_{k=0}^{\infty} V_k w^{2k+2}$ where $V_k = \sum_{n=0}^{k} D_n(c^2)U_{k-n}(a^2,b^2)$

Now integrate to get

$$\int_0^T e^{-c^2w^2}\,\mathrm{erf}(aw)\,\mathrm{erf}(bw)\,dw = \frac{4abT^3}{\pi}\sum_{k=0}^{\infty}\frac{V_kT^{2k}}{2k+3}$$

Now we distribute T^{2k} among the terms of V_k to achieve better scaling:

$$V_kT^{2k} = \sum_{n=0}^{k} D_n(c^2T^2)U_{k-n}(a^2T^2,b^2T^2)$$

and

$$U_k(a^2T^2,b^2T^2) = \sum_{m=0}^{k} C_m(a^2T^2)C_{k-m}(b^2T^2)$$

Now, we apply the series on $(0,T)$ to get the integral on (T,∞):

$$I_3(a,b,c,T) = I_3(a,b,c,0) - \int_0^T e^{-c^2w^2}\,\mathrm{erf}(aw)\,\mathrm{erf}(bw)\,dw$$

where $I_3(a,b,c,0)$ can be computed from Cases I & II. Numerical experiments show satisfactory errors on the order of $O(10^{-12})$ or better in double precision arithmetic for aT, bT, and cT all less than or equal to 2. Some refinements are needed for $I_3(a,b,c,0)$ to achieve significant digits:

For both Cases I and II we have:

$$I_3(a,b,c,T) = \frac{\sqrt{\pi}}{2c}\operatorname{erf}(cT) - I_5(c,a,T) - I_5(c,b,T) + J_3(a,b,c,T)$$

<u>Case I for $I_3(a,b,c,0)$</u> $\qquad a,c \le b$

For $T = 0$,

$$I_5(c,a,0) = \frac{1}{c\sqrt{\pi}}\tan^{-1}\frac{c}{a}, \qquad I_5(c,b,0) = \frac{1}{c\sqrt{\pi}}\tan^{-1}\frac{c}{b}$$

$$J_3(a,b,c,T) = \operatorname{erfc}(aT) I_5(c,b,T) - S(a,b,c,T)$$

$$J_3(a,b,c,0) = I_5(c,b,0) - S(a,b,c,0)$$

$$I_3(a,b,c,0) = \frac{\sqrt{\pi}}{2c} - \frac{1}{c\sqrt{\pi}}\tan^{-1}\frac{c}{a} - S(a,b,c,0)$$

$$= \frac{1}{c\sqrt{\pi}}\left[\frac{\pi}{2} - \tan^{-1}\frac{c}{a}\right] - S(a,b,c,0)$$

$$I_3(a,b,c,0) = \frac{1}{c\sqrt{\pi}}\tan^{-1}\frac{a}{c} - S(a,b,c,0)$$

<u>Case II for $I_3(a,b,c,0)$</u> $\qquad a \le b \le c$

$$J_3(a,b,c,T) = \frac{\sqrt{\pi}}{2c}\operatorname{erfc}(aT)\operatorname{erfc}(bT)\operatorname{erfc}(cT) - \frac{a}{c}J_3(b,c,a,T) - \frac{b}{c}J_3(a,c,b,T)$$

Since c is dominant we use the permuted parameters in Case I for J_3:

$$J_3(a,b,c,0) = \frac{\sqrt{\pi}}{2c} - \frac{a}{c}[I_5(a,c,0) - S(b,c,a,0)] - \frac{b}{c}[I_5(b,c,0) - S(a,c,b,0)]$$

and $$J_3(a,b,c,0) = \frac{\sqrt{\pi}}{2c} - \frac{a}{c}\cdot\frac{1}{a\sqrt{\pi}}\tan^{-1}\frac{a}{c} - \frac{b}{c}\cdot\frac{1}{b\sqrt{\pi}}\tan^{-1}\frac{b}{c} + \frac{a}{c}S(b,c,a,0) + \frac{b}{c}S(a,c,b,0)$$

$$J_3(a,b,c,0) = \frac{1}{c\sqrt{\pi}}\left[\frac{\pi}{2} - \tan^{-1}\frac{a}{c} - \tan^{-1}\frac{b}{c}\right] + \frac{a}{c}S(b,c,a,0) + \frac{b}{c}S(a,c,b,0)$$

$$I_3(a,b,c,0) = \frac{1}{c\sqrt{\pi}}\left[\frac{\pi}{2} - \tan^{-1}\frac{c}{a} - \tan^{-1}\frac{c}{b}\right] + J_3(a,b,c,0)$$

$$= \frac{1}{c\sqrt{\pi}}\left[\tan^{-1}\frac{a}{c} - \tan^{-1}\frac{c}{b}\right] + \frac{1}{c\sqrt{\pi}}\left[-\tan^{-1}\frac{a}{c} + \tan^{-1}\frac{c}{b}\right] + \frac{a}{c}S(b,c,a,0) + \frac{b}{c}S(a,c,b,0)$$

$$I_3(a,b,c,0) = \frac{a}{c}S(b,c,a,0) + \frac{b}{c}S(a,c,b,0)$$

Notice that the computation of $S(a,b,c,0)$ and all permutations of a,b,c needed here involve only the $P_k(y)$ sum since the coefficient of the $E_{m+3/2}$ sum is zero. Furthermore the $P_k(y)$ sum contains all positive terms and no significance is lost in the computation. Explicitly,

$$S(a,b,c,0) = \frac{a}{d\sqrt{\pi}} \cdot \frac{1}{X+d}\sum_{k=0}^{\infty} \frac{(1/2)_k}{(k+1)!}\left(\frac{c^2}{d^2}\right)^k P_k(y) = \frac{2a}{\sqrt{\pi}}\int_0^{\infty} e^{-a^2w^2} I_5(c,b,w)dw$$

where $d^2 = b^2+c^2$, $X^2 = a^2+b^2+c^2$, $y = a^2/(X+d)^2$, $P_0(y) = 1$, $P_k(y) = F(-k, 1, k+2, y)$, $k \geq 1$ for $a,c \leq b$ and $a = \min(a,b)$, $b = \max(a,b)$. In Case II, c takes the place of b as the dominant parameter.

Folder 7c

Series Expansion for Large c

We start by using the series expansions for erf(aw) and erf(bw) and multiplying them to obtain

$$\operatorname{erf}(aw)\operatorname{erf}(bw) = \frac{4ab}{\pi}\sum_{k=0}^{\infty} U_k(a^2,b^2)w^{2k+2}$$

and integration with the exponential gives

$$\int_T^{\infty} e^{-c^2w^2}\operatorname{erf}(aw)\operatorname{erf}(bw)dw = \frac{4ab}{\pi}\sum_{k=0}^{\infty} U_k(a^2,b^2)\int_T^{\infty} w^{2k+2}e^{-c^2w^2}dw$$

where

$$U_k(a^2,b^2) = \sum_{m=0}^{k} \frac{(-a^2)^m}{m!(2m+1)}\cdot\frac{(-b^2)^{k-m}}{(k-m)![2(k-m)+1]} = \frac{(-1)^k}{k!}\sum_{m=0}^{k}\binom{k}{m}\frac{(a^2)^m}{2m+1}\cdot\frac{(b^2)^{k-m}}{2(k-m)+1}$$

In the integral, let $c^2w^2 = u$, $w = \sqrt{u}/c$ and

$$\int_T^{\infty} w^{2k+2}e^{-c^2w^2}dw = \frac{1}{2c^{2k+3}}\int_{c^2T^2}^{\infty} e^{-u}u^{k+1/2}du = \frac{1}{2c^{2k+3}}\Gamma(k+3/2,\, c^2T^2)$$

Now, this incomplete gamma function has a two-term recurrence relation in which forward recurrence is numerically stable: Let $Y_k(x) = \Gamma(k+1/2, x)/\Gamma(k+1/2)$. Then

$$Y_{k+1} = Y_k + \frac{e^{-x}x^{k+1/2}}{\Gamma(k+3/2)} = Y_k + T_k \qquad k = 0,1,\ldots$$

where $T_{k+1} = T_k\cdot\dfrac{x}{(k+3/2)}$ with $T_o = \dfrac{e^{-x}\sqrt{x}}{(\sqrt{\pi}/2)}$

and $$Y_o = \frac{\Gamma(1/2,x)}{\Gamma(1/2)} = \frac{1}{\sqrt{\pi}}\int_x^{\infty} e^{-u}u^{-1/2}du = \frac{2}{\sqrt{\pi}}\int_{\sqrt{x}}^{\infty} e^{-v^2}dv = \operatorname{erfc}(\sqrt{x})$$

Then,

$$\int_T^{\infty} e^{-c^2w^2}\operatorname{erf}(aw)\operatorname{erf}(bw)dw = \frac{2ab}{\pi c^3}\sum_{k=0}^{\infty} Y_{k+1}(c^2T^2)U_k\left(\frac{a^2}{c^2},\frac{b^2}{c^2}\right)\Gamma(k+3/2)$$

$$= \frac{2ab}{\pi c^3} \sum_{k=0}^{\infty} (-1)^k Y_{k+1}(c^2T^2) \frac{\Gamma(k+3/2)}{\Gamma(k+1)} \sum_{m=0}^{k} \binom{k}{m} \frac{(a^2/c^2)^m}{2m+1} \cdot \frac{(b^2/c^2)^{k-m}}{2(k-m)+1}$$

Notice that the binomial sum is dominated by $[(a^2+b^2)/c^2]^k/(2k+1)$ because the quadratic $[2m+1][2(k\text{-}m)+1]$ has a maximum near $m = [k/2]$ and minima at $m = 0$ and $m = k$. So, to obtain terms which decrease rapidly and avoid losses of significance, we want $(a^2+b^2)/c^2 \leq ½$. Notice also that $\Gamma(k+3/2)Y_k(c^2T^2) \sim (c^2T^2)^{k+3/2} e^{-c^2T^2}$ can get relatively large if c^2T^2 is large even with k factorial in the denominator. Therefore we modify the condition from

$$(a^2+b^2)/c^2 \leq 1/2 \qquad \text{to} \qquad (a^2+b^2)/c^2 \leq 1/(2p)$$

to counteract the buildup of powers in c^2T^2 in $\Gamma(k+3/2)Y_k(c^2T^2)$ where p was determined experimentally to be

$$p = \begin{cases} 1 & \text{for} \quad c^2T^2 \leq 5 \\ c^2T^2 - 4 & \text{for} \quad c^2T^2 > 5 \end{cases}$$

Thus, a simple formula with recurrences for the major terms can be used for parameters satisfying

$$\frac{a^2+b^2}{c^2} \leq \frac{1}{2p}$$

Notice also that the binomial coefficients can be generated very rapidly and efficiently by additions using the Pascal triangle relation $B_{k+1,m} = B_{k,m-1} + B_{k,m}$, $m = k,\ 0$, $B_{k+1,k+1} = 1$. Only one storage vector need be used if this relation is applied by backward recurrence on m since the (k+1,m) value can be stored in the (k,m) position without destroying data needed for the next smaller m.

Folder 7d

$$\textbf{Quadrature for } \int_T^\infty e^{-c^2w^2}\,\mathrm{erf}(aw)\,\mathrm{erf}(bw)dw = \frac{1}{2}\int_0^t \frac{e^{-c^2/u}}{u^{3/2}}\,\mathrm{erf}\left(\frac{a}{\sqrt{u}}\right)\mathrm{erf}\left(\frac{b}{\sqrt{u}}\right)du$$

Now
$$I_3 = \int_T^\infty e^{-c^2w^2}\,\mathrm{erf}(aw)\,\mathrm{erf}(bw)dw, \qquad T = 1/\sqrt{t}$$

For the quadrature, we follow the procedure of Folder 3 by replacing the erf functions with 1 when the arguments exceed 6. As noted, the error is uniformly $O(10^{-15})$ for erf(x) = 1 when $x \geq 6$. Then, we have the same cases as in Folder 3:

Let $\quad W_m = \min\left(\frac{6}{a},\frac{6}{b}\right), \qquad W_M = \max\left(\frac{6}{a},\frac{6}{b}\right),$

and let $X = \min(a,b)$:

Case I: $\qquad 1/\sqrt{t} = T \leq W_m$

$$I_3 = \int_T^{W_m} e^{-c^2w^2}\,\mathrm{erf}(aw)\,\mathrm{erf}(bw)dw + \int_{W_m}^{W_M} e^{-c^2w^2}\,\mathrm{erf}(Xw)dw + \int_{W_M}^\infty e^{-c^2w^2}dw$$

Case II: $\qquad W_m < T \leq W_M$

$$I_3 = \int_T^{W_M} e^{-c^2w^2}\,\mathrm{erf}(Xw)dw + \int_{W_M}^\infty e^{-c^2w^2}dw$$

Case III: $\qquad W_M < T < \infty$

$$I_3 = \int_T^\infty e^{-c^2w^2}dw$$

Now, notice that the integrals in Cases I and II are of the form

$$R\,(c,X,A,B) = \int_A^B e^{-c^2w^2}\,\mathrm{erf}(Xw)dw = J_5(c,X,A) - J_5(c,X,B)$$

which can be computed from $J_5(a,b,z)$, (Folder 5). In terms of R, we get

Case I: $1/\sqrt{t} = T \le W_m$

$$I_3(a,b,c,T) = \int_T^{W_m} e^{-c^2w^2}\,\mathrm{erf}(aw)\,\mathrm{erf}(bw)\,dw + R(c,X,W_m,W_M) + \frac{\sqrt{\pi}}{2c}\mathrm{erfc}(cW_M)$$

Case II: $W_m < T \le W_M$

$$I_3(a,b,c,T) = R(c,X,T,W_M) + \frac{\sqrt{\pi}}{2c}\mathrm{erfc}(cW_M)$$

Case III: $W_M < T < \infty$

$$I_3(a,b,c,T) = \frac{\sqrt{\pi}}{2c}\mathrm{erfc}(cT)$$

Combining the $\sqrt{\pi}\,\mathrm{erfc}(cW_M)/(2c)$ term with $R(c,X,A,W_M)$ and using the relation

$$\frac{\sqrt{\pi}}{2c}\,erfc(cW_M) - J_5(c,X,W_M) = I_5(c,X,W_M)$$

gives all positive terms and eliminates losses of significance by subtraction:

Case I: $1/\sqrt{t} = T \le W_m$

$$I_3(a,b,c,T) = \int_T^{W_m} e^{-c^2w^2}\,\mathrm{erf}(aw)\,\mathrm{erf}(bw)\,dw + J_5(c,X,W_m) + I_5(c,X,W_M)$$

Case II: $W_m < T \le W_M$

$$I_3(a,b,c,T) = J_5(c,X,T) + I_5(c,X,W_M)$$

Case III: $W_M < T < \infty$

$$I_3(a,b,c,T) = \frac{\sqrt{\pi}}{2c}\mathrm{erfc}(cT)$$

If c varies widely, then the stepping procedure described in Section (1.1) for DQUAD8 should be applied on $[[1/\sqrt{t} = T \le W_m]$ to reflect the scale of integration. An appropriate step for would be $\sigma = \min\{W_m, 6/c\}/2$. For $I_3^c(a,b,c,T)$, a quadrature with DQUAD8 and $\sigma = 3/\sqrt{a^2+b^2+c^2}$, is also used in PROGRAM I3COMP from file BECKDRVR.FOR to compare formulas.

Folder 7e

Special Cases

$c = 0$ (See Folder 9 for $\int_0^T \operatorname{erf}(aw)\operatorname{erf}(bw)dw$)

$$J_3(a,b,0,T) = \int_T^\infty \operatorname{erfc}(aw)\operatorname{erfc}(bw)dw$$

Integrate by parts: $u = \operatorname{erfc}(aw)$ $dv = \operatorname{erfc}(bw)dw$

$$du = \frac{-2a}{\sqrt{\pi}}e^{-a^2w^2}dw \qquad v = -\frac{i\operatorname{erfc}(bw)}{b}$$

$$J_3 = \frac{\operatorname{erfc}(aT)i\operatorname{erfc}(bT)}{b} - \frac{2a}{b\sqrt{\pi}}\int_T^\infty e^{-a^2w^2}i\operatorname{erfc}(bw)dw$$

(This integral is $R_o/(2\sqrt{\pi})$ in the recurrence for R_k).

Now,

$$i\operatorname{erfc}(bw) = -bw\operatorname{erfc}(bw) + \frac{e^{-b^2w^2}}{\sqrt{\pi}}$$

$$J_3 = \frac{\operatorname{erfc}(aT)i\operatorname{erfc}(bT)}{b} - \frac{2a}{b\sqrt{\pi}}\left[-b\int_T^\infty e^{-a^2w^2}w\operatorname{erfc}(bw)dw + \frac{1}{\sqrt{\pi}}\int_T^\infty e^{-(a^2+b^2)w^2}dw\right]$$

Let $u = \operatorname{erfc}(bw)$ $dv = e^{-a^2w^2}wdw$

$$du = \frac{-2b}{\sqrt{\pi}}e^{-b^2w^2}dw \qquad v = -\frac{e^{-a^2w^2}}{2a^2}$$

$$J_3 = \frac{\operatorname{erfc}(aT)i\operatorname{erfc}(bT)}{b} + \frac{2a}{\sqrt{\pi}}\left[\frac{e^{-a^2T^2}\operatorname{erfc}(bT)}{2a^2} - \frac{b}{a^2\sqrt{\pi}}\int_T^\infty e^{-(a^2+b^2)w^2}dw\right] - \frac{2a}{b\pi}\int_T^\infty e^{-(a^2+b^2)w^2}dw$$

$$= \frac{\operatorname{erfc}(aT)i\operatorname{erfc}(bT)}{b} + \frac{e^{-a^2T^2}\operatorname{erfc}(bT)}{a\sqrt{\pi}} - \left(\frac{2b}{a\pi} + \frac{2a}{b\pi}\right)\frac{\operatorname{erfc}(T\sqrt{a^2+b^2})}{\sqrt{a^2+b^2}}\cdot\frac{\sqrt{\pi}}{2}$$

$$= \frac{\operatorname{erfc}(aT)i\operatorname{erfc}(bT)}{b} + \frac{e^{-a^2T^2}\operatorname{erfc}(bT)}{a\sqrt{\pi}} - \frac{\sqrt{a^2+b^2}}{ab\sqrt{\pi}}\operatorname{erfc}(T\sqrt{a^2+b^2})$$

$$J_3 = \frac{e^{-a^2T^2}\,\mathrm{erfc}(bT)}{a\sqrt{\pi}} + \frac{e^{-b^2T^2}\,\mathrm{erfc}(aT)}{b\sqrt{\pi}} - T\,\mathrm{erfc}(bT)\,\mathrm{erfc}(aT) - \frac{\sqrt{a^2+b^2}}{ab\sqrt{\pi}}\mathrm{erfc}(T\sqrt{a^2+b^2})$$

Finally for $c = 0$, $J_3(a,b,0,T) = \int_T^\infty \mathrm{erfc}(aw)\,\mathrm{erfc}(bw)dw$

$$J_3(a,b,0,T) = \frac{e^{-a^2T^2}\,\mathrm{erfc}(bT)}{a\sqrt{\pi}} + \frac{e^{-b^2T^2}\,\mathrm{erfc}(aT)}{b\sqrt{\pi}} - T\,\mathrm{erfc}(aT)\,\mathrm{erfc}(bT) - \frac{\sqrt{a^2+b^2}}{ab\sqrt{\pi}}\mathrm{erfc}(T\sqrt{a^2+b^2})$$

We also have a rapidly convergent series form for c=0, which resolves the indeterminacy in the previous relation for a to zero,

$$J_3(a,b,0,T) = \mathrm{erfc}(aT)ierfc(bT)/b - S(a,b,0,T)$$

$$S(a,b,0,T) = \frac{a}{\pi b}\left[\frac{\sqrt{\pi}\,\mathrm{erfc}(TX)}{X+b} - \frac{bT}{2X}\sum_{m=0}^{\infty}\frac{(3/2)_m}{(m+1)!}\left(\frac{a^2}{a^2+b^2}\right)^m E_{m+3/2}(T^2X^2)\right]$$

$$X^2 = a^2+b^2, \qquad a \le b.$$

Limiting case for $a = 0$ (and similarly for $b = 0$ due to symmetry)

$$J_3(0,b,0,T) = \int_T^\infty \mathrm{erfc}(bw)dw = \frac{i\,\mathrm{erfc}(bT)}{b}$$

$$e^{-a^2T} = 1 - a^2T + \ldots\,, \qquad \sqrt{a^2+b^2} = b\sqrt{1+\frac{a^2}{b^2}} = b\left(1+\frac{1}{2}\frac{a^2}{b^2}+\ldots\right)$$

for $a \to 0$ we get

$$\frac{e^{-a^2T^2}\,\mathrm{erfc}(bT)}{a\sqrt{\pi}} - \frac{\sqrt{a^2+b^2}}{ab\sqrt{\pi}}\mathrm{erfc}(T\sqrt{a^2+b^2}) \to$$

$$\left(\frac{1}{a} - aT^2\ldots\right)\frac{\mathrm{erfc}(bT)}{\sqrt{\pi}} - \frac{b}{ab\sqrt{\pi}}\left(1+\frac{1}{2}\frac{a^2}{b^2}\ldots\right)\mathrm{erfc}(bT) \to 0$$

and we get

$$J_3(0,b,0,T) = \frac{e^{-b^2T}}{b\sqrt{\pi}} - T\,\mathrm{erfc}(bT) = \frac{i\,\mathrm{erfc}(bT)}{b}$$

also, $$J_3(a,b,0,0) = \frac{1}{a\sqrt{\pi}} + \frac{1}{b\sqrt{\pi}} - \frac{\sqrt{a^2+b^2}}{ab\sqrt{\pi}} = \frac{a+b-\sqrt{a^2+b^2}}{ab\sqrt{\pi}}$$

and $$J_3(a,b,0,0) = \int_0^\infty \operatorname{erfc}(aw)\operatorname{erfc}(bw)dw = \frac{(2/\sqrt{\pi})}{a+b+\sqrt{a^2+b^2}}$$

This also checks with $J_3(0,b,0,T)$ for $T = 0$:

$$J_3(0,b,0,0) = \frac{1}{b\sqrt{\pi}} \text{ since } i\operatorname{erfc}(0) = 1/\sqrt{\pi}$$

Also, if $a = 0$, we have another representation of $I_5(c,b,T)$: $J_3(0,b,c,T) = I_5(c,b,T)$ from the definition of $J_3(a,b,c,T)$.

Folder 7f

Fine Tuning of Cases I and II for Numerical Evaluation

Thus far, we have written I_3 in terms of J_3:

$$I_3(a,b,c,T) = \frac{\sqrt{\pi}}{2c}\operatorname{erfc}(cT) - I_5(c,a,T) - I_5(c,b,T) + J_3(a,b,c,T).$$

However, in both cases I and II, J_3 has terms which can combine with those above to provide significant digits where only absolute or decimal place accuracy is possible with direct computation. For example, it is apparent from the definition that $I_3(0,b,c,T) = 0$ but this is not apparent from this formula and the formulae for J_3.

Case I, $c \le a \le b$

For this case, Folder 7a [page (7a-2)] gives

$$J_3(a,b,c,T) = \operatorname{erfc}(aT)I_5(c,b,T) - S(a,b,c,T)$$

where $S(a,b,c,T)$ is defined on page (7a-3) and computed on page (7a-7).

We manipulate the I_5 functions as we did in Folder 5 by splitting out the dominant term for small T. Let $\alpha^2 = a^2 + c^2$. Then from Folder 5, we have for $c \le a$

$$I_5(c,a,T) = \frac{e^{-\alpha^2T^2}}{c\sqrt{\pi}}\tan^{-1}\frac{c}{a} - \frac{\alpha T^2}{2\sqrt{\pi}}\tilde{S}(c,a,T), \qquad \alpha^2 = a^2 + c^2, \qquad c \le a$$

where $\tilde{S}$ is the S function of Folder 5. The result is

$$I_3(a,b,c,T) = \begin{cases} \left[\dfrac{1}{c\sqrt{\pi}}\tan^{-1}\dfrac{a}{c} - \dfrac{\sqrt{\pi}}{2c}\operatorname{erf}(cT) + \dfrac{\phi(\alpha^2T^2)}{c\sqrt{\pi}}\tan^{-1}\dfrac{c}{a} + \dfrac{\alpha T^2}{2\sqrt{\pi}}\tilde{S}(c,a,T)\right] & \\ \qquad - \operatorname{erf}(aT)I_5(c,b,T) - S(a,b,c,T) & \alpha^2T^2 \le 3 \\ \left[\dfrac{\sqrt{\pi}}{2c}\operatorname{erfc}(cT) - I_5(c,a,T)\right] - \operatorname{erf}(aT)I_5(c,b,T) - S(a,b,c,T) & \alpha^2T^2 > 3 \end{cases}$$

where $\phi(z) = 2e^{-z/2}\sinh(z/2)$, $\qquad \alpha^2 = a^2 + c^2$.

Case I, $a \le c \le b$

Again,

$$J_3(a,b,c,T) = \text{erfc}(aT)I_5(c,b,T) - S(a,b,c,T)$$

Let $\beta^2 = b^2 + c^2$. Then, as in the previous case, we have from Folder 5 for $c \ge a$ and $c < b$,

$$I_5(c,a,T) = \frac{\sqrt{\pi}}{2c}\text{erfc}(aT)\,\text{erfc}(cT) - \frac{a}{c}I_5(a,c,T) \qquad c \ge a$$

$$I_5(c,b,T) = \frac{e^{-\beta^2T^2}}{c\sqrt{\pi}}\tan^{-1}\frac{c}{b} - \frac{\beta T^2}{2\sqrt{\pi}}\tilde{S}(c,b,T) \qquad c < b$$

and I_3 above becomes

$$I_3(a,b,c,T) = \begin{cases} \left[\dfrac{1}{c\sqrt{\pi}}\tan^{-1}\dfrac{b}{c} - \dfrac{\sqrt{\pi}}{2c}\text{erf}(cT) + \dfrac{\phi(\beta^2T^2)}{c\sqrt{\pi}}\tan^{-1}\dfrac{c}{b} + \dfrac{\beta T^2}{2\sqrt{\pi}}\tilde{S}(c,b,T)\right]\text{erf}(aT) & \\ \qquad\qquad + \dfrac{a}{c}I_5(a,c,T) - S(a,b,c,T) & \beta^2T^2 \le 3 \\ \left[\dfrac{\sqrt{\pi}}{2c}\text{erfc}(cT) - I_5(c,b,T)\right]\text{erf}(aT) + \dfrac{a}{c}I_5(a,c,T) - S(a,b,c,T) & \beta^2T^2 > 3 \end{cases}$$

Notice that $I_3(0,b,c,T) = 0$ is apparent since $S(0,b,c,T) = 0$.

In this case, the form for $\beta^2T^2 \le 3$ not only shows the behavior for small T, but eliminates the loss of significance in

$$D = \frac{\sqrt{\pi}}{2c}\text{erfc}(cT) - I_5(c,b,T) \sim \frac{\sqrt{\pi}}{2c} - \frac{1}{c\sqrt{\pi}}\tan^{-1}\frac{c}{b} = \frac{1}{c\sqrt{\pi}}\left[\frac{\pi}{2} - \tan^{-1}\frac{c}{b}\right] = \frac{1}{c\sqrt{\pi}}\tan^{-1}\frac{b}{c}$$

when c is large and cT small.

Case II, $a \le b \le c$

In this case, the reflexive equation is (page 7a-2)

$$J_3(a,b,c,T) = \frac{\sqrt{\pi}}{2c}\mathrm{erfc(aT)}\,\mathrm{erfc}(bT)\,\mathrm{erfc}(cT) - \frac{a}{c}J_3(b,c,a,T) - \frac{b}{c}J_3(a,c,b,T).$$

For this set of parameters,

$$J_3(b,c,a,T) = \mathrm{erfc}\ (bT)I_5(a,c,T) - S(b,c,a,T)$$

$$J_3(a,c,b,T) = \mathrm{erfc}\ (aT)I_5(b,c,T) - S(a,c,b,T)$$

and from Folder 5 for $c > a$ and $c > b$,

$$I_5(c,a,T) = \frac{\sqrt{\pi}}{2c}\mathrm{erfc}(aT)\,\mathrm{erfc}(cT) - \frac{a}{c}I_5(a,c,T) \qquad c \geq a$$

$$I_5(c,b,T) = \frac{\sqrt{\pi}}{2c}\mathrm{erfc}(bT)\,\mathrm{erfc}(cT) - \frac{b}{c}I_5(b,c,T) \qquad c \geq b.$$

Then the first equation in this Folder ($a \leq b \leq c$) becomes

$$I_3(a,b,c,T) = \frac{\sqrt{\pi}}{2c}\mathrm{erf}(aT)\,\mathrm{erf}(bT)\,\mathrm{erfc}(cT) + \frac{b}{c}\mathrm{erf}(aT)I_5(b,c,T) + \frac{a}{c}\mathrm{erf}(bT)I_5(a,c,T) + \frac{a}{c}S(b,c,a,T) + \frac{b}{c}S(a,c,b,T)$$

Notice also that $I_3(0,b,c,T) = 0$ is apparent since $S(0,c,b,T) = 0$. These expressions in Cases I and II also agree with those in Folder 7b for $T = 0$.

Folder 7A

APPENDIX

Inequalities

We derived the relation

$$W \equiv \frac{\sqrt{\pi}}{2}\operatorname{erfc}(aT)\operatorname{erfc}(bT)\operatorname{erfc}(cT) = aJ_3(b,c,a,T) + bJ_3(a,c,b,T) + cJ_3(a,b,c,T)$$

Following the derivation in the APPENDIX of Folder 5, divide both sides by $a+b+c$ and we get a convex linear combination of the J_3's. Then

$$\min[J_3(b,c,a,T), J_3(a,c,b,T), J_3(a,b,c,T)] \le \frac{W}{a+b+c} \le \max[J_3(b,c,a,T), J_3(a,c,b,T), J_3(a,b,c,T)]$$

Also, if we divide by $\sqrt{a^2+b^2+c^2}$ we have the dot product of two vectors on the right, one of which has length 1. By the Cauchy Inequality, we get

$$\left(\frac{W}{\sqrt{a^2+b^2+c^2}}\right)^2 \le J_3^2(b,c,a,T) + J_3^2(a,c,b,T) + J_3^2(a,b,c,T)$$

Notice that if $c = b = a$, then the max and min are the same and $J_3(a,a,a,T) = \dfrac{W}{3a} = \dfrac{\sqrt{\pi}}{6a}\operatorname{erfc}^3(aT)$ as it should be: $\int_T^\infty e^{-a^2w^2}\operatorname{erfc}^2(aw)dw = \dfrac{\sqrt{\pi}}{6a}\operatorname{erfc}^3(aT)$ since $-\dfrac{2ae^{-a^2w^2}}{\sqrt{\pi}}dw$ is the differential of $\operatorname{erfc}(aw)$.

Folder 8

Evaluation of

$$I_4(a,b,c,T)=\int_T^\infty w^2e^{-c^2w^2}\,\mathrm{erf}(aw)\,\mathrm{erf}(bw)dw=\frac{1}{2}\int_0^t\frac{e^{-c^2/u}}{u^{5/2}}\mathrm{erf}\left(\frac{a}{\sqrt{u}}\right)\mathrm{erf}\left(\frac{b}{\sqrt{u}}\right)du$$

$$I_4^c(a,b,c,T)=\int_T^\infty w^2e^{-c^2w^2}\mathrm{erfc}(aw)\mathrm{erfc}(bw)dw=\frac{1}{2}\int_0^t\frac{e^{-c^2/u}}{u^{5/2}}\mathrm{erfc}\left(\frac{a}{\sqrt{u}}\right)\mathrm{erfc}\left(\frac{b}{\sqrt{u}}\right)du$$

$$J_4(a,b,c,T)=\int_T^\infty w\;e^{-c^2w^2}\mathrm{erf}(aw)\mathrm{erf}(bw)dw=\frac{1}{2}\int_0^t\frac{e^{-c^2/u}}{u^2}\mathrm{erf}\left(\frac{a}{\sqrt{u}}\right)\mathrm{erf}\left(\frac{b}{\sqrt{u}}\right)du$$

$$J_4^c(a,b,c,T)=\int_T^\infty w\;e^{-c^2w^2}\mathrm{erfc}(aw)\mathrm{erfc}(bw)dw=\frac{1}{2}\int_0^t\frac{e^{-c^2/u}}{u^2}\mathrm{erfc}\left(\frac{a}{\sqrt{u}}\right)\mathrm{erfc}\left(\frac{b}{\sqrt{u}}\right)du$$

$$a\ge 0,\quad b\ge 0,\quad c>0,\quad T=\frac{1}{\sqrt{t}},\quad t>0$$

Donald E. Amos, October 2000, October 2001, December,2005

Summary

Folder 8 has four subfolders: Folders 8a-8d.

Folder 8a reduces the computation of $I_4(a,b,c,T)$ *and* $I_4^c(a,b,c,T)$ to the $I_3(a,b,c,T)$ *and* $I_3^c(a,b,c,T)$ functions of Folder 7 by integration by parts.

Folder 8b contains the derivation for I_4 used previously in Folders 3 and 7 for a quadrature procedure where each erf function is replaced by 1 when the argument exceeds 6.

Folder 8c reduces the computation of $J_4(a,b,c,T)$ *and* $J_4^c(a,b,c,T)$ 4 to the J and I functions of Folder 5, denoted in most applications as. J_5 and I_5.

Folder 8d contains a quadrature procedure for J_4 whose manipulations mimic those of Folder 8b.

Folder 8a

Series Expansions for

$$I_4(a,b,c,T) = \int_T^\infty w^2 e^{-c^2w^2}\,\mathrm{erf}(aw)\,\mathrm{erf}(bw)\,dw = \frac{1}{2}\int_0^t \frac{e^{-c^2/u}}{u^{5/2}}\,\mathrm{erf}\left(\frac{a}{\sqrt{u}}\right)\mathrm{erf}\left(\frac{b}{\sqrt{u}}\right)du\,,$$

$$I_4^c(a,b,c,T) = \int_T^\infty w^2 e^{-c^2w^2}\,\mathrm{erfc}(aw)\,\mathrm{erfc}(bw)\,dw = \frac{1}{2}\int_0^t \frac{e^{-c^2/u}}{u^{5/2}}\,\mathrm{erfc}\left(\frac{a}{\sqrt{u}}\right)\mathrm{erfc}\left(\frac{b}{\sqrt{u}}\right)du$$

$$T = \frac{1}{\sqrt{t}}$$

The approach here is to reduce I_4 to the computation of I_3 in Folder 7. Integrate by parts gives

$$u = w\,\mathrm{erf}(aw)\,\mathrm{erf}(bw) \qquad dv = we^{-c^2w^2}dw$$

$$du = \left\{ w\left[\frac{2a}{\sqrt{\pi}}e^{-a^2w^2}\,\mathrm{erf}(bw) + \frac{2b}{\sqrt{\pi}}e^{-b^2w^2}\,\mathrm{erf}(aw)\right] + \mathrm{erf}(aw)\,\mathrm{erf}(bw) \right\}dw \qquad v = -\frac{e^{-c^2w^2}}{2c^2}$$

Then,

$$I_4 = \frac{Te^{-c^2T^2}\,\mathrm{erf}(aT)\,\mathrm{erf}(bT)}{2c^2} + \frac{1}{c^2\sqrt{\pi}}\left[a\int_T^\infty e^{-(a^2+c^2)w^2}\,w\,\mathrm{erf}(bw)\,dw + b\int_T^\infty e^{-(b^2+c^2)w^2}\,w\,\mathrm{erf}(aw)\,dw\right]$$

$$+\frac{1}{2c^2}I_3(a,b,c,T) \quad \text{where} \quad I_3 = \int_T^\infty e^{-c^2w^2}\,\mathrm{erf}(aw)\,\mathrm{erf}(bw)\,dw.$$

Now we integrate the integrals with (a^2+c^2) and (b^2+c^2) by parts

$$u = \mathrm{erf}(bw) \qquad dv = we^{-(a^2+c^2)w^2}$$

$$du = \frac{2b}{\sqrt{\pi}}e^{-b^2w^2}dw \qquad v = \frac{-e^{-(a^2+c^2)w^2}}{2(a^2+c^2)}\,.$$

Then,

$$I_4 = \frac{Te^{-c^2T^2}\ \mathrm{erf}(aT)\,\mathrm{erf}(bT)}{2c^2} + \frac{1}{c^2\sqrt{\pi}}\left\{a\left[\frac{e^{-(a^2+c^2)T^2}\ \mathrm{erf}(bT)}{2(a^2+c^2)} + \frac{2b}{\sqrt{\pi}}\int_T^\infty \frac{e^{-X^2w^2}\,dw}{2(a^2+c^2)}\right] + \right.$$

$$\left. +b\left[\frac{e^{-(b^2+c^2)T^2}\ \mathrm{erf}(aT)}{2(b^2+c^2)} + \frac{2a}{\sqrt{\pi}}\int_T^\infty \frac{e^{-X^2w^2}\,dw}{2(b^2+c^2)}\right]\right\} + \frac{1}{2c^2}I_3(a,b,c,T), \qquad X^2 = a^2+b^2+c^2$$

$$I_4 = \frac{Te^{-c^2T^2}\ \mathrm{erf}(aT)\,\mathrm{erf}(bT)}{2c^2} + \frac{1}{2c^2\sqrt{\pi}}\left\{\frac{ae^{-(a^2+c^2)T^2}\ \mathrm{erf}(bT)}{(a^2+c^2)} + \frac{be^{-(b^2+c^2)T^2}\ \mathrm{erf}(aT)}{(b^2+c^2)}\right\}$$

$$+\frac{ab}{2c^2\sqrt{\pi}}\left\{\frac{1}{a^2+c^2} + \frac{1}{b^2+c^2}\right\}\frac{\mathrm{erfc}(TX)}{X} + \frac{1}{2c^2}I_3(a,b,c,T)$$

where $X^2 = a^2+b^2+c^2$. Notice also that $I_4(a,b,c,T) = -\partial I_3(a,b,c,T)/\partial(c^2)$.

Similarly for I_4^c. We integrate by parts

$$u = w\,\mathrm{erfc}(aw)\mathrm{erf}(bw) \qquad\qquad dv = we^{-c^2w^2}\,dw$$

$$du = \left\{w\left[-\frac{2a}{\sqrt{\pi}}e^{-a^2w^2}\,\mathrm{erfc(bw)} - \frac{2b}{\sqrt{\pi}}e^{-b^2w^2}\,\mathrm{erfc(aw)}\right] + \mathrm{erfc(aw)erfc(bw)}\right\}\mathrm{dw} \qquad\qquad v = -\frac{e^{-c^2w^2}}{2c^2}$$

Then,

$$\mathrm{I}_4^c = \frac{\mathrm{Te}^{-c^2T^2}\,\mathrm{erfc(aT)erfc(bT)}}{2c^2} - \frac{1}{c^2\sqrt{\pi}}\left[\mathrm{a}\int_T^\infty \mathrm{e}^{-(a^2+c^2)w^2}\,\mathrm{w\,erfc(bw)dw} + \mathrm{b}\int_T^\infty \mathrm{e}^{-(b^2+c^2)w^2}\,\mathrm{w\,erfc(aw)dw}\right]$$

$$+\frac{1}{2c^2}I_3^c(a,b,c,T) \qquad \text{where}\quad I_3^c = \int_T^\infty e^{-c^2w^2}\,erfc(aw)erfc(bw)dw.$$

Now we integrate the integrals with (a^2+c^2) and (b^2+c^2) by parts

$$u = \mathrm{erfc}(bw) \qquad\qquad dv = we^{-(a^2+c^2)w^2}$$

$$du = -\frac{2b}{\sqrt{\pi}}e^{-b^2w^2}\,dw \qquad\qquad v = \frac{-e^{-(a^2+c^2)w^2}}{2(a^2+c^2)}\ .$$

Then,

$$I_4^c = \frac{Te^{-c^2T^2}\,erfc(aT)erfc(bT)}{2c^2} - \frac{1}{c^2\sqrt{\pi}}\left\{a\left[\frac{e^{-(a^2+c^2)T^2}\,erfc(bT)}{2(a^2+c^2)} - \frac{2b}{\sqrt{\pi}}\int_T^\infty \frac{e^{-X^2w^2}\,dw}{2(a^2+c^2)}\right] + \right.$$

$$\left. +b\left[\frac{e^{-(b^2+c^2)T^2}\,erfc(aT)}{2(b^2+c^2)} - \frac{2a}{\sqrt{\pi}}\int_T^\infty \frac{e^{-X^2w^2}\,dw}{2(b^2+c^2)}\right]\right\} + \frac{1}{2c^2} I_3^c(a,b,c,T), \qquad X^2 = a^2+b^2+c^2$$

$$I_4^c = \frac{Te^{-c^2T^2}\,erfc(aT)erfc(bT)}{2c^2} - \frac{1}{2c^2\sqrt{\pi}}\left\{\frac{ae^{-(a^2+c^2)T^2}\,erfc(bT)}{(a^2+c^2)} + \frac{be^{-(b^2+c^2)T^2}\,erfc(aT)}{(b^2+c^2)}\right\}$$

$$+\frac{ab}{2c^2\sqrt{\pi}}\left\{\frac{1}{a^2+c^2} + \frac{1}{b^2+c^2}\right\}\frac{\mathrm{erfc}(TX)}{X} + \frac{1}{2c^2} I_3^c(a,b,c,T)$$

where $X^2 = a^2+b^2+c^2$. Notice also that $I_4^c(a,b,c,T) = -\partial I_3^c(a,b,c,T)/\partial(c^2)$.

<u>Special Cases</u>

Taking $b \to \infty$ gives

$$\int_T^\infty w^2 e^{-c^2w^2}\,erf(aw)dw = \lim_{b\to\infty} I_4(a,b,c,T) = \frac{1}{2c^2}\left[Te^{-c^2T^2}\,erf(aT) + \frac{a}{a^2+c^2}\frac{e^{-(a^2+c^2)T^2}}{\sqrt{\pi}} + \lim_{b\to\infty} I_3(a,b,c,T)\right]$$

$$\int_T^\infty w^2 e^{-c^2w^2}\,erf(aw)dw = \frac{1}{2c^2}\left[Te^{-c^2T^2}\,erf(aT) + \frac{a}{a^2+c^2}\frac{e^{-(a^2+c^2)T^2}}{\sqrt{\pi}} + J_5(c,a,T)\right]$$

and taking $b \to 0$ gives

$$\int_T^\infty w^2 e^{-c^2w^2}\,erfc(aw)dw = \lim_{b\to 0} I_4^c(a,b,c,T) = \frac{1}{2c^2}\left[Te^{-c^2T^2}\,erfc(aT) - \frac{a}{a^2+c^2}\frac{e^{-(a^2+c^2)T^2}}{\sqrt{\pi}} + \lim_{b\to 0} I_3^c(a,b,c,T)\right]$$

$$\int_T^\infty w^2 e^{-c^2w^2}\,erfc(aw)dw = \frac{1}{2c^2}\left[Te^{-c^2T^2}\,erfc(aT) - \frac{a}{a^2+c^2}\frac{e^{-(a^2+c^2)T^2}}{\sqrt{\pi}} + I_5(c,a,T)\right].$$

Adding these two results gives the check computation

$$\int_T^\infty w^2 e^{-c^2w^2}\,dw = \frac{1}{2c^2}\left[Te^{-c^2T^2} + \frac{\sqrt{\pi}}{2c}erfc(cT)\right].$$

Using the relations erf(x)=1-erfc(x) and erfc(x)=1-erf(x) we also have

$$I_4(a,b,c,T) = \int_T^\infty w^2 e^{-c^2w^2}\,dw - \int_T^\infty w^2 e^{-c^2w^2}\,erfc(aw)dw - \int_T^\infty w^2 e^{-c^2w^2}\,erfc(bw)dw + I_4^c(a,b,c,T)$$

$$I_4^c(a,b,c,T) = \int_T^\infty w^2 e^{-c^2w^2}\,dw - \int_T^\infty w^2 e^{-c^2w^2}\,erf(aw)dw - \int_T^\infty w^2 e^{-c^2w^2}\,erf(bw)dw + I_4(a,b,c,T)$$

where all integrals are computed above.

Folder 8b

Quadrature for

$$\boldsymbol{I_4(a,b,c,T) = \int_T^{\infty} w^2 e^{-c^2 w^2}\,\mathrm{erf}(aw)\,\mathrm{erf}(bw)\,dw = \frac{1}{2}\int_0^t \frac{e^{-c^2/u}}{u^{5/2}}\,\mathrm{erf}\left(\frac{a}{\sqrt{u}}\right)\mathrm{erf}\left(\frac{b}{\sqrt{u}}\right)du\,,\quad T = \frac{1}{\sqrt{t}}}$$

For this quadrature we follow the procedures described in Folders 3 and 7b. This procedure replaces the error functions by 1 when the argument exceeds 6. As noted in Folders 3 and 7b, the error is uniformly $O(10^{-15})$ for the argument larger than 6. Then we have the same cases:

Let $\quad W_m = \min\left(\frac{6}{a},\frac{6}{b}\right)$, $\qquad W_M = \max\left(\frac{6}{a},\frac{6}{b}\right)$ $\qquad$ and

$X = \min(a,b)$

<u>Case I</u>: $\qquad \frac{1}{\sqrt{t}} = T \le W_m$

$$I_4 = \int_T^{W_m} w^2 e^{-c^2w^2}\,\mathrm{erf}(aw)\,\mathrm{erf}(bw)\,dw + \int_{W_m}^{W_M} w^2 e^{-c^2w^2}\,\mathrm{erf}(Xw)\,dw + \int_{W_M}^{\infty} w^2 e^{-c^2w^2}\,dw$$

<u>Case II</u>: $\qquad W_m < T \le W_M$

$$I_4 = \int_T^{W_M} w^2 e^{-c^2w^2}\,\mathrm{erf}(Xw)\,dw + \int_{W_M}^{\infty} w^2 e^{-c^2w^2}\,dw$$

<u>Case III</u>: $\qquad W_M < T < \infty$

$$I_4 = \int_T^{\infty} w^2 e^{-c^2w^2}\,dw$$

Now, define Q by

$$Q = \int_U^{\infty} w^2 e^{-c^2w^2}\,dw = \frac{Ue^{-c^2U^2}}{2c^2} + \frac{\sqrt{\pi}}{4c^2}\frac{\mathrm{erfc}(cU)}{c}$$

and we can integrate $\int_L^U w^2 e^{-c^2w^2}\,\mathrm{erf}(Xw)\,dw$ by parts to obtain a closed form:

Let $\qquad u = w\,\mathrm{erf}(Xw) \qquad\qquad dv = we^{-c^2w^2}\,dw$

$$du = \left[\frac{2Xw}{\sqrt{\pi}}e^{-x^2w^2} + \mathrm{erf}(Xw)\right]dw \qquad\qquad v = \frac{-e^{-c^2w^2}}{2c^2}$$

Define R:

$$R(c,X,L,U)=\int_L^U w^2 e^{-c^2w^2}\operatorname{erf}(Xw)dw$$

$$=\frac{1}{2c^2}\left[Le^{-c^2L^2}\operatorname{erf}(XL)-Ue^{-c^2U^2}\operatorname{erf}(XU)\right]+\frac{X}{c^2\sqrt{\pi}}\int_L^U we^{-(X^2+c^2)w^2}dw+\frac{1}{2c^2}\int_L^U e^{-c^2w^2}\operatorname{erf}(Xw)dw$$

Now this last integral is in the form of $J(c,X,L)$-$J(c,X,U)$ where J is defined and computed in Folder 5. Integration gives

$$R(c,X,L,U)=\frac{1}{2c^2}\left[Le^{-c^2L^2}\operatorname{erf}(XL)-Ue^{-c^2U^2}\operatorname{erf}(XU)\right]$$

$$+\frac{X}{2\sqrt{\pi}c^2(X^2+c^2)}\left[e^{-(X^2+c^2)L^2}-e^{-(X^2+c^2)U^2}\right]+\frac{1}{2c^2}\left[J_5(c,X,L)-J_5(c,X,U)\right].$$

After combining R with Q in Cases I and II and using the relation

$$\frac{\sqrt{\pi}}{2c}erfc(cU)-J_5(c,X,U)=I_5(c,X,U),$$

we get:

<u>Case I</u>: $\frac{1}{\sqrt{t}}=T\le W_m$

$$I_4(a,b,c,T)=\int_T^{W_m} w^2e^{-c^2w^2}\operatorname{erf}(aw)\operatorname{erf}(bw)dw+P(c,X,W_m,W_M)$$

<u>Case II</u>: $W_m < T \le W_M$

$$I_4(a,b,c,T)=P(c,X,T,W_M)$$

<u>Case III</u>: $W_M < T < \infty$

$$I_4(a,b,c,T)=\frac{1}{2c^2}\left[Te^{-c^2T^2}+\frac{\sqrt{\pi}}{2c}\operatorname{erfc}(cT)\right]$$

where $X=\min(a,b)$ and,

$$P(c,X,L,U)=\frac{1}{2c^2}\Bigg\{Le^{-c^2L^2}\operatorname{erf}(XL)+Ue^{-c^2U^2}\operatorname{erfc}(XU)$$

$$+\frac{(X/\sqrt{\pi})}{X^2+c^2}\left[e^{-(X^2+c^2)L^2}-e^{-(X^2+c^2)U^2}\right]+\left[J_5(c,X,L)+I_5(c,X,U)\right]\Bigg\}$$

and I_5 is the integral I in Folder 5. Notice that two of the differences were replaced by positive sums.

If c varies widely, then the stepping procedure described in Section **(1.1)** for DQUAD8 should be applied on $\left[T,W_m\right]$ to reflect the scale of integration. Define $\sigma_c = 3/c$ to get an appropriate step when c is large. Also define $\sigma_m = W_m/2$ to get a nominal step size when c is small. To be sure that we cover both cases, we take steps

$$\sigma = \min(\sigma_c, \sigma_m),$$

and quadrature intervals (w_k, w_{k+1}) , $w_{k+1} = w_k + \sigma$, $w_0 = T$, k=1,2,…. When w_{k+1} exceeds W_m, for some k, we take the upper limit to be W_m and terminate the stepping procedure after the quadrature computation.

Any time a quadrature , relative to the accumulated sum, decreases below 100 times unit round-off, the stepping procedure is terminated.

In the main program, where we use quadrature exlusively for a comparison over $\left[T,\infty\right)$, we also apply the first part of this recipe at the end of every step. This means that T is replaced by the left end point of each quadrature interval. Thus the quadrature intervals (X1,X2) are incremented up by X1=X2 and $X2 = X1 + \sigma$ to obtain the next (new) interval. In the case X1 is less than or equal to W_m then $\sigma = \min(\sigma_c, W_m/2)$: in the case the X1 is in the interval $W_m < X1 \le W_M$, then $\sigma = \min(\sigma_c, W_M/2)$; and in the case X1 is greater than W_M we take $\sigma = \sigma_c$. The stepping procedure is terminated when any quadrature relative to the accumulated sum decreases below 100 times unit round-off (see also the procedure for PROGRAM I3COMP).

This procedure is used to compare formulas in PROGRAM I4COMP from file RESEARCH.FOR.

Folder 8c

Evaluation of

$$J_4(a,b,c,T) = \int_T^\infty w\ e^{-c^2w^2} erf(aw)erf(bw)dw = \frac{1}{2}\int_0^t \frac{e^{-c^2/u}}{u^2} erf\left(\frac{a}{\sqrt{u}}\right) erf\left(\frac{b}{\sqrt{u}}\right) du$$

$$J_4^c(a,b,c,T) = \int_T^\infty w\, e^{-c^2w^2} erfc(aw)erfc(bw)dw = \frac{1}{2}\int_0^t \frac{e^{-c^2/u}}{u^2} erfc\left(\frac{a}{\sqrt{u}}\right) erfc\left(\frac{b}{\sqrt{u}}\right) du$$

$$T = \frac{1}{\sqrt{t}}$$

Integrate J_4 by parts:

$$u = \mathrm{erf}(aw)\mathrm{erf}(bw) \qquad dv = we^{-c^2w^2}dw$$

$$du = \frac{2}{\sqrt{\pi}}\left[be^{-b^2w^2}\ \mathrm{erf}(aw) + ae^{-a^2w^2}\ \mathrm{erf}(bw)\right]dw \qquad v = \frac{-1}{2c^2}e^{-c^2w^2}$$

$$\begin{aligned} J_4(a,b,c,T) &= \frac{e^{-c^2T^2}}{2c^2}\mathrm{erf}(aT)\,\mathrm{erf}(bT) \\ &\quad + \frac{1}{c^2\sqrt{\pi}}\left[b\int_T^\infty e^{-(b^2+c^2)w^2}\ \mathrm{erf}(aw)dw + a\int_T^\infty e^{-(a^2+c^2)w^2}\ \mathrm{erf}(bw)dw\right] \\ &= \frac{e^{-c^2T^2}}{2c^2}\mathrm{erf}(aT)\,\mathrm{erf}(bT) + \frac{1}{c^2\sqrt{\pi}}\left[bJ_5(\sqrt{b^2+c^2},a,T) + aJ_5(\sqrt{a^2+c^2},b,T)\right] \end{aligned}$$

where $J_5(a,b,T)$ is the J function of Folder 5. The functions of Folder 5 were analyzed from a numerical point of view to minimize losses of significance and are the preferred forms for computation.

We follow the same pattern for $J_4^c(a,b,c,T)$:

$$u = \mathrm{erfc}(aw)\mathrm{erfc}(bw) \qquad dv = we^{-c^2w^2}dw$$

$$du = \frac{2}{\sqrt{\pi}}\left[-be^{-b^2w^2} erfc(aw) - ae^{-a^2w^2} erfc(bw)\right]dw \qquad v = \frac{-1}{2c^2}e^{-c^2w^2}$$

$$J_4^c(a,b,c,T)=\frac{e^{-c^2T^2}}{2c^2}erfc(aT)erfc(bT)$$
$$+\frac{1}{c^2\sqrt{\pi}}\left[-b\int_T^\infty e^{-(b^2+c^2)w^2}erfc(aw)dw-a\int_T^\infty e^{-(a^2+c^2)w^2}erfc(bw)dw\right]$$

$$=\frac{e^{-c^2T^2}}{2c^2}erfc(aT)erfc(bT)-\frac{1}{c^2\sqrt{\pi}}\left[bI_5(\sqrt{b^2+c^2},a,T)+aI_5(\sqrt{a^2+c^2},b,T)\right]$$

Special Cases

Taking $b\to\infty$ gives

$$\int_T^\infty we^{-c^2w^2}erf(aw)dw=\lim_{b\to\infty}J_4(a,b,c,T)=\frac{1}{2c^2}\left[e^{-c^2T^2}erf(aT)+\frac{2a}{\sqrt{\pi}}\lim_{b\to\infty}J_5(\sqrt{a^2+c^2},b,T)\right]$$
$$=\frac{1}{2c^2}\left[e^{-c^2T^2}erf(aT)+\frac{2a}{\sqrt{\pi}}\int_T^\infty e^{-(a^2+c^2)w^2}dw\right],$$
$$=\frac{1}{2c^2}\left[e^{-c^2T^2}erf(aT)+\frac{a}{\sqrt{a^2+c^2}}erfc(T\sqrt{a^2+c^2})\right]$$

and taking $b\to 0$ gives

$$\int_T^\infty we^{-c^2w^2}erfc(aw)dw=\lim_{b\to 0}J_4^c(a,b,c,T)=\frac{1}{2c^2}\left[e^{-c^2T^2}erfc(aT)-\frac{2a}{\sqrt{\pi}}\lim_{b\to 0}I_5(\sqrt{a^2+c^2},b,T)\right]$$
$$=\frac{1}{2c^2}\left[e^{-c^2T^2}erfc(aT)-\frac{2a}{\sqrt{\pi}}\int_T^\infty e^{-(a^2+c^2)w^2}dw\right]$$
$$=\frac{1}{2c^2}\left[e^{-c^2T^2}erfc(aT)-\frac{a}{\sqrt{a^2+c^2}}erfc(T\sqrt{a^2+c^2})\right]$$

Adding these two results gives the check computation

$$\int_T^\infty we^{-c^2w^2}dw=\frac{1}{2c^2}e^{-c^2T^2}.$$

Using erf(x)=1-erfc(x) and erfc(x)=1-erf(x) we also have the relations

$$J_4(a,b,c,T)=\frac{1}{2c^2}e^{-c^2T^2}-\int_T^\infty we^{-c^2w^2}erfc(aw)dw-\int_T^\infty we^{-c^2w^2}erfc(bw)dw+J_4^c(a,b,c,T)$$
$$J_4^c(a,b,c,T)=\frac{1}{2c^2}e^{-c^2T^2}-\int_T^\infty we^{-c^2w^2}erf(aw)dw-\int_T^\infty we^{-c^2w^2}erf(bw)dw+J_4(a,b,c,T)$$

where the integrals are computed above. Some losses of significance can be avoided in these expressions if the leading term is combined analytically with the larger of the leading terms of the integrals.

Folder 8d

Quadrature Procedure for $J_4(a,b,c,T)$

As in Folder 8b, we let

$$W_m = \min\left(\frac{6}{a},\frac{6}{b}\right), \quad W_M = \max\left(\frac{6}{a},\frac{6}{b}\right), \quad \text{and} \quad X = \min(a,b)$$

Case I, $\dfrac{1}{\sqrt{t}} = T \le W_m$

$$J_4(a,b,c,T) = \int_T^{W_m} we^{-c^2w^2}\,\text{erf}(aw)\,\text{erf}(bw)dw + \int_{W_m}^{W_M} we^{-c^2w^2}\,\text{erf}(Xw)dw + \int_{W_M}^{\infty} we^{-c^2w^2}dw$$

Case II, $W_m < T \le W_M$

$$J_4(a,b,c,T) = \int_T^{W_M} we^{-c^2w^2}\,\text{erf}(Xw)dw + \int_{W_M}^{\infty} we^{-c^2w^2}dw$$

Case III, $W_M < T < \infty$

$$J_4(a,b,c,T) = \int_T^{\infty} we^{-c^2w^2}dw$$

Now,

$$\int_Y^{\infty} we^{-c^2w^2}dw = \frac{1}{2c^2}e^{-c^2Y^2}$$

and we can integrate

$$S(c,X,L,U) = \int_L^{U} we^{-c^2w^2}\,\text{erf}(Xw)dw$$

by parts to obtain a closed form:

$$u = \text{erf}(Xw) \qquad\qquad dv = we^{-c^2w^2}dw$$

$$du = \frac{2X}{\sqrt{\pi}}e^{-X^2w^2}dw \qquad\qquad v = \frac{-1}{2c^2}e^{-c^2w^2}$$

$$S(c,X,L,U) = \frac{1}{2c^2}\left[e^{-c^2L^2}\,\mathrm{erf}(XL) - e^{-c^2U^2}\,\mathrm{erf}(XU)\right] + \frac{X}{c^2\sqrt{\pi}}\int_L^U e^{-(c^2+X^2)w^2}\,dw$$
$$= \frac{1}{2c^2}\left[e^{-c^2L^2}\,\mathrm{erf}(XL) - e^{-c^2U^2}\,\mathrm{erf}(XU)\right] + \frac{X}{2c^2\sqrt{c^2+X^2}}\left[\mathrm{erfc}(L\sqrt{c^2+X^2}) - \mathrm{erfc}(U\sqrt{c^2+X^2})\right]$$

Then,

<u>Case I</u>, $\frac{1}{\sqrt{t}} = T \le W_m$

$$J_4(a,b,c,T) = \int_T^{W_m} we^{-c^2w^2}\,\mathrm{erf}(aw)\,\mathrm{erf}(bw)dw + S(c,X,W_m,W_M) + \frac{1}{2c^2}e^{-c^2W_M^2}$$

<u>Case II</u>, $W_m < T \le W_M$

$$J_4(a,b,c,T) = S(c,X,T,W_M) + \frac{1}{2c^2}e^{-c^2W_M^2}$$

<u>Case III</u>, $W_M < T < \infty$

$$J_4(a,b,c,T) = \frac{1}{2c^2}e^{-c^2T^2}$$

We can save some losses of significance by combining the terms

$$\frac{1}{2c^2}e^{-c^2W_M^2} + S(c,X,L,W_M) = \frac{1}{2c^2}\left[e^{-c^2L^2}\,erf(XL) + e^{-c^2W_M^2}\,erfc(XW_M)\right]$$
$$+ \frac{X}{2c^2\sqrt{c^2+X^2}}\left[erfc(L\sqrt{c^2+X^2}) - erfc(W_M\sqrt{c^2+X^2})\right]$$

The stepping procedures for quadratures described in Folder 8b are used in PROGRAM J4COMP to compare formulas. This program is found in file RESEARCH.FOR.

Folder 9

Integrals Related to Folder 7

$$\mathbf{I_2(a,b,T) = \int_0^T erf(aw)erf(bw)dw = \frac{1}{2}\int_t^\infty \frac{erf(a/\sqrt{u})erf(b/\sqrt{u})du}{u^{3/2}}},$$

$$\mathbf{I_2^c(a,b,T) = \int_T^\infty erfc(aw)erfc(bw)dw = \frac{1}{2}\int_0^t \frac{erfc(a/\sqrt{u})erfc(b/\sqrt{u})du}{u^{3/2}}}$$

$$\mathbf{I_9(a,b,T) = \int_0^T w\,erf(aw)erf(bw)dw = \frac{1}{2}\int_t^\infty \frac{erf(a/\sqrt{u})erf(b/\sqrt{u})du}{u^2}}$$

$$\mathbf{I_9^c(a,b,T) = \int_T^\infty w\,erfc(aw)erfc(bw)dw = \frac{1}{2}\int_0^t \frac{erfc(a/\sqrt{u})erfc(b/\sqrt{u})du}{u^2}}$$

$$\boldsymbol{T = \frac{1}{\sqrt{t}}}, \quad \mathbf{t>0}$$

Donald E. Amos, December 2000, December 2005

Summary

Folder 9 is divided into Folder 9a and Folder 9b. Folder 9a computes representations for $I_2(a,b,T)$ and $I_2^c(a,b,T)$. Folder 9b computes representations for $I_9(a,b,T)$ and $I_9^c(a,b,T)$. $I_2^c(a,b,T)$ can be evaluated in terms of the $J_3(a,b,0,T)$ function of Folder 7e. Further manipulation produces a closed form for $I_2(a,b,T)$:

$$I_2(a,b,T) = T\,\mathrm{erf}(aT)\mathrm{erf}(bT) + \frac{e^{-a^2T^2}}{a\sqrt{\pi}}\mathrm{erf}(bT) + \frac{e^{-b^2T^2}}{b\sqrt{\pi}}\mathrm{erf}(aT) - \frac{\sqrt{a^2+b^2}}{ab\sqrt{\pi}}\mathrm{erf}(T\sqrt{a^2+b^2})$$

and

$$I_2^c(a,b,T) = -T erfc(aT)erfc(bT) + \frac{e^{-a^2T^2}}{a\sqrt{\pi}}erfc(bT) + \frac{e^{-b^2T^2}}{b\sqrt{\pi}}erfc(aT) - \frac{\sqrt{a^2+b^2}}{ab\sqrt{\pi}}erfc(T\sqrt{a^2+b^2})$$

$$I_2^c(a,b,T) = \frac{T}{(aT)(bT)}\left[\frac{ierfc(T\sqrt{a^2+b^2})}{\sqrt{\pi}} - ierfc(aT)ierfc(bT)\right].$$

In addition to the power series for $I_2(a,b,T)$ and $I_2^c(a,b,T)$, two other expressions are derived for the cases $a \le b$ (no loss of generality here) to resolve the indeterminant forms for a to zero.

$$I_2(a,b,T) = Terf(aT)erf(bT) + \frac{e^{-b^2T^2}}{b\sqrt{\pi}} erf(aT) - 2(\frac{1}{a\sqrt{\pi}} + R)e^{-a^2T^2/2}\sinh(a^2T^2/2) - Re^{-a^2T^2} erf(bT)$$

$$+RT\sqrt{a^2+b^2}\, e^{-a^2T^2}\sum_{n=1}^{\infty}(-2)^n i^n erfc(bT)\left[\frac{a^2T}{b+\sqrt{a^2+b^2}}\right]^{n-1}$$

$$I_2^c(a,b,T) = erfc(aT)\frac{ierfc(bT)}{b} - Re^{-a^2T^2} erfc(bT)$$

$$-RTe^{-a^2T^2}\sqrt{a^2+b^2}\sum_{n=1}^{\infty}(-2)^n i^n erfc(bT)\left[\frac{a^2T}{b+\sqrt{a^2+b^2}}\right]^{n-1} \qquad , R = \frac{1}{b\sqrt{\pi}}\cdot\frac{a}{b+\sqrt{a^2+b^2}}.$$

For $I_9(a,b,T)$ and $I_9^c(a,b,T)$, we get

$$2I_9(a,b,T) = T\cdot I_2(a,b,T) - \frac{1}{a\sqrt{\pi}}V_5(a,b,T) - \frac{1}{b\sqrt{\pi}}V_5(b,a,T)$$
$$-\frac{\sqrt{a^2+b^2}}{ab\sqrt{\pi}}\left[T\, erf(T\sqrt{a^2+b^2}) - \frac{1}{\sqrt{\pi}\sqrt{a^2+b^2}}\left(1-e^{-T^2(a^2+b^2)}\right)\right]$$

$$2I_9(a,b,T) = T^2 erf(aT)erf(bT) + \frac{Te^{-a^2T^2} erf(bT)}{a\sqrt{\pi}} + \frac{Te^{-b^2T^2} erf(aT)}{b\sqrt{\pi}}$$
$$-\frac{1}{a\sqrt{\pi}}V_5(a,b,T) - \frac{1}{b\sqrt{\pi}}V_5(b,a,T) - \frac{1}{\mathrm{ab}\pi}\left(1-e^{-T^2(a^2+b^2)}\right)$$

$$2I_9^c(a,b,T) = T\cdot I_2^c(a,b,T) + \frac{1}{a\sqrt{\pi}}I_5(a,b,T) + \frac{1}{b\sqrt{\pi}}I_5(b,a,T) - \frac{1}{ab\sqrt{\pi}}ierfc(T\sqrt{a^2+b^2})$$

and

$$2I_9^c(a,b,T) = -\mathrm{T}^2\mathrm{erfc(aT)erfc(bT)} + \frac{\mathrm{Te}^{-\mathrm{a}^2\mathrm{T}^2}\mathrm{erfc(bT)}}{\mathrm{a}\sqrt{\pi}} + \frac{\mathrm{Te}^{-\mathrm{b}^2\mathrm{T}^2}\mathrm{erfc(aT)}}{\mathrm{b}\sqrt{\pi}} + \frac{1}{\mathrm{a}\sqrt{\pi}}\mathrm{I}_5(\mathrm{a,b,T}) + \frac{1}{\mathrm{b}\sqrt{\pi}}\mathrm{I}_5(\mathrm{b,a,T})$$
$$-\frac{1}{\mathrm{ab}\sqrt{\pi}}\frac{\mathrm{e}^{-\mathrm{T}^2(\mathrm{a}^2+\mathrm{b}^2)}}{\sqrt{\pi}} \quad = \quad \frac{-ierfc(aT)ierfc(bT)}{ab} + \frac{1}{a\sqrt{\pi}}I_5(a,b,T) + \frac{1}{b\sqrt{\pi}}I_5(b,a,T)$$

The V_5 and I_5 functions are the V and I functions of Folder 5.

The corresponding expressions for $I_9(a,b,T)$ and $I_9^c(a,b,T)$ for $a \le b$ which remove the indeterminancy for a to zero are

$$2I_9(a,b,T) = T^2 erf(aT)erf(bT) + \frac{Te^{-b^2T^2} erf(aT)}{b\sqrt{\pi}} - \frac{1}{b\sqrt{\pi}}V_5(b,a,T) + U(a,b,T)$$

$$U(a,b,T) = H_1(a,b,T) + H_2(a,b,T) - W(a,b,T)$$

and

$$2I_9^c(a,b,T) \;=\; \frac{\mathrm{Terfc(aT)}}{\mathrm{b}}\, ierfc(bT) + \frac{1}{\mathrm{b}\sqrt{\pi}}\,\mathrm{I}_5(\mathrm{b,a,T}) + \mathrm{W(a,b,T)}, \qquad \mathrm{a} \le \mathrm{b}$$

where

$$H_1(a,b,T) = \frac{1}{a\sqrt{\pi}}\left[Te^{-a^2T^2} - \frac{\sqrt{\pi}}{2a} erf(aT) \right], \qquad H_2(a,b,T) = \left[\frac{1}{a^2\pi}\tan^{-1}\left(\frac{a}{b}\right) - \frac{1}{ab\pi} \right]$$

$$W(a,b,T) \;=\; -\frac{e^{-a^2T^2}\, ierfc(bT)}{ab\sqrt{\pi}} + \frac{1}{a\sqrt{\pi}} I_5(a,b,T)$$

$$= \frac{1}{a\sqrt{\pi}}\left[-\frac{e^{-a^2T^2}\, ierfc(bT)}{b} + \frac{ierfc(T\sqrt{a^2+b^2})}{d} \right] + G_9(a,b,T)$$

$$G_9(a,b,T) = \frac{1}{2\pi d^2}\sum_{k=1}^{\infty}\frac{(1/2)_k}{k!}\left(\frac{a}{d}\right)^{2k-1} E_{k+3/2}\left(d^2T^2\right), \qquad d^2 = a^2 + b^2 .$$

and the representations which remove the indeterminancies for a to zero are

$$H_1(a,b,T) = \frac{2\mathrm{T}^2}{\sqrt{\pi}}\sum_{k=1}^{\infty}\frac{(-1)^k (aT)^{2k-1}}{(k-1)!(2k+1)}\;, \qquad H_2(a,b,T) = \frac{1}{b^2\pi}\sum_{k=1}^{\infty}\frac{(-1)^k (a/b)^{2k-1}}{(2k+1)}, \quad \frac{\mathrm{a}}{\mathrm{b}} < 1$$

$$\mathrm{W(a,b,T)} \;=\frac{\mathrm{(a/b)e^{-a^2T^2}}}{\sqrt{\pi}\sqrt{\mathrm{a}^2+\mathrm{b}^2}\,(b+\sqrt{\mathrm{a}^2+\mathrm{b}^2})}\left[-\mathrm{ierfc(bT)} + \mathrm{bT}\sum_{\mathrm{n=2}}^{\infty}(-2)^{\mathrm{n-1}}\mathrm{n}\;\mathrm{i^n erfc(bT)(bT}\phi)^{\mathrm{n}-2} \right] + \mathrm{G}_9(a,b,T)$$

$$G_9(a,b,T) = \frac{1}{2\pi d^2}\sum_{k=1}^{\infty}\frac{(1/2)_k}{k!}\left(\frac{a}{d}\right)^{2k-1} E_{k+3/2}\left(d^2T^2\right), \qquad d^2 = a^2 + b^2 .$$

Power series expressions for all four functions are also developed, which for numerical purposes, would be applied first to filter out many small a and b cases.

Folder 9a

Computation of $\mathbf{I_2(a,b,T)}$ and $\mathbf{I_2^c(a,b,T)}$

Derivations

We start the computation of $I_2(a,b,T)$ by using the relation

$$erf(aw)erf(bw) = 1 - erfc(aw) - erfc(bw) + erfc(aw)erfc(bw)$$

Then,

$$\int_0^T \mathrm{erfc}(aw)dw = \int_0^\infty \mathrm{erfc}(aw)dw - \int_T^\infty \mathrm{erfc}(aw)dw = \frac{1}{\mathrm{a}} ierfc(0) - \frac{1}{a} ierfc(aT) = \frac{1}{a\sqrt{\pi}} - \frac{1}{a} ierfc(aT)$$

and

$$I_2(a,b,T) = T - \frac{1}{a\sqrt{\pi}} + \frac{ierfc(aT)}{a} - \frac{1}{b\sqrt{\pi}} + \frac{ierfc(bT)}{b} + [J_3(a,b,0,0) - J_3(a,b,0,T)]$$

where $J_3(a,b,c,T) = \int_T^\infty e^{-c^2w^2} \mathrm{erfc}(aw)\mathrm{erfc}(bw)dw$ is computed in Folder 7 with an explicit expression for both $J_3(a,b,0,0)$ and $J_3(a,b,0,T)$ in Folder 7e:

$$I_2^c(a,b,T) = J_3(a,b,0,T) = \frac{e^{-a^2T^2}\,\mathrm{erfc}(bT)}{a\sqrt{\pi}} + \frac{e^{-b^2T^2}\,\mathrm{erfc}(aT)}{b\sqrt{\pi}} - T\,\mathrm{erfc}(aT)\,\mathrm{erfc}(bT)$$
$$- \frac{\sqrt{a^2+b^2}}{ab\sqrt{\pi}}\mathrm{erfc}(T\sqrt{a^2+b^2})$$

Now with $T = 0$ we get

$$I_2^c(a,b,0) = J_3(a,b,0,0) = \frac{1}{a\sqrt{\pi}} + \frac{1}{b\sqrt{\pi}} - \frac{\sqrt{a^2+b^2}}{ab\sqrt{\pi}} = \frac{(2/\sqrt{\pi})}{a+b+\sqrt{a^2+b^2}}$$

Making the replacements, $I_2(a,b,T)$ is expressed by

$$I_2(a,b,T) = T + \frac{i\,\mathrm{erfc}(aT)}{a} + \frac{i\,\mathrm{erfc}(bT)}{b} - \frac{e^{-a^2T^2}\,\mathrm{erfc}(bT)}{a\sqrt{\pi}} - \frac{e^{-b^2T^2}\,\mathrm{erfc}(aT)}{b\sqrt{\pi}}$$
$$+ T\,\mathrm{erfc}(aT)\,\mathrm{erfc}(bT) - \frac{\sqrt{a^2+b^2}}{ab\sqrt{\pi}}\mathrm{erf}(T\sqrt{a^2+b^2})$$

Notice that the symmetry in a and b is retained. Now

$$i\,\mathrm{erfc}(x) = -x\,\mathrm{erfc}(x) + \frac{e^{-x^2}}{\sqrt{\pi}}$$

and, using the expression for $\mathrm{erf}(aw)\mathrm{erf}(bw)$ above, we get

$$I_2(a,b,T) = T\,\mathrm{erf}(aT)\,\mathrm{erf}(bT) + \frac{e^{-a^2T^2}}{a\sqrt{\pi}}\mathrm{erf}(bT) + \frac{e^{-b^2T^2}}{b\sqrt{\pi}}\mathrm{erf}(aT) - \frac{\sqrt{a^2+b^2}}{ab\sqrt{\pi}}\mathrm{erf}(T\sqrt{a^2+b^2}).$$

The asymptotics are:

$$I_2(a,b,T) \sim T - \frac{\sqrt{a^2+b^2}}{ab\sqrt{\pi}} \ \text{ for } T \to \infty \ or\ t \to 0.$$

with $I_2^c(a,b,\infty)=0$. For $T\to 0$ or $t\to\infty$, $I_2(a,b,0) = 0$ and

$$I_2^c(a,b,0)=\frac{1}{a\sqrt{\pi}}+\frac{1}{b\sqrt{\pi}}-\frac{\sqrt{a^2+b^2}}{ab\sqrt{\pi}}=\frac{(2/\sqrt{\pi})}{a+b+\sqrt{a^2+b^2}}.$$

Other Representations of $\mathbf{I_2(a,b,T)}$ and $\mathbf{I_2^c(a,b,T)}$

Numerically, there is a problem with these formulae for a to zero or b to zero because of the indeterminant form where a or b appear in the denominator. We therefore develop the power series for both quantities as well as series for a or b to zero.

Power Series for $\mathbf{I_2(a,b,T)}$ and $\mathbf{I_2^c(a,b,T)}$ for Small Parameters

In Folder 7b we developed the power series for the product erf(aw)erf(bw)

$$\operatorname{erf}(aw)=\frac{2aw}{\sqrt{\pi}}\sum_{k=0}^{\infty} C_k(a^2)w^{2k};\qquad \operatorname{erf}(bw)=\frac{2bw}{\sqrt{\pi}}\sum_{k=0}^{\infty} C_k(b^2)w^{2k}$$

and
$$\operatorname{erf}(aw)\operatorname{erf}(bw)=\frac{4abw^2}{\pi}\sum_{k=0}^{\infty} U_k(a^2,b^2)w^{2k}$$

where
$$U_k(a^2,b^2)=\sum_{m=0}^{k} C_m(a^2)C_{k-m}(b^2),\qquad C_k(a^2)=\frac{(-1)^k a^{2k}}{k!(2k+1)}$$

Integration yields

$$I_2(a,b,T)=\int_0^T erf(aw)erf(bw)dw=\frac{4abT^3}{\pi}\sum_{k=0}^{\infty}\frac{U_k(a^2,b^2)T^{2k}}{2k+3}=\frac{4T(aT)(bT)}{\pi}\sum_{k=0}^{\infty}\frac{U_k(a^2T^2,b^2T^2)}{2k+3}$$

where the powers of T^2 have been distributed for better scaling. For $I_2^c(a,b,T)$ we use the power series for

$$I_2^c(a,b,T)=I_2^c(a,b,0)-\int_0^T erfc(aw)erfc(bw)dw$$

with the power series for each of the terms in

$$erfc(aw)erfc(bw)=1-erf(aw)-erf(bw)+erf(aw)erf(bw)$$

The result is

$$I_2^c(a,b,T)=$$
$$\frac{(2/\sqrt{\pi})}{a+b+\sqrt{a^2+b^2}}-T+\frac{2T(aT)}{\sqrt{\pi}}\sum_{k=0}^{\infty}\frac{C_k(a^2T^2)}{2k+2}+\frac{2T(bT)}{\sqrt{\pi}}\sum_{k=0}^{\infty}\frac{C_k(b^2T^2)}{2k+2}-\frac{4T(aT)(bT)}{\pi}\sum_{k=0}^{\infty}\frac{U_k(a^2T^2,b^2T^2)}{2k+3}$$

An Alternate form for $\mathbf{I_2(a,b,T)}$ and $\mathbf{I_2^c(a,b,T)}$ small a or b

In the expressions for $I_2(a,b,T)$ and $I_2^c(a,b,T)$ we can resolve the indeterminant form by expanding the terms involving

$$\frac{\sqrt{a^2+b^2}}{ab\sqrt{\pi}} erfc(T\sqrt{a^2+b^2}) \text{ or } \frac{\sqrt{a^2+b^2}}{ab\sqrt{\pi}} erf(T\sqrt{a^2+b^2})$$

into Taylor series about the point bT and combining it with the appropriate term in each expression. We can, without loss of generality, take $a \le b$ since one parameter is always smaller or equal to the other. Thus with, $a \le b$ we start with the Taylor series about x

$$e^{z^2} erfc(z) = \sum_{n=0}^{\infty} (-2)^n e^{x^2} ierfc(x)(z-x)^n$$

since(A&S Eqn. (7.2.9)) $\frac{d^n}{dz^n}\left[e^{z^2} erfc(z)\right] = (-2)^n n! e^{z^2} i^n erfc(z)$.

With $z = T\sqrt{a^2+b^2}$ *and* $x = bT$ we get

$$e^{T^2(a^2+b^2)} erfc(T\sqrt{a^2+b^2}) = e^{b^2T^2} \sum_{n=0}^{\infty} (-2)^n i^n erfc(bT)(bT\phi)^n, \quad \phi = \sqrt{1+a^2/b^2} - 1 = \frac{a^2/b^2}{1+\sqrt{1+a^2/b^2}}$$

or $$erfc(T\sqrt{a^2+b^2}) = e^{-a^2T^2} \sum_{n=0}^{\infty} (-2)^n i^n erfc(bT)(bT\phi)^n, \quad \mathrm{bT}\phi = \frac{\mathrm{a}^2 T}{b+\sqrt{a^2+b^2}}.$$

Then,

$$I_2^c(a,b,T) = -Terfc(aT)erfc(bT) + \frac{e^{-b^2T^2} erfc(aT)}{b\sqrt{\pi}} + \frac{e^{-a^2T^2} erfc(bT)}{a\sqrt{\pi}} - \frac{\sqrt{a^2+b^2}}{ab\sqrt{\pi}} erfc(T\sqrt{a^2+b^2})$$

$$= -Terfc(aT)erfc(bT) + \frac{e^{-b^2T^2} erfc(aT)}{b\sqrt{\pi}} + \frac{e^{-a^2T^2} erfc(bT)}{a\sqrt{\pi}} - \frac{\sqrt{a^2+b^2}}{ab\sqrt{\pi}} e^{-a^2T^2} erfc(bT)$$

$$- e^{-a^2T^2} \frac{\sqrt{a^2+b^2}}{ab\sqrt{\pi}} \sum_{n=1}^{\infty} (-2)^n i^n erfc(bT)(bT\phi)^n, \quad bT\phi = \frac{a^2T}{b+\sqrt{a^2+b^2}}$$

$$I_2^c(a,b,T) = -Terfc(aT)erfc(bT) + \frac{e^{-b^2T^2} erfc(aT)}{b\sqrt{\pi}} + \frac{e^{-a^2T^2} erfc(bT)}{a\sqrt{\pi}}\left[1 - \frac{\sqrt{a^2+b^2}}{b}\right]$$

$$- \frac{\sqrt{a^2+b^2}}{b\sqrt{\pi}} \frac{aTe^{-a^2T^2}}{b+\sqrt{a^2+b^2}} \sum_{n=1}^{\infty} (-2)^n i^n erfc(bT)\left(\frac{a^2T}{b+\sqrt{a^2+b^2}}\right)^{n-1}, \quad bT\phi = \frac{a^2T}{b+\sqrt{a^2+b^2}}$$

and finally,

$$I_2^c(a,b,T) = erfc(aT)\frac{ierfc(bT)}{b} - Re^{-a^2T^2} erfc(bT)$$

$$- RTe^{-a^2T^2} \sqrt{a^2+b^2} \sum_{n=1}^{\infty} (-2)^n i^n erfc(bT)\left[\frac{a^2T}{b+\sqrt{a^2+b^2}}\right]^{n-1}$$

where

$$R = \frac{1}{b\sqrt{\pi}} \cdot \frac{a}{b+\sqrt{a^2+b^2}}.$$

Notice that the expression for T=0 takes the form

$$I_2^c(a,b,0)=\frac{1}{b\sqrt{\pi}}-\frac{1}{b\sqrt{\pi}}\left[\frac{a}{b+\sqrt{a^2+b^2}}\right]=\frac{1}{b\sqrt{\pi}}\left[1-\frac{a(\sqrt{a^2+b^2}-b)}{a^2}\right]=\frac{1}{b\sqrt{\pi}}\left[\frac{a+b-\sqrt{a^2+b^2})}{a}\right]$$

$$=\frac{(2/\sqrt{\pi})}{a+b+\sqrt{a^2+b^2}}$$

For $I_2(a,b,T)$ we write the expression in the form

$$I_2(a,b,T)=Terf(aT)erf(bT)+\frac{e^{-b^2T^2}}{b\sqrt{\pi}}erf(aT)+\frac{e^{-a^2T^2}}{a\sqrt{\pi}}erf(bT)-\frac{\sqrt{a^2+b^2}}{ab\sqrt{\pi}}erf(T\sqrt{a^2+b^2})=$$

$$=Terf(aT)erf(bT)+\frac{e^{-b^2T^2}}{b\sqrt{\pi}}erf(aT)+\frac{e^{-a^2T^2}}{a\sqrt{\pi}}-\frac{\sqrt{a^2+b^2}}{ab\sqrt{\pi}}-\left[\frac{e^{-a^2T^2}}{a\sqrt{\pi}}erfc(bT)-\frac{\sqrt{a^2+b^2}}{ab\sqrt{\pi}}erfc(T\sqrt{a^2+b^2})\right]$$

and manipulate the expressions with a in the denominator. Notice that the quantity in brackets is the form we manipulated to compute $I_2^c(a,b,T)$. It only remains to reduce the expression

$$D=\frac{e^{-a^2T^2}}{a\sqrt{\pi}}-\frac{\sqrt{a^2+b^2}}{ab\sqrt{\pi}}$$

to a manageable form. Then,

$$D=\frac{1-1+e^{-a^2T^2}}{a\sqrt{\pi}}-\frac{\sqrt{a^2+b^2}-b+b}{ab\sqrt{\pi}}=\frac{1}{a\sqrt{\pi}}-\frac{2e^{-a^2T^2/2}\sinh(a^2T^2/2)}{a\sqrt{\pi}}-\frac{\sqrt{a^2+b^2}-b}{ab\sqrt{\pi}}-\frac{1}{a\sqrt{\pi}}$$

$$=-\frac{2e^{-a^2T^2/2}\sinh(a^2T^2/2)}{a\sqrt{\pi}}-R$$

and the quotient sinh(*)/a does not lose significant digits for a to zero. Further more, the quantity

$$R=\frac{1}{b\sqrt{\pi}}\cdot\frac{a}{b+\sqrt{a^2+b^2}}$$

appears in the analysis for

$$\left[\frac{e^{-a^2T^2}}{a\sqrt{\pi}}erfc(bT)-\frac{\sqrt{a^2+b^2}}{ab\sqrt{\pi}}erfc(T\sqrt{a^2+b^2})\right]$$

and we get

$$I_2(a,b,T)=Terf(aT)erf(bT)+\frac{e^{-b^2T^2}}{b\sqrt{\pi}}erf(aT)+D+Re^{-a^2T^2}erfc(bT)$$

$$+RT\sqrt{a^2+b^2}\,e^{-a^2T^2}\sum_{n=1}^{\infty}(-2)^n\,i^n erfc(bT)\left[\frac{a^2T}{b+\sqrt{a^2+b^2}}\right]^{n-1}$$

But

$$D + Re^{-a^2T^2} erfc(bT) = -\frac{2e^{-a^2T^2/2}\sinh(a^2T^2/2)}{a\sqrt{\pi}} - R\left[1 - e^{-a^2T^2} erfc(bT)\right]$$

can be computed with significant digits for T to zero since

$$\left[1 - e^{-a^2T^2} erfc(bT)\right] = 1 - e^{-a^2T^2} + e^{-a^2T^2} erf(bT) = 2e^{-a^2T^2/2}\sinh(a^2T^2/2) + e^{-a^2T^2} erf(bT)$$

and both terms have T as a factor in the series expansion. Finally,

$$I_2(a,b,T) = Terf(aT)erf(bT) + \frac{e^{-b^2T^2}}{b\sqrt{\pi}} erf(aT) - 2\left(\frac{1}{a\sqrt{\pi}} + R\right)e^{-a^2T^2/2}\sinh(a^2T^2/2) - Re^{-a^2T^2} erf(bT)$$
$$+RT\sqrt{a^2+b^2}\,e^{-a^2T^2}\sum_{n=1}^{\infty}(-2)^n i^n erfc(bT)\left[\frac{a^2T}{b+\sqrt{a^2+b^2}}\right]^{n-1}$$

where

$$R = \frac{1}{b\sqrt{\pi}} \cdot \frac{a}{b+\sqrt{a^2+b^2}}.$$

The bound

$$\frac{a^2T}{b+\sqrt{a^2+b^2}} \le \frac{aT}{1+\sqrt{2}} < \frac{aT}{2.4}$$

(replace b by a in the denominator since a is smaller than b) ensures that we get satisfactory convergence for $aT<2.4$ since

$$i^n erfc(bT) \le i^n erfc(0) = \frac{1}{2^n\Gamma(n/2+1)}.$$

We get (relative) errors $O(10^{-16})$ in about 40 terms in the worst case. The iterated co-error function sequence can be computed in subroutine DINERFC.

$\mathbf{I_2(a,b,T)}$ and $\mathbf{I_2^c(a,b,T)}$ for Large Parameters

The form developed above for $I_2(a,b,T)$ in terms of error functions gives very good numerical values and does not suffer seriously from losses of significance. However, the closed form expression for $I_2^c(a,b,T)$ above in terms of the co-error function loses significant digits when the parameters are large. This comes about because the asymptotics for each term are dominated by the same quantity, but sum to zero. This makes the evaluation depend on lower order magnitude contributions to the asymptotics. To remedy this, we write the formula in terms of the iterated coerror function ierfc(*) using

$$erfc(x) = \frac{1}{x}\left[\frac{e^{-x^2}}{\sqrt{\pi}} - ierfc(x)\right], \qquad ierfc(x) \sim \frac{e^{-x^2}}{2x^2\sqrt{\pi}}, \quad \mathrm{x} \to \infty$$

which shows the dominant behavior for both functions. The result for $I_2^c(a,b,T)$ is

$$I_2^c(a,b,T) = \frac{T}{(aT)(bT)}\left[\frac{ierfc(T\sqrt{a^2+b^2})}{\sqrt{\pi}} - ierfc(aT)ierfc(bT)\right].$$

Numerical Procedures

A tree for programming a subroutine for $I_2(a,b,T)$ and $I_2^c(a,b,T)$ is outlined below:
First of all take $a \le b$ since, for any problem, one is a maximum and one is a minimum.

(1) Test the expression: $(aT \le 1.25)$ *and* $(bT \le 1.25)$ *and* $(T \le 1.00)$. If the expression is true,. evaluate the power series to filter out cases with small parameters.

(2) If (1) fails, test $aT \le 2.4$. If this is satisfied, evaluate the series in powers of $\dfrac{a^2T}{b+\sqrt{a^2+b^2}}$.

(3) If(1) and (2) fail, then evaluate the original closed form for $I_2(a,b,T)$ and the formula developed in the previous section for $I_2^c(a,b,T)$. Notice that because (2) failed and we take $a \le b$, then $bT \ge aT > 2.4$ and the product (aT)(bT) is bounded away from zero.

Special Cases

The definition of the iterated co-error function gives

b=0 in $\mathrm{I}_2^c(a,b,T)$: $\qquad I_9^c(a,0,T) = \int_T^\infty erfc(aw)dw = \frac{1}{a} ierfc(aT)$.

[Beck et al., Table F.2, p.422] gives the following:

$b \to \infty$ in $\mathrm{I}_2(a,b,T)$:

$$\mathrm{I}_2(a,\infty,T) = \int_0^T erf(aw)\,dw = T\,erf(aT) - \frac{1}{a\sqrt{\pi}}\left[1-e^{-a^2T^2}\right] = T\,erf(aT) - \frac{2e^{-a^2T^2/2}}{a\sqrt{\pi}}\sinh(a^2T^2/2).$$

Folder 9b

Computation of $\mathbf{I_9(a,b,T)}$ and $\mathbf{I_9^c(a,b,T)}$

Derivations

We start with

$$I_9(a,b,T) = \int_0^T w\,erf(aw)erf(bw)dw$$

and integrate by parts

$$u = w \qquad dv = erf(aw)erf(bw)dw$$

$$du = dw \qquad v = I_2(a,b,w)$$

to get

$$I_9(a,b,T) = T\,I_2(a,b,T) - \int_0^T I_2(a,b,w)dw$$

where $I_2(a,b,T)$ is computed in Folder 9a,

$$I_2(a,b,T) = T\,\mathrm{erf}(aT)\,\mathrm{erf}(bT) + \frac{e^{-a^2T^2}}{a\sqrt{\pi}}\mathrm{erf}(bT) + \frac{e^{-b^2T^2}}{b\sqrt{\pi}}\mathrm{erf}(aT) - \frac{\sqrt{a^2+b^2}}{ab\sqrt{\pi}}\mathrm{erf}(T\sqrt{a^2+b^2})\,.$$

We use this form to integrate $I_2(a,b,w)$,

$$I_9(a,b,T) = \mathrm{T}\cdot\mathrm{I_2(a,b,T)} - \mathrm{I_9(a,b,T)} - \frac{1}{\mathrm{a}\sqrt{\pi}}\mathrm{V_5(a,b,T)} - \frac{1}{\mathrm{b}\sqrt{\pi}}\mathrm{V_5(b,a,T)} + \frac{\sqrt{\mathrm{a}^2+\mathrm{b}^2}}{\mathrm{ab}\sqrt{\pi}}\int_0^{\mathrm{T}}\mathrm{erf}(\mathrm{w}\sqrt{\mathrm{a}^2+\mathrm{b}^2})\mathrm{dw}$$

where

$$V_5(a,b,T) = \int_0^T e^{-a^2w^2}erf(bw)$$

is the V function of Folder 5. Solving for $I_9(a,b,w)$ gives

$$2I_9(a,b,T) = T\cdot I_2(a,b,T) - \frac{1}{a\sqrt{\pi}}V_5(a,b,T) - \frac{1}{b\sqrt{\pi}}V_5(b,a,T)$$
$$- \frac{\sqrt{a^2+b^2}}{ab\sqrt{\pi}}\left[T\,erf(T\sqrt{a^2+b^2}) - \frac{1}{\sqrt{a^2+b^2}\sqrt{\pi}}(1-e^{-T^2(a^2+b^2)})\right]$$

using

$$\int_0^T erf(aw)dw = T\,erf(aT) - \frac{1}{a\sqrt{\pi}}\left[1-e^{-a^2T^2}\right] = T\,erf(aT) - \frac{2}{a\sqrt{\pi}}e^{-a^2T^2/2}\sinh(a^2T^2/2)\,.$$

Substituting $I_2(a,b,T)$ from Folder 9a and combining terms gives

$$2I_9(a,b,T) = T^2erf(aT)erf(bT) + \frac{Te^{-a^2T^2}erf(bT)}{a\sqrt{\pi}} + \frac{Te^{-b^2T^2}erf(aT)}{b\sqrt{\pi}}$$
$$- \frac{1}{a\sqrt{\pi}}V_5(a,b,T) - \frac{1}{b\sqrt{\pi}}V_5(b,a,T) - \frac{1}{\mathrm{ab}\pi}\left(1-e^{-T^2(a^2+b^2)}\right)$$

A similar integration by parts for

$$I_9^c(a,b,T) = \int_T^\infty w\, erfc(aw) erfc(bw) dw$$

$$u = w \qquad\qquad dv = erfc(aw)erfc(bw)dw$$
$$du = dw \qquad\qquad v = -I_2^c(a,b,w)$$

gives

$$2I_9^c(a,b,T) = T \cdot I_2^c(a,b,T) + \frac{1}{a\sqrt{\pi}} I_5(a,b,T) + \frac{1}{b\sqrt{\pi}} I_5(b,a,T) - \frac{1}{ab\sqrt{\pi}} ierfc(T\sqrt{a^2+b^2})$$

where I_5 is the I function of Folder 5 and $I_2^c(a,b,T)$ is computed in Folder 9a**.** If we combine appropriate terms from $\mathrm{T}\, I_2^c(a,b,T)$ with the iterated co-error function and use

$$ierfc(x) + x\, erfc(x) = \frac{e^{-x^2}}{\sqrt{\pi}}$$

we get

$$2I_9^c(a,b,T) = -\mathrm{T}^2\mathrm{erfc(aT)erfc(bT)} + \frac{\mathrm{Te}^{-a^2T^2}\,\mathrm{erfc(bT)}}{\mathrm{a}\sqrt{\pi}} + \frac{\mathrm{Te}^{-b^2T^2}\,\mathrm{erfc(aT)}}{\mathrm{b}\sqrt{\pi}} + \frac{1}{\mathrm{a}\sqrt{\pi}}\mathrm{I}_5\mathrm{(a,b,T)} + \frac{1}{\mathrm{b}\sqrt{\pi}}\mathrm{I}_5\mathrm{(b,a,T)}$$
$$-\frac{1}{\mathrm{ab}\sqrt{\pi}}\frac{\mathrm{e}^{-T^2(a^2+b^2)}}{\sqrt{\pi}} = \frac{-\mathrm{ierfc(aT)\,ierfc(bT)}}{\mathrm{ab}} + \frac{1}{a\sqrt{\pi}}\mathrm{I}_5\mathrm{(a,b,T)} + \frac{1}{\mathrm{b}\sqrt{\pi}}\mathrm{I}_5\mathrm{(b,a,T)}$$

These representations suffer numerical difficulties for small a and b, and the power series should be used to screen out many of the small parameter cases in a numerical subroutine.

<u>Power Series for $\mathbf{I_9(a,b,T)}$ and $\mathbf{I_9^c(a,b,T)}$</u>

From above,

$$\mathrm{erf}(aw) = \frac{2aw}{\sqrt{\pi}}\sum_{k=0}^{\infty} C_k(a^2)w^{2k}; \qquad \mathrm{erf}(bw) = \frac{2bw}{\sqrt{\pi}}\sum_{k=0}^{\infty} C_k(b^2)w^{2k}$$

and
$$\mathrm{erf}(aw)\,\mathrm{erf}(bw) = \frac{4abw^2}{\pi}\sum_{k=0}^{\infty} U_k(a^2,b^2)w^{2k}$$

where
$$U_k(a^2,b^2) = \sum_{m=0}^{k} C_m(a^2)C_{k-m}(b^2), \qquad C_k(a^2) = \frac{(-1)^k a^{2k}}{k!(2k+1)}$$

Integration yields

$$I_9(a,b,T) = \int_0^T w\, erf(aw) erf(bw) dw = \frac{4abT^4}{\pi}\sum_{k=0}^{\infty}\frac{U_k(a^2,b^2)T^{2k}}{2k+4} = \frac{4T^2(aT)(bT)}{\pi}\sum_{k=0}^{\infty}\frac{U_k(a^2T^2,b^2T^2)}{2k+4}$$

where the powers of T^2 have been distributed for better scaling. For $I_9^c(a,b,T)$ we use the power series for

$$I_9^c(a,b,T) = I_9^c(a,b,0) - \int_0^T w\, erfc(aw) erfc(bw) dw$$

with the power series for each of the terms in

$$erfc(aw)erfc(bw) = 1 - erf(aw) - erf(bw) + erf(aw)erf(bw)$$

The result is

$$I_9^c(a,b,T) =$$

$$\mathrm{I_9^c(a,b,0)} - \frac{\mathrm{T}^2}{2} + \frac{2\mathrm{T}^2(\mathrm{aT})}{\sqrt{\pi}}\sum_{\mathrm{k=0}}^{\infty}\frac{\mathrm{C_k(a^2T^2)}}{\mathrm{2k+3}} + \frac{2\mathrm{T}^2(\mathrm{bT})}{\sqrt{\pi}}\sum_{\mathrm{k=0}}^{\infty}\frac{\mathrm{C_k(b^2T^2)}}{\mathrm{2k+3}} - \frac{4\mathrm{T}^2(\mathrm{aT})(\mathrm{bT})}{\pi}\sum_{\mathrm{k=0}}^{\infty}\frac{\mathrm{U_k(a^2T^2,b^2T^2)}}{\mathrm{2k+4}}$$

where $I_5(a,b,0)$ is computed in Folder 5 and a formula for $I_9^c(a,b,T)$ from the previous section is used to compute

$$I_9^c(a,b,0) = \frac{1}{2ab\pi}\left[\ \frac{b}{a}\tan^{-1}\left(\frac{a}{b}\right) + \frac{a}{b}\tan^{-1}\left(\frac{b}{a}\right) - 1\right].$$

Since one parameter is always dominant, we can take $a \le b$, use $\tan^{-1}(x) + \tan^{-1}(1/x) = \pi/2$, and for small a we have

$$I_9^c(a,b,0) = \frac{1}{2ab\pi}\left[\frac{b}{a}\tan^{-1}\left(\frac{a}{b}\right) - 1\right] - \frac{1}{2b^2\pi}\tan^{-1}\left(\frac{a}{b}\right) + \frac{1}{4b^2}, \qquad \mathrm{a \le b}$$

The term in brackets is $O(a^2/b^2)$ and the product goes to zero as a goes to zero since

$$\frac{1}{x}\tan^{-1}(x) - 1 = \sum_{k=1}^{\infty}(-1)^k\frac{x^{2k}}{2k+1}, \qquad |\mathrm{x}| < 1.$$

<u>Series for</u> $\mathbf{I_9(a,b,T)}$ <u>and</u> $\mathbf{I_9^c(a,b,T)}$ <u>for small</u> $\mathbf{a \le b}$.

Rewriting the expression above and grouping the reciprocal powers of a, we have

$$2I_9^c(a,b,T) = -\mathrm{T^2 erfc(aT)erfc(bT)} + \frac{\mathrm{Te^{-b^2T^2}erfc(aT)}}{\mathrm{b}\sqrt{\pi}} + \frac{1}{\mathrm{b}\sqrt{\pi}}\mathrm{I_5(b,a,T)} + \mathrm{W(a,b,T)}$$

$$= \frac{\mathrm{Terfc(aT)}}{\mathrm{b}} ierfc(bT) + \frac{1}{\mathrm{b}\sqrt{\pi}}\mathrm{I_5(b,a,T)} + \mathrm{W(a,b,T)}$$

$$\mathrm{W(a,b,T)} = \frac{\mathrm{Te^{-a^2T^2}erfc(bT)}}{\mathrm{a}\sqrt{\pi}} - \frac{1}{\mathrm{ab}\sqrt{\pi}}\frac{\mathrm{e^{-T^2(a^2+b^2)}}}{\sqrt{\pi}} + \frac{1}{\mathrm{a}\sqrt{\pi}}\mathrm{I_5(a,b,T)}$$

Now we write the first two terms in terms of ierfc(*) and use the G series for I_5 with $E_{3/2}(x^2) = 2\sqrt{\pi}\ ierfc(x)$ to obtain

$$W(a,b,T) \;=\; -\frac{e^{-a^2T^2}\, ierfc(bT)}{ab\sqrt{\pi}} + \frac{1}{a\sqrt{\pi}} I_5(a,b,T)$$

$$= \frac{1}{a\sqrt{\pi}}\left[-\frac{e^{-a^2T^2}\, ierfc(bT)}{b} + \frac{ierfc(T\sqrt{a^2+b^2})}{d}\right] + G_9(a,b,T)$$

where

$$G_9(a,b,T) = \frac{1}{2\pi d^2}\sum_{k=1}^{\infty}\frac{(1/2)_k}{k!}\left(\frac{a}{d}\right)^{2k-1} E_{k+3/2}\left(d^2T^2\right), \qquad d^2 = a^2+b^2 .$$

Now, the expansion used in the computation of I_2^c

$$e^{z^2} erfc(z) = \sum_{n=0}^{\infty}(-2)^n e^{x^2} i^n erfc(x)(z-x)^n$$

can be differentiated wrt z to get

$$e^{z^2} ierfc(z) = \sum_{n=1}^{\infty}(-2)^{n-1} n e^{x^2} i^n erfc(x)(z-x)^{n-1} .$$

With $z = T\sqrt{a^2+b^2}$ *and* $x = bT$, we get

$$ierfc(T\sqrt{a^2+b^2}) = e^{-a^2T^2} ierfc(bT) + e^{-a^2T^2}\sum_{n=2}^{\infty}(-2)^{n-1} n\, i^n erfc(bT)(bT\phi)^{n-1}$$

where, as before,

$$bT\phi = \frac{a^2T}{b+\sqrt{a^2+b^2}} .$$

Then, for $a \le b$, W becomes

$$\mathrm{W(a,b,T)} = -\frac{\mathrm{e}^{-a^2T^2}\,\mathrm{ierfc(bT)}}{\mathrm{ab}\sqrt{\pi}} + \frac{1}{\mathrm{ab}\sqrt{\pi}}\frac{\mathrm{b}}{\sqrt{\mathrm{a}^2+\mathrm{b}^2}}\left[\mathrm{e}^{-a^2T^2}\,\mathrm{ierfc(bT)} + \mathrm{e}^{-a^2T^2}\sum_{n=2}^{\infty}(-2)^{n-1}\mathrm{n\ i^n erfc(bT)}(\mathrm{bT}\phi)^{n-1}\right]$$
$$+\mathrm{G}_9(a,b,T)$$

$$= \frac{\mathrm{e}^{-a^2T^2}\,\mathrm{ierfc(bT)}}{\mathrm{ab}\sqrt{\pi}}\left[-1+\frac{b}{\sqrt{\mathrm{a}^2+\mathrm{b}^2}}\right] + \frac{1}{\sqrt{\pi}}\frac{\mathrm{e}^{-a^2T^2}}{\sqrt{\mathrm{a}^2+\mathrm{b}^2}}\frac{aT}{b+\sqrt{\mathrm{a}^2+\mathrm{b}^2}}\sum_{n=2}^{\infty}(-2)^{n-1}\mathrm{n\ i^n erfc(bT)}(\mathrm{bT}\phi)^{n-2}$$
$$+\mathrm{G}_9(a,b,T)$$

$$\mathrm{W(a,b,T)} = \frac{(a/b)\mathrm{e}^{-a^2T^2}}{\sqrt{\pi}\sqrt{\mathrm{a}^2+\mathrm{b}^2}\,(b+\sqrt{\mathrm{a}^2+\mathrm{b}^2})}\left[-\mathrm{ierfc(bT)}+\mathrm{bT}\sum_{n=2}^{\infty}(-2)^{n-1}\mathrm{n\ i^n erfc(bT)}(\mathrm{bT}\phi)^{n-2}\right] + \mathrm{G}_9(a,b,T)$$

The final expression for $I_9^c(a,b,T)$ is

$$2I_9^c(a,b,T) \;=\; \frac{\mathrm{Terfc(aT)}}{\mathrm{b}} ierfc(bT) + \frac{1}{\mathrm{b}\sqrt{\pi}}\mathrm{I}_5(\mathrm{b,a,T}) + \mathrm{W(a,b,T)}, \qquad \mathrm{a} \le \mathrm{b}.$$

The convergence analysis for W is the same as before, with rapid convergence for $aT \le 1+\sqrt{2} < 2.4$.

The challenge now is to manipulate the expression for $I_9(a,b,T)$,

$$2I_9(a,b,T) = T^2 erf(aT)erf(bT) + \frac{Te^{-a^2T^2}erf(bT)}{a\sqrt{\pi}} + \frac{Te^{-b^2T^2}erf(aT)}{b\sqrt{\pi}}$$
$$-\frac{1}{a\sqrt{\pi}}V_5(a,b,T) - \frac{1}{b\sqrt{\pi}}V_5(b,a,T) - \frac{1}{\mathrm{ab}\pi}\left(1-e^{-T^2(a^2+b^2)}\right)$$

to remove the indeterminant form when a goes to zero. To this end, we re-group the terms,

$$2I_9(a,b,T) = T^2 erf(aT)erf(bT) + \frac{Te^{-b^2T^2}erf(aT)}{b\sqrt{\pi}} - \frac{1}{b\sqrt{\pi}}V_5(b,a,T) + U(a,b,T)$$

$$\mathrm{U(a,b,T)} = \frac{Te^{-a^2T^2}erf(bT)}{a\sqrt{\pi}} + \frac{e^{-T^2(a^2+b^2)}}{ab\pi} - \frac{1}{a\sqrt{\pi}}V_5(a,b,T) - \frac{1}{\mathrm{ab}\pi}$$

and work with U(a,b,T). Then

$$\frac{Te^{-a^2T^2}erf(bT)}{a\sqrt{\pi}} + \frac{e^{-T^2(a^2+b^2)}}{ab\pi} = \frac{e^{-a^2T^2}}{a\sqrt{\pi}}\left[\frac{bTerf(bT)}{b} + \frac{e^{-b^2T^2}}{b\sqrt{\pi}}\right] = \frac{e^{-a^2T^2}}{a\sqrt{\pi}}\left[T + \frac{1}{b}ierfc(bT)\right]$$

and from Folder 5d,

$$-\frac{1}{a\sqrt{\pi}}V_5(a,b,T) - \frac{1}{\mathrm{ab}\pi} = -\frac{1}{a\sqrt{\pi}}\left[\frac{\sqrt{\pi}}{2a}erf(aT) - \frac{1}{a\sqrt{\pi}}\tan^{-1}\left(\frac{a}{b}\right) + I_5(a,b,T)\right] - \frac{1}{ab\pi}$$

where, with $d^2 = a^2 + b^2$,

$$I_5(a,b,T) = \frac{ierfc(dT)}{d} + \frac{1}{2d\sqrt{\pi}}\sum_{k=1}^{\infty}\frac{(1/2)_k}{k!}\left(\frac{a^2}{d^2}\right)^k E_{k+3/2}(d^2T^2), \qquad d^2 = a^2 + b^2,$$

and

$$-\frac{1}{a\sqrt{\pi}}V_5(a,b,T) - \frac{1}{\mathrm{ab}\pi} = \left[-\frac{1}{2a^2}erf(aT) + \frac{1}{a^2\pi}\tan^{-1}\left(\frac{a}{b}\right) - \frac{1}{ad\sqrt{\pi}}ierfc(dT) - G_9(a,b,T) - \frac{1}{\mathrm{ab}\pi}\right]$$

where $G_9(a,b,T)$ was defined in the computation of $I_9^c(a,b,T)$ above. Then, U becomes

$$U(a,b,T) = H_1(a,b,T) + H_2(a,b,T) - W(a,b,T)$$

where

$$H_1(a,b,T) = \frac{1}{a\sqrt{\pi}}\left[Te^{-a^2T^2} - \frac{\sqrt{\pi}}{2a}erf(aT)\right], \qquad H_2(a,b,T) = \left[\frac{1}{a^2\pi}\tan^{-1}\left(\frac{a}{b}\right) - \frac{1}{ab\pi}\right]$$

$$W(a,b,T) = \frac{1}{a\sqrt{\pi}}\left[-\frac{e^{-a^2T^2}ierfc(bT)}{b} + \frac{1}{d}ierfc(dT)\right] + G_9(a,b,T)$$

and W is the function developed above for $I_9^c(a,b,T)$. Our goal now is to expand the terms in H_1 and H_2 into a proper series which will allow the cancellation of the reciprocal powers of a, remembering $a \le b$,

$$H_1(a,b,T)=\left[\frac{-T(1-e^{-a^2T^2})}{a\sqrt{\pi}}+\frac{T}{a\sqrt{\pi}}-\frac{1}{2a^2}erf(aT)\right]=\frac{T}{a\sqrt{\pi}}\sum_{k=1}^{\infty}\frac{(-1)^k(aT)^{2k}}{k!}-\frac{T}{a\sqrt{\pi}}\sum_{k=1}^{\infty}\frac{(-1)^k(aT)^{2k}}{k!(2k+1)}$$

$$=\frac{2\mathrm{T}^2}{\sqrt{\pi}}\sum_{k=1}^{\infty}\frac{(-1)^k(aT)^{2k-1}}{(k-1)!(2k+1)}$$

$$H_2(a,b,T)=\left[\frac{1}{a^2\pi}\tan^{-1}\left(\frac{a}{b}\right)-\frac{1}{ab\pi}\right]=\frac{1}{a^2\pi}\left[\sum_{k=1}^{\infty}\frac{(-1)^k(a/b)^{2k+1}}{(2k+1)}\right]=\frac{1}{b^2\pi}\sum_{k=1}^{\infty}\frac{(-1)^k(a/b)^{2k-1}}{(2k+1)},\quad \frac{\mathrm{a}}{\mathrm{b}}<1$$

$$\mathrm{W(a,b,T)}=\frac{(\mathrm{a/b})\mathrm{e}^{-\mathrm{a}^2\mathrm{T}^2}}{\sqrt{\pi}\sqrt{\mathrm{a}^2+\mathrm{b}^2}\,(b+\sqrt{\mathrm{a}^2+\mathrm{b}^2})}\left[-\mathrm{ierfc(bT)+bT}\sum_{\mathrm{n}=2}^{\infty}(-2)^{\mathrm{n}-1}\mathrm{n\ i^n erfc(bT)(bT}\phi)^{\mathrm{n}-2}\right]+\mathrm{G}_9(a,b,T)$$

$$G_9(a,b,T)=\frac{1}{2\pi d^2}\sum_{k=1}^{\infty}\frac{(1/2)_k}{k!}\left(\frac{a}{d}\right)^{2k-1}E_{k+3/2}\left(d^2T^2\right),\qquad d^2=a^2+b^2.$$

The final expression for $I_9(a,b,T)$ is

$$2I_9(a,b,T)=T^2erf(aT)erf(bT)+\frac{Te^{-b^2T^2}erf(aT)}{b\sqrt{\pi}}-\frac{1}{b\sqrt{\pi}}V_5(b,a,T)+U(a,b,T)$$

$$U(a,b,T)=H_1(a,b,T)+H_2(a,b,T)-W(a,b,T).$$

The numerical scheme is to use the closed form expressions when convergence of the series in powers of a is poor and the series in H_1, H_2 and W when the convergence is rapid. The numerical scheme outlined in Folder 9a is the model.

Special Cases

b=0 in_$\mathrm{I}_9^c(\mathrm{a,b,T})$**:** $\qquad I_9^c(a,0,T)=\int_T^{\infty}w\,erfc(aw)dw$**.**

Integration by parts or the recurrence relation for n=1, $aw\,erfc(aw)=\frac{e^{-a^2w^2}}{\sqrt{\pi}}-ierfc(aw)$, leads to

$$\mathrm{I}_9^c(\mathrm{a,0,T})=\frac{1}{\mathrm{a}^2}\left[\frac{1}{2}erfc(aT)-i^2erfc(aT)\right]=\frac{T}{a}ierfc(aT)+\frac{1}{a^2}i^2erfc(aT)$$

where we have used the recurrence relation for n=2,

$$erfc(aT)=2aT\,ierfc(aT)+4i^2erfc(aT),$$

to produce a better numerical form with no subtractions. Subroutine DINERFC computes sequences of iterated co-error functions while subroutine DIERFC can be used if a single value of ierfc(*) is needed.

$b \to \infty$ in $\mathrm{I}_9(a,b,T)$: $\quad \mathrm{I}_9(a,\infty,T) = \int_0^T w\, erf(aw)\, dw = \left[\frac{T^2}{2} - \frac{1}{4a^2}\right] erf(aT) + \frac{T}{a\sqrt{\pi}} e^{-a^2T^2}$

is found in tables [Beck et al., Table F.2, p.422].

Folder 10

Closed Form for I_1 of Folders 1 and 2 and Related Integrals

$$I_1(a,b,T) = \int_T^\infty \frac{e^{-a^2w^2}\mathrm{erf}(bw)}{w^2}dw = \frac{1}{2}\int_0^t \frac{e^{-a^2/\tau}\mathrm{erf}(b/\sqrt{\tau})}{\sqrt{\tau}}d\tau, \qquad T = \frac{1}{\sqrt{t}}$$

$$W_3(a,b,T) = \int_T^\infty \frac{\mathrm{erf}(aw)\mathrm{erf}(bw)}{w^3}dw = \frac{1}{2}\int_0^t \mathrm{erf}(a/\sqrt{\tau})\mathrm{erf}(b/\sqrt{\tau})d\tau$$

$$I_1^c(a,b,T) = \int_T^\infty \frac{e^{-a^2w^2}\mathrm{erfc}(bw)}{w^2}dw, \qquad W_3^c(a,b,T = \int_T^\infty \frac{\mathrm{erfc}(aw)\,\mathrm{erfc}(bw)}{w^3}dw$$

$$J_\nu(a,T) = \int_T^\infty \frac{\mathrm{erf}(aw)}{w^\nu}dw,\ \nu > 1, \qquad J_\nu^c(a,T) = \int_T^\infty \frac{\mathrm{erfc}(aw)}{w^\nu}dw,\ \nu \neq 1$$

where the superscript *c* denotes the replacement of erf with erfc. (See Folders 3, 6, 11 for similar pairs)

Donald E. Amos, September 2001, March 2003. March 2006

Summary

Folder 10 is composed of Folders 10a and 10b. Folder 10a deals with the basic integrals which contain erf functions. Folder 10b extends the results of Folder 10a to the integrals where erfc replaces erf. These are called complementary functions and denoted with a superscript *c*. Folder 10a can be summarized as follows:

$$I_1(a,b,T) = \frac{e^{-a^2T^2}\mathrm{erf}(bT)}{T} + \frac{b}{\sqrt{\pi}}E_1(X) - 2a^2 J_5(a,b,T), \qquad X = (a^2+b^2)\mathrm{T}^2$$

$$I_1(a,b,T) = \frac{e^{-a^2T^2}\,erf(bT)}{T} + \frac{b}{\sqrt{\pi}}E_1(X) - a\sqrt{\pi}erfc(aT) + 2a^2 I_5(a,b,T)$$

$$= -\frac{e^{-a^2T^2}\,erfc(bT)}{T} + \frac{b}{\sqrt{\pi}}E_1(X) + \frac{\sqrt{\pi}}{T}ierfc(aT) + 2a^2 I_5(a,b,T)$$

$$I_1(a,b,T) = \frac{\sqrt{\pi}}{T}\mathrm{erf}(bT)i\,\mathrm{erfc}(aT) + \frac{b}{\sqrt{\pi}}E_1(X) - 2abI_5(b,a,T)$$

are derived where J_5 and I_5 are the J and I integrals of Folder 5,

$$J_5(a,b,T) = J(a,b,T) = \int_T^\infty e^{-a^2x^2}\,\mathrm{erf}(bx)dx,$$

$$I_5(a,b,T) = I(a,b,T) = \int_T^\infty e^{-a^2x^2}\,\mathrm{erfc}(bx)dx$$

These formulae for I_1 are analyzed for numerical computation.

The integrals

$$W_3 = \int_T^\infty \frac{\operatorname{erf}(ax)\operatorname{erf}(bx)}{x^3}dx = \frac{1}{2}\int_0^t \operatorname{erf}(a/\sqrt{\tau})\operatorname{erf}(b/\sqrt{\tau})d\tau\,,$$
$$W_3 = \frac{\operatorname{erf}(aT)\operatorname{erf}(bT)}{2T^2} + \frac{1}{\sqrt{\pi}}[aI_1(a,b,T) + bI_1(b,a,T)]$$

$$W_3^c = \int_T^\infty \frac{erfc(ax)erfc(bx)}{x^3}dx = \frac{1}{2}\int_0^t erfc(a/\sqrt{\tau})erfc(b/\sqrt{\tau})d\tau$$
$$W_3^c = \frac{erfc(aT)erfc(bT)}{2T^2} - \frac{1}{\sqrt{\pi}}[aI_1^c(a,b,T) + bI_1^c(b,a,T)]$$

are also derived.

Folder 10b can be summarized similarly:

$$I_1^c(a,b,T) = \frac{e^{-a^2T^2}\operatorname{erfc}(bT)}{T} - \frac{b}{\sqrt{\pi}}E_1(X) - 2a^2I_5(a,b,T)\,, \qquad X = (a^2+b^2)\mathrm{T}^2$$
$$I_1^c(a,b,T) = \frac{\sqrt{\pi}}{T}\operatorname{erfc}(bT)\operatorname{ierfc}(aT) - \frac{b}{\sqrt{\pi}}E_1(X) + 2abI_5(b,a,T)$$

$$W_3^c(a,b,T) = \frac{\operatorname{erfc}(aT)\operatorname{erfc}(bT)}{2T^2} - \frac{1}{\sqrt{\pi}}[aI_1^c(a,b,T) + bI_1^c(b,a,T)]$$

$$J_3(a,T) = \frac{\operatorname{erf}(aT)}{2T^2} + \frac{a}{T}i\operatorname{erfc}(aT)$$

$$J_3^c(a,T) = \frac{\operatorname{erfc}(aT)}{2T^2} - \frac{a}{T}i\operatorname{erfc}(aT) = \frac{2i^2\operatorname{erfc}(aT)}{T^2}$$

$$I_1(a,b,T) + I_1^c(a,b,T) = \frac{\sqrt{\pi}}{T}i\operatorname{erfc}(aT), \qquad J_3(a,T) + J_3^c(a,T) = \frac{1}{2T^2}$$

Folder 10a

Computation of Basic Functions

$$\mathbf{I_1(a,b,T)=\int_T^\infty \frac{e^{-a^2w^2}erf(bw)}{w^2}dw,\quad W_3(a,b,T)=\int_T^\infty \frac{erf(aw)erf(bw)}{w^3}dw}$$

$$\boldsymbol{a>0,\quad b>0,\quad T>0}$$

Donald E. Amos, March 2003, March 2006

Derivations

Integrate I_1 by parts $\quad I_1(a,b,T)=\int_T^\infty \frac{e^{-a^2w^2}\,\mathrm{erf}(bw)}{w^2}dw$

$$u=e^{-a^2w^2}\,\mathrm{erf}(bw) \qquad\qquad dv=\frac{dw}{w^2}$$

$$du=\left[\frac{2b}{\sqrt{\pi}}e^{-(a^2+b^2)w^2}-2a^2we^{-a^2w^2}\,\mathrm{erf}(bw)\right]dw \qquad v=\frac{-1}{w}$$

Then,

$$I_1(a,b,T)=\frac{e^{-a^2T^2}\,\mathrm{erf}(bT)}{T}+\frac{2b}{\sqrt{\pi}}\int_T^\infty \frac{e^{-(a^2+b^2)w^2}}{w}dw-2a^2\int_T^\infty e^{-a^2w^2}\,\mathrm{erf}(bw)dw$$

$$=\frac{e^{-a^2T}\,\mathrm{erf}(bT)}{T}+\frac{b}{\sqrt{\pi}}E_1[T^2(a^2+b^2)]-2a^2J_5(a,b,T)$$

where J_5 is the J integral of Folder 5,

$$J_5(a,b,T)=\int_T^\infty e^{-a^2x^2}\,\mathrm{erf}(bx)dx.$$

Since J_5 is also related to I_5,

$$J_5(a,b,T)=\frac{\sqrt{\pi}}{2a}\mathrm{erfc}(aT)-I_5(a,b,T),$$

we can write

$$I_1(a,b,T)=\frac{e^{-a^2T^2}erf(bT)}{T}+\frac{b}{\sqrt{\pi}}E_1[T^2(a^2+b^2)]-a\sqrt{\pi}erfc(aT)+2a^2I_5(a,b,T)$$

$$=-\frac{e^{-a^2T^2}erfc(bT)}{T}+\frac{b}{\sqrt{\pi}}E_1[T^2(a^2+b^2)]+\frac{\sqrt{\pi}}{T}ierfc(aT)+2a^2I_5(a,b,T).$$

Using the relations (Folder 5)

$$I_5(a,b,T)=\frac{\sqrt{\pi}}{2a}\mathrm{erfc}(aT)\,\mathrm{erfc}(bT)-\frac{b}{a}I_5(b,a,T)$$

$$\mathrm{ierfc}(x)=\frac{e^{-x^2}}{\sqrt{\pi}}-x\,\mathrm{erfc}(x),\quad \text{x=aT},$$

gives the form for $b \to 0$

$$I_1(a,b,T) = \frac{\sqrt{\pi}}{T}\operatorname{erf}(bT)\operatorname{ierfc}(aT) + \frac{b}{\sqrt{\pi}}E_1[T^2(a^2+b^2)] - 2abI_5(b,a,T).$$

Computational Forms

Notice that

$$\operatorname{erf}(bw) = 1 + O(10^{-16})$$

uniformly for $bw \geq 6$, and

$$I_1(a,b,T) \cong \int_T^\infty \frac{e^{-a^2w^2}}{w^2}dw = \frac{1}{2T}E_{3/2}(a^2T^2) = \frac{\sqrt{\pi}}{T}\operatorname{ierfc}(aT), \qquad bT \geq 6$$

with a relative error $O(10^{-16})$ for $bT \geq 6$. This test for $bT \geq 6$ is applied first to screen out many underflow cases since $E_{3/2}$ or *i*erfc functions have underflow tests in these routines.

Notice also that

$$D = \frac{e^{-a^2T^2}}{T} - a\sqrt{\pi}\operatorname{erfc}(aT) = \frac{e^{-a^2T^2}}{T}[1 - aT\sqrt{\pi}e^{a^2T^2}\operatorname{erfc}(aT)] = \frac{\sqrt{\pi}}{T}i\operatorname{erfc}(aT)$$

occurs in the second form for $I_1(a,b,T)$ where bT is large and erf(bT) is close to one. The asymptotic form, with $z = aT$,

$$e^{z^2}\operatorname{erfc}(z) = \frac{1}{z\sqrt{\pi}}\left[1 + \sum_{k=1}^{K}\frac{(-1)^k(1/2)_k}{z^{2k}} + R_K\right] \text{ where } |R_K| \leq |\operatorname{term}(K+1)|$$

shows that the difference $Te^{z^2}D$ is $O(1/(2a^2T^2))$ and for $z = aT \geq 7$ we will lose at least 2 significant digits. Therefore we do the subtraction analytically and stop the sum at term K where $|\operatorname{term}(K+1)|$ is less than 10^{-16} or when the minimum term (in magnitude) is reached. The $|\operatorname{term}(K+1)|$ or $|\text{minimum term}|$ is then a bound on the error in summing the previous terms. For $aT \geq 7$ we need no more than 21 terms for relative errors $O(10^{-15})$ ($aT = 7$ is the worst case and the relative error test is always met). This form is implemented in subroutine DIERFC.

Evaluation of $W_3 = \int_T^\infty \frac{\operatorname{erf}(aw)\operatorname{erf}(bw)}{w^3}dw$

Integrate W_3 by parts:

$u = \operatorname{erf}(aw)\operatorname{erf}(bw)$ $\qquad\qquad$ $dv = dw/w^3$

$$du = \frac{2}{\sqrt{\pi}}[ae^{-a^2w^2}\operatorname{erf}(bw) + be^{-b^2w^2}\operatorname{erf}(aw)]dw \qquad v = -1/(2w^2)$$

Then,

$$W_3 = \frac{\mathrm{erf}(aT)\,\mathrm{erf}(bT)}{2T^2} + \frac{1}{\sqrt{\pi}}[aI_1(a,b,T) + bI_1(b,a,T)]$$

<u>Special Cases,</u> $T > 0$

$b = 0$ $\qquad I_1(a,0,T) = 0$,

$a = 0$ $\qquad I_1(0,b,T) = \int_T^\infty \frac{\mathrm{erf}(bw)}{w^2}\,dw$

Integration by parts gives

$$I_1(0,b,T) = \frac{\mathrm{erf}(bT)}{T} + \frac{b}{\sqrt{\pi}}E_1(b^2T^2), \qquad a = 0$$

<u>Quadrature Procedure for W_3</u>

We follow the format of Folder 3 where we replace the erf functions by 1 when the argument exceeds 6. As noted, the error is uniformly $O(10^{-16})$ for arguments 6 or larger. Let

$$X = \min(a,b), \qquad W_m = \min\left(\frac{6}{a},\frac{6}{b}\right), \qquad W_M = \max\left(\frac{6}{a},\frac{6}{b}\right)$$

<u>Case I, $T \le W_m$</u>

$$W_3 = \int_T^{W_m} \frac{\mathrm{erf}(aw)\,\mathrm{erf}(bw)}{w^3}\,dw + \int_{W_m}^{W_M} \frac{\mathrm{erf}(Xw)}{w^3}\,dw + \int_{W_M}^\infty \frac{dw}{w^3}$$

<u>Case II, $W_m < T \le W_M$</u>

$$W_3 = \int_T^{W_M} \frac{\mathrm{erf}(Xw)}{w^3}\,dw + \int_{W_M}^\infty \frac{dw}{w^3}$$

<u>Case III, $W_M < T < \infty$</u>

$$W_3 = \int_T^\infty \frac{dw}{w^3}$$

Now

$$\int_A^\infty \frac{dw}{w^3} = \frac{1}{2A^2} \qquad \text{and} \qquad P(X,L,U) = \int_L^U \frac{\mathrm{erf}(Xw)}{w^3}\,dw$$

can be integrated by parts:

$$u = \text{erf}(Xw) \qquad\qquad dv = dw/w^3$$

$$du = \frac{2X}{\sqrt{\pi}} e^{-X^2 w^2} dw \qquad\qquad v = -\frac{1}{2w^2}$$

$$P(X,L,U) = \frac{\text{erf}(XL)}{2L^2} - \frac{\text{erf}(XU)}{2U^2} + \frac{X}{\sqrt{\pi}}\left[\int_L^\infty \frac{e^{-X^2 w^2}}{w^2} dw - \int_U^\infty \frac{e^{-X^2 w^w}}{w^2} dw\right]$$

Now,

$$\int_L^\infty \frac{e^{-X^2 w^2}}{w^2} dw = \frac{1}{2L}\int_1^\infty \frac{e^{-X^2 L^2 u}}{u^{3/2}} du = \frac{1}{2L} E_{3/2}(X^2 L^2) = \frac{\sqrt{\pi}}{L} i\,\text{erfc}(XL)$$

$$P(X,L,U) = \frac{\text{erf}(XL)}{2L^2} - \frac{\text{erf}(XU)}{2U^2} + X\left[\frac{i\,\text{erfc}(XL)}{L} - \frac{i\,\text{erfc}(XU)}{U}\right]$$

The final result is

Case I, $T \le W_m$

$$W_3 = \int_T^{W_m} \frac{\text{erf}(aw)\,\text{erf}(bw)}{w^3} dw + P(X, W_m, W_M) + \frac{1}{2W_M^2}$$

Case II, $W_m < T \le W_M$

$$W_3 = P(X, T, W_M) + \frac{1}{2W_M^2}$$

Case III, $W_M < T < \infty$

$$W_3 = \frac{1}{2T^2}$$

with

$$P(X,L,W_M) + \frac{1}{2W_M^2} = \frac{erf(XL)}{2L^2} + \frac{erfc(XW_M)}{2W_M^2} + X\left[\frac{ierfc(XL)}{L} - \frac{ierfc(XW_M)}{W_M}\right]$$

and L=W_m or L=T in Cases I and II.

Numerical Experiments

Numerical evaluation of the formulae in terms of J_5 and I_5, when compared with the quadrature, gave maximum errors $O(10^{-11})$, with the J_5 formula being slightly better in some cases. To bring the accuracy to full double precision arithmetic, $O(10^{-14})$ or better, we probably need to combine the leading terms with those from J_5 or I_5 and do analytic reductions in order eliminate losses of significance.

Folder 10b

Computation of Complementary Functions

$$\mathbf{I_1^c(a,b,T)=\int_T^\infty \frac{e^{-a^2w^2}\,erfc(bw)}{w^2}dw\,,\qquad W_3^c(a,b,T)=\int_T^\infty \frac{erfc(aw)\,erfc(bw)}{w^3}dw}$$

$$\mathbf{J_\nu(a,T)=\int_T^\infty \frac{erf(aw)}{w^\nu}dw\,,\ \nu>1\,,\qquad J_\nu^c(a,T)=\int_T^\infty \frac{erfc(aw)}{w^\nu}dw\,,\ \nu\neq 1}$$

$$\boldsymbol{a>0,\quad b>0,\quad T>0}$$

Donald E. Amos, March 2003, March 2006

Derivations

We start with $I_1^c(a,b,T)$

$$I_1^c(a,b,T)=\int_T^\infty \frac{e^{-a^2w^2}\,\mathrm{erfc}(bw)}{w^2}dw$$

which is complementary to $I_1(a,b,T)$ of Folder 10a,

$$I_1(a,b,T)=\int_T^\infty \frac{e^{-a^2w^2}\,\mathrm{erf}(bw)}{w^2}dw$$

and

$$I_1(a,b,T)+I_1^c(a,b,T)=\int_T^\infty \frac{e^{-a^2w^2}}{w^2}dw=\frac{\sqrt{\pi}}{T}\mathrm{ierfc}(aT)$$

Integration of $I_1^c(a,b,T)$ by parts gives the explicit expression similar to $I_1(a,b,T)$,

$$I_1^c(a,b,T)=\frac{e^{-a^2T^2}\,\mathrm{erfc}(bT)}{T}-\frac{b}{\sqrt{\pi}}E_1(X)-2a^2I_5(a,b,T)\,,\qquad T=\frac{1}{\sqrt{\mathrm{t}}},\qquad \mathrm{X}=(a^2+b^2)T^2$$

$$\begin{aligned}I_1(a,b,T)&=\frac{e^{-a^2T^2}\,erf(bT)}{T}+\frac{b}{\sqrt{\pi}}E_1(X)+2a^2I_5(a,b,T)-a\sqrt{\pi}erfc(aT)\\&=-\frac{e^{-a^2T^2}\,erfc(bT)}{T}+\frac{b}{\sqrt{\pi}}E_1(X)+\frac{\sqrt{\pi}}{T}ierfc(aT)+2a^2I_5(a,b,T)\end{aligned}$$

and the form for $b\to 0$ is

$$I_1^c(a,b,T)=\frac{\sqrt{\pi}}{T}\mathrm{erfc}(bT)\mathrm{ierfc}(aT)-\frac{b}{\sqrt{\pi}}E_1(X)+2abI_5(b,a,T)$$

where $I_5(a,b,T)$ is the I integral of Folder 5.

We also note that by integration by parts,

$$J_\nu(a,T) = \int_T^\infty \frac{\operatorname{erf}(aw)}{w^\nu} dw = \frac{1}{(\nu-1)T^{\nu-1}} \left[\operatorname{erf}(aT) + \frac{aT}{\sqrt{\pi}} E_{\nu/2}(a^2T^2) \right], \qquad \nu > 1$$

$$J_\nu^c(a,T) = \int_T^\infty \frac{\operatorname{erfc}(aw)}{w^\nu} dw = \frac{1}{(\nu-1)T^{\nu-1}} \left[\operatorname{erfc}(aT) - \frac{aT}{\sqrt{\pi}} E_{\nu/2}(a^2T^2) \right], \qquad \nu \neq 1$$

$$J_3(a,T) = \int_T^\infty \frac{\operatorname{erf}(aw)}{w^3} dw = \frac{\operatorname{erf}(aT)}{2T^2} + \frac{a}{T} i \operatorname{erfc}(aT)$$

$$J_3^c(a,T) = \int_T^\infty \frac{\operatorname{erfc}(aw)}{w^3} dw = \frac{\operatorname{erfc}(aT)}{2T^2} - \frac{a}{T} i \operatorname{erfc}(aT) = \frac{2i^2 \operatorname{erfc}(aT)}{T^2}$$

and

$$J_\nu(a,T) + J_\nu^c(a,T) = \frac{1}{(\nu-1)T^{\nu-1}} \qquad \nu > 1.$$

Similarly for $W_3(a,b,T)$ and $W_3^c(a,b,T)$, we integrate by parts to obtain

$$W_3(a,b,T) = \frac{\operatorname{erf}(aT)\operatorname{erf}(bT)}{2T^2} + \frac{1}{\sqrt{\pi}}[aI_1(a,b,T) + bI_1(b,a,T)]$$

$$W_3^c(a,b,T) = \frac{\operatorname{erfc}(aT)\operatorname{erfc}(bT)}{2T^2} - \frac{1}{\sqrt{\pi}}[aI_1^c(a,b,T) + bI_1^c(b,a,T)]$$

Using the relation erfc(x) = 1–erf(x) with

erfc(aw)erfc(bw) = 1–erf(aw)–erf(bw)+erf(aw)erf(bw)

we get

$$W_3^c(a,b,T) = \frac{1}{2T^2} - J_3(a,T) - J_3(b,T) + W_3(a,b,T)$$

and

$$W_3^c(a,b,T) - W_3(a,b,T) = \frac{1}{2T^2} - J_3(a,T) - J_3(b,T) = J_3^c(a,T) - J_3(b,T)$$

Special Cases, $T > 0$

$b = 0$ $$I_1^c(a,0,T) = \int_T^\infty \frac{e^{-a^2w^2}}{w^2} dw = \frac{1}{2T} E_{3/2}(a^2T^2) = \frac{\sqrt{\pi}}{T} \operatorname{ierfc}(aT),$$

$a = 0$ $$I_1^c(0,b,T) = \int_T^\infty \frac{\operatorname{erfc}(bw)}{w^2} dw = J_2(bT) = \frac{\operatorname{erfc}(bT)}{T} - \frac{b}{\sqrt{\pi}} E_1(b^2T^2)$$

Computation of $I_1^c(a,b,T)$ with the Functions of Folder 21

We note that erfc(bw) can be written in terms of $i$erfc(bw) and i^2erfc(bw) from the recurrence for iterated co-error functions i^nerfc(bw):

$$\text{erfc}(bw) = 2bw\,i\text{erfc}(bw)+4i^2\text{erfc}(bw)$$

and the definition of $I_1^c(a,b,T)$ gives

$$I_1^c(a,b,T) = 2bG_1(a,b,T) + 4G_2(a,b,T) = e^{-X}[by_1(a,b,T) + \frac{2}{T}y_2(a,b,T)]$$

where G_n and y_n are defined in Folder 21 as

$$G_n(a,b,T) = \int_T^\infty e^{-a^2w^2}\frac{\text{i}^n\text{erfc}(bw)}{w^n}dw, \qquad , \qquad n = 0.1.2\ldots,$$

$$y_n(a,b,T) = 2T^{n-1}e^X G_n(a,b,T), \qquad \text{X} = (a^2+b^2)T^2.$$

The y_n sequence can be generated from subroutine GNSEQ. Notice that the y_n's are properly scaled and an underflow test can be made on X. Notice also that y_1 and y_2 are computed to full significance and the addition of positive quantities retains full significance (relative error).

Folder 11

Evaluation of

$$P=\int_T^\infty \frac{e^{-a^2w^2}\,\mathrm{erf}(bw)}{w}dw\,,\ \mathrm{P^c}=\int_T^\infty \frac{e^{-a^2w^2}\,\mathrm{erfc}(bw)}{w}dw\,,\ Q=\int_T^\infty e^{-a^2w^2}E_1(b^2w^2)dw$$

$$a>0,\quad b>0,\quad T>0$$

Donald E. Amos, September 2001

Summary

The expressions

$$P(a,b,T)=\begin{cases}\dfrac{1}{2}E_1(a^2T^2)-G\left(\sqrt{X}\right)-\ln\left[\dfrac{2\sqrt{a^2+b^2}}{b+\sqrt{a^2+b^2}}\right]\mathrm{erfc}(\sqrt{X})+\dfrac{\sqrt{X}}{2\sqrt{\pi}}S_1(a,b,X) & a\le b\\[2ex] \dfrac{1}{2}E_1(a^2T^2)\,\mathrm{erf}(bT)+\ln\left[\dfrac{b+\sqrt{a^2+b^2}}{a}\right]\mathrm{erfc}(\sqrt{X})-\dfrac{bT}{\sqrt{\pi}}S_2(b,a,X) & a>b\end{cases}$$

$$Q(a,b,T)=\begin{cases}\dfrac{\sqrt{\pi}}{a}\ln\left[\dfrac{a+\sqrt{a^2+b^2}}{b}\right]\mathrm{erfc}(\sqrt{X})-TS_2(a,b,X) & a\le b\\[2ex] \dfrac{\sqrt{\pi}}{2a}E_1(b^2T^2)\,\mathrm{erfc}(aT)-\dfrac{\sqrt{\pi}}{a}\ln\left[\dfrac{2\sqrt{a^2+b^2}}{a+\sqrt{a^2+b^2}}\right]\mathrm{erfc}(\sqrt{X})-\dfrac{\sqrt{\pi}}{a}G\left(\sqrt{X}\right) & \\ \qquad +\dfrac{\sqrt{X}}{2a}S_1(b,a,X) & a>b\end{cases}$$

are derived where $X=T^2(a^2+b^2)$,

$$S_1(a,b,X)=\sum_{k=1}^{\infty}C_k\left(\frac{a^2}{a^2+b^2}\right)^k\frac{E_{k+1/2}(X)}{k},\qquad S_2(a,b,X)=\sum_{k=0}^{\infty}\left(\frac{a^2}{a^2+b^2}\right)^k\frac{E_{k+1}(X)}{2k+1}$$

and the computation of

$$G\left(\sqrt{X}\right) \equiv \int_{\sqrt{X}}^{\infty} \frac{\text{erfc}(x)}{x} dx = \frac{1}{2} E_1(X) - e^{-X} \ln 2 + \frac{X}{2\pi} \sum_{k=0}^{\infty} \frac{C_k}{(k+1/2)^2} E_{k+1/2}(X), \qquad C_k = \frac{(1/2)_k}{k!},$$

is described in Folders 6 and 16. Notice that whenever S_1 and S_2 are used, the powers of $a^2/(a^2+b^2)$ or $b^2/(a^2+b^2)$ are less than (or equal) to $1/2^k$, making the convergence of these series quite acceptable numerically. The special cases

$$P(a,b,0) = \ln\left[\frac{b+\sqrt{a^2+b^2}}{a}\right], \qquad Q(a,b,0) = \frac{\sqrt{\pi}}{a} \ln\left[\frac{a+\sqrt{a^2+b^2}}{b}\right]$$

$$P(a,0,T) = 0, \qquad Q(0,b,T) = \frac{\sqrt{\pi}}{b} \text{erfc}(bT) - TE_1(b^2T^2)$$

follow and integration by parts on P gives

$$P(a,b,T) = \frac{1}{2} E_1(a^2T^2) \text{erf}(bT) + \frac{b}{\sqrt{\pi}} Q(b,a,T).$$

We also have, using erfc(x)=1-erf(x),

$$P^c(a,b,T) = \int_T^{\infty} \frac{e^{-a^2w^2} erfc(bw)}{w} dw = \frac{1}{2} E_1(a^2T^2) - P(a,b,T) = \frac{1}{2} E_1(a^2T^2) erfc(bT) - \frac{b}{\sqrt{\pi}} Q(b,a,T)$$

Asymptotic expansions for large aT and bT are also derived

<u>Derivation for *P(a,b,T)*</u> Integrate P by parts

$$u = \frac{\text{erf}(bw)}{w} \qquad dv = e^{-a^2w^2} dw$$

$$du = \left[-\frac{\text{erf}(bw)}{w^2} + \frac{2b}{\sqrt{\pi}} \frac{e^{-b^2w^2}}{w} \right] dw \qquad v = \frac{\sqrt{\pi}}{2a} \text{erf}(aw)$$

$$P(a,b,T) = -\frac{\sqrt{\pi}}{2a} \frac{\text{erf}(aT)\,\text{erf}(bT)}{T} + \frac{\sqrt{\pi}}{2a} \int_T^{\infty} \frac{\text{erf}(aw)\,\text{erf}(bw)}{w^2} dw - \frac{b}{a} P(b,a,T)$$

Now, the difference

$$D(a,b,T) = \int_T^{\infty} \frac{\text{erf}(aw)\,\text{erf}(bw)}{w^2} dw - \frac{\text{erf}(aT)\,\text{erf}(bT)}{T}$$
$$= I_6(a,b,T) - \frac{\text{erf}(aT)\,\text{erf}(bT)}{T} = D(b,a,T)$$

can be computed directly from the (non-symmetric) form from Folder 6, which for $a \le b$, is

$$D(a,b,T) = \frac{a}{\sqrt{\pi}} E_1(a^2T^2) + \frac{b}{\sqrt{\pi}} E_1(b^2T^2)\,\mathrm{erf}(aT)$$
$$+ \frac{2}{\sqrt{\pi}}\left[b \ln\left(\frac{a+\sqrt{a^2+b^2}}{b} \right) - a \ln\left(\frac{2\sqrt{a^2+b^2}}{b+\sqrt{a^2+b^2}} \right) \right] \mathrm{erfc}(\sqrt{X})$$
$$- \frac{2a}{\sqrt{\pi}} G\left(\sqrt{X}\right) + \frac{a\sqrt{X}}{\pi} S_1(a,b,X) - \frac{2abT}{\pi} S_2(a,b,X)$$

where $X = T^2(a^2+b^2)$,

$$S_1(a,b,X) = \sum_{k=1}^{\infty} C_k \left(\frac{a^2}{a^2+b^2} \right)^k \frac{E_{k+1/2}(X)}{k}, \qquad S_2(a,b,X) = \sum_{k=0}^{\infty} \left(\frac{a^2}{a^2+b^2} \right)^k \frac{E_{k+1}(X)}{2k+1},$$

$$G\left(\sqrt{X}\right) \equiv \int_{\sqrt{X}}^{\infty} \frac{\mathrm{erfc}(x)}{x} dx = \frac{1}{2} E_1(X) - e^{-X} \ln 2 + \frac{X}{2\pi} \sum_{k=0}^{\infty} \frac{C_k}{(k+1/2)^2} E_{k+1/2}(X), \qquad C_k = \frac{(1/2)_k}{k!}.$$

A similar expression for $a > b$ exists by exchanging a and b since $D(a,b,T)$ is symmetric in a and b. From Folder 6 we also have, $X = T^2(a^2 + b^2)$,

$$P(a,b,T) = \frac{1}{2} E_1(a^2T^2) - G\left(\sqrt{X}\right) - \ln\left[\frac{2\sqrt{a^2+b^2}}{b+\sqrt{a^2+b^2}} \right] \mathrm{erfc}(\sqrt{X}) + \frac{\sqrt{X}}{2\sqrt{\pi}} S_1(a,b,X) \qquad a \le b$$

and with $P^c(a,b,T) = \frac{1}{2} E_1(a^2T^2) - P(a,b,T)$ we have

$$P^c(a,b,T) = G\left(\sqrt{X}\right) + \ln\left[\frac{2\sqrt{a^2+b^2}}{b+\sqrt{a^2+b^2}} \right] \mathrm{erfc}(\sqrt{X}) - \frac{\sqrt{X}}{2\sqrt{\pi}} S_1(a,b,X) \qquad a \le b$$

For $a > b$ we use the P by parts expression above

$$P(a,b,T) = \frac{\sqrt{\pi}}{2a} D(b,a,T) - \frac{b}{a} P(b,a,T) \qquad a > b$$

with $D(a,b,T) = D(b,a,T)$. Explicitly,

$$P(a,b,T) = \frac{1}{2} E_1(a^2T^2)\,\mathrm{erf}(bT) + \ln\left[\frac{b+\sqrt{a^2+b^2}}{a} \right] \mathrm{erfc}(\sqrt{X}) - \frac{bT}{\sqrt{\pi}} S_2(b,a,X) \qquad a > b$$

$$P^c(a,b,T) = \frac{1}{2} E_1(a^2T^2)\, erfc(bT) - \ln\left[\frac{b+\sqrt{a^2+b^2}}{a} \right] erfc(\sqrt{X}) + \frac{bT}{\sqrt{\pi}} S_2(b,a,X) \qquad a > b$$

The computation of $G\left(\sqrt{X}\right)=\int_{\sqrt{X}}^{\infty}\frac{\operatorname{erfc}(x)}{x}dx$ is described in Folder 6, where the convergence of the series is accelerated. A faster, more accurate method is described in Folder 16.

Special Cases

For $b = 0$ we have $P(a,0,T) = 0$ (expression for $a > b \geq 0$ or the definition of P)

For $T = 0$ we have $P(a,b,0) = \ln\left[\frac{b+\sqrt{a^2+b^2}}{a}\right]$ from the expression for $a > b$. For $a \leq b$, we take the limit as $T \to 0$ with $X=T^2(a^2 + b^2)$,

$$P(a,b,0) = \lim_{T\to 0}\left[\frac{1}{2}E_1(a^2T^2)-\frac{1}{2}E_1(X)\right]+\ln 2-\ln\left[\frac{2\sqrt{a^2+b^2}}{b+\sqrt{a^2+b^2}}\right].$$

Now $E_1(z) \sim -\gamma-\ln z+z+\ldots$ for $z \to 0$, and with $X = T^2(a^2 + b^2)$ we get

$$P(a,b,0) = \ln\left[\frac{b+\sqrt{a^2+b^2}}{a}\right].$$

For $a \to 0$ we get

$$\begin{aligned} P(a,b,T) &= \frac{1}{2}E_1(a^2T^2)-G(bT)+O(a^2) \\ &= -\frac{\gamma}{2}-\ln(aT)-G(bT)+O(a^2) \end{aligned} \qquad a \to 0.$$

Derivations for $Q(a,b,T)$ In Folder 6 we developed the relation, $X = T^2(a^2 + b^2)$,

$$Q(a,b,T) = \frac{\sqrt{\pi}}{a}\operatorname{erfc}(\sqrt{X})\ln\left[\frac{a+\sqrt{a^2+b^2}}{b}\right]-TS_2(a,b,X) \qquad a \leq b$$

where $S_2(a,b,X)$ for $a \leq b$ is defined above. The relation

$$Q(a,b,T)+Q(b,a,T) = \frac{\pi}{2ab}D(a,b,T)-\frac{\sqrt{\pi}}{2}\left[\frac{1}{b}E_1(a^2T^2)\operatorname{erf}(bT)+\frac{1}{a}E_1(b^2T^2)\operatorname{erf}(aT)\right]$$

is also derived from the symmetric form of $I_6(a,b,T)$ in Folder 6. This means that, for $a > b$, $Q(a,b,T)$ can be computed from

$$Q(a,b,T) = \frac{\pi}{2ab}D(b,a,T)-\frac{\sqrt{\pi}}{2}\left[\frac{1}{b}E_1(a^2T^2)\operatorname{erf}(bT)+\frac{1}{a}E_1(b^2T^2)\operatorname{erf}(aT)\right]-Q(b,a,T) \quad a > b$$

since $Q(b,a,T)$ has the power $[b^2/(a^2 + b^2)]^k \leq 1/2^k$ in $S_2(b,a,X)$ and $D(a,b,T) = D(b,a,T)$. Then, explicitly,

$$Q(a,b,T) = \frac{\sqrt{\pi}}{2a} E_1(b^2T^2)\,\mathrm{erfc}(aT) - \frac{\sqrt{\pi}}{a} \ln\left[\frac{2\sqrt{a^2+b^2}}{a+\sqrt{a^2+b^2}}\right] \mathrm{erfc}(\sqrt{X})$$
$$- \frac{\sqrt{\pi}}{a} \int_{\sqrt{X}}^{\infty} \frac{\mathrm{erfc}(x)}{x} dx + \frac{\sqrt{X}}{2a} S_1(b,a,X) \qquad a > b.$$

Special Case, $T = 0$

For the case $a \le b$, we have $Q(a,b,0) = \frac{\sqrt{\pi}}{a} \ln\left[\frac{a+\sqrt{a^2+b^2}}{b}\right]$.

For $a > b$ we take the limit as $T \to 0$, $X = T^2(a^2 + b^2)$,

$$Q(a,b,0) = \lim_{T\to 0}\left[\frac{\sqrt{\pi}}{2a} E_1(b^2T^2) - \frac{\sqrt{\pi}}{2a} E_1(X) - \ln 2\right] - \frac{\sqrt{\pi}}{a} \ln\left[\frac{2\sqrt{a^2+b^2}}{a+\sqrt{a^2+b^2}}\right]$$
$$= \frac{\sqrt{\pi}}{a} \ln\left[\frac{2\sqrt{a^2+b^2}}{b}\right] - \frac{\sqrt{\pi}}{a} \ln\left[\frac{2\sqrt{a^2+b^2}}{a+\sqrt{a^2+b^2}}\right]$$
$$= \frac{\sqrt{\pi}}{a} \ln\left[\frac{a+\sqrt{a^2+b^2}}{b}\right]$$

using $E_1(z) \sim -\gamma - \ln z + z + \ldots$ for $z \to 0$.

Special Case, $a = 0$

$$Q(0,b,T) = \int_T^{\infty} E_1(b^2w^2)\,dw = \frac{1}{2b} \int_{b^2T^2}^{\infty} \frac{E_1(u)}{\sqrt{u}} du$$

with a change of variables $u = b^2w^2$.

This integral is known,

$$\int_X^{\infty} \frac{E_n(u)}{u^{\nu}} du = \frac{1}{X^{\nu-1}} \frac{[E_{\nu}(X) - E_n(X)]}{n-\nu}, \qquad n \ne \nu$$

and we get ($n = 1$, $\nu = ½$, $X = b^2T^2$)

$$Q(0,b,T) = T[E_{1/2}(b^2T^2) - E_1(b^2T^2)] = T\left[\sqrt{\pi}\frac{\mathrm{erfc}(bT)}{bT} - E_1(b^2T^2)\right] = \frac{\sqrt{\pi}}{b}\mathrm{erfc}(bT) - TE_1(b^2T^2)$$

This can also be obtained from the case $a \le b$ and $a \to 0$.

Refinements for the Logarithmic Computations When the Arguments are Close to 1.

If we write the logarithms in the P and Q formulas

$$\ln\left[\frac{2\sqrt{a^2+b^2}}{b+\sqrt{a^2+b^2}}\right] \quad \text{and} \quad \ln\left[\frac{2\sqrt{a^2+b^2}}{a+\sqrt{a^2+b^2}}\right] \quad \text{in the form} \quad \ln\left[\frac{2\sqrt{1+r^2}}{1+\sqrt{1+r^2}}\right]$$

where r=a/b or r=b/a we can deduce that the computation will lose significant digits when r is small. To be sure we wind up with significant digits, we compute the logarithm in the form

$$\ln(1+x) = x\sum_{k=0}^{\infty}(-1)^k \frac{x^k}{k+1} \quad , \qquad |x| < 1$$

with x < 0.1. Then,

$$y = \ln\left[\frac{2\sqrt{1+r^2}}{1+\sqrt{1+r^2}}\right] = \ln 2 + \frac{1}{2}\ln(1+r^2) - \ln(1+\sqrt{1+r^2})$$

Notice that $\ln(1+r^2)$ is in the proper form with $x = r^2$. The last logarithm can be brought into the proper form with the manipulations

$$y = \ln 2 + \frac{1}{2}\ln(1+r^2) - \ln(1+\sqrt{1+r^2}) = \ln 2 + \frac{1}{2}\ln(1+r^2) - \ln\left[2 + (\sqrt{1+r^2} - 1)\right]$$

$$= \ln 2 + \frac{1}{2}\ln(1+r^2) - \ln\left[2 + \frac{r^2}{(1+\sqrt{1+r^2})}\right] = \frac{1}{2}\ln(1+r^2) - \ln\left[1 + \frac{r^2/2}{(1+\sqrt{1+r^2})}\right]$$

Thus with $x = (r^2/2)/(1+\sqrt{1+r^2})$ we have the proper form for the last logarithm and no losses of significance since

$$y \approx \frac{r^2}{2} - \frac{r^2/2}{(1+\sqrt{1+r^2})} \approx \frac{r^2}{2} - \frac{r^2}{4} = \frac{r^2}{4} \quad \text{for } r \to 0\,.$$

Similarly for the logarithms,

$$\ln\left[\frac{b+\sqrt{a^2+b^2}}{a}\right] \quad \text{and} \quad \ln\left[\frac{a+\sqrt{a^2+b^2}}{b}\right]$$

we have the form

$$y = \ln(r+\sqrt{1+r^2}) = \ln\left[1 + r + (\sqrt{1+r^2} - 1)\right] = \ln\left[1 + r + \frac{r^2}{1+\sqrt{1+r^2}}\right]$$

where
$$x = r + \frac{r^2}{1+\sqrt{1+r^2}}$$

in the logarithmic computation when r is small.

Asymptotic Expansions

We generate the large bT asymptotic expansion for $P^c(a,b,T)$ by replacing the co-error function by its asypmptotic expansion,

$$erfc(bw) \sim \frac{e^{-b^2w^2}}{bw\sqrt{\pi}} \sum_{k=0}^{\infty} (-1)^k \frac{(1/2)_k}{(bw)^{2k}}$$

and integrating

$$P^c(a,b,T) \sim \frac{1}{2bT\sqrt{\pi}} \sum_{k=0}^{\infty} (-1)^k \frac{(1/2)_k}{(bT)^{2k}} E_{k+3/2}(X), \qquad X = T^2(a^2 + b^2)\,.$$

Using erf(x)=1-erfc(x), $P(a,b,T)$ for large bT follows from the relation

$$P(a,b,T) = \frac{1}{2} E_1(a^2T^2) - P^c(a,b,T)\,,$$

$$P(a,b,T) \sim \frac{1}{2} E_1(a^2T^2) - \frac{1}{2bT\sqrt{\pi}} \sum_{k=0}^{\infty} (-1)^k \frac{(1/2)_k}{(bT)^{2k}} E_{k+3/2}(X), \qquad X = T^2(a^2 + b^2)$$

For $Q(a,b,T)$, we replace $E_1(b^2w^2)$ by its expansion,

$$E_1(b^2w^2) \sim \frac{e^{-b^2w^2}}{(bw)^2} \sum_{k=0}^{\infty} (-1)^k \frac{(k)!}{(bw)^{2k}},$$

to get the large bT expansion

$$Q(a,b,T) \sim \frac{T}{2(bT)^2} \sum_{k=0}^{\infty} (-1)^k \frac{(k)!}{(bT)^{2k}} E_{k+3/2}(X), \qquad X = T^2(a^2 + b^2)\,.$$

For the large aT expansions, we use

$$P(a,b,T) \;=\; \frac{1}{2} E_1(a^2T^2)\,\mathrm{erf}\,(bT) \;+\; \frac{b}{\sqrt{\pi}} Q(b,a,T)$$

to get

$$P(a,b,T) \sim \frac{1}{2} E_1(a^2T^2)\mathrm{erf}\,(bT) + \frac{b}{\sqrt{\pi}} \left[\frac{T}{2(aT)^2} \sum_{k=0}^{\infty} (-1)^k \frac{(k)!}{(aT)^{2k}} E_{k+3/2}(X) \right], \quad X = T^2(a^2 + b^2)\,.$$

Then, the large aT expansion for $P^c(a,b,T)$ is

$$P^c(a,b,T) \sim \frac{1}{2} E_1(a^2T^2) erfc(bT) - \frac{b}{\sqrt{\pi}} \left[\frac{T}{2(aT)^2} \sum_{k=0}^{\infty} (-1)^k \frac{(k)!}{(aT)^{2k}} E_{k+3/2}(X) \right], \; X = T^2(a^2 + b^2)\,.$$

For Q, we solve for $Q(b,a,T)$ above, exchange a and b to get

$$Q(a,b,T) = \frac{\sqrt{\pi}}{a} \left[\; P(b,a,T) - \frac{1}{2} E_1(b^2T^2)\,\mathrm{erf}(aT) \;\right].$$

then replace $P(b,a,T)$ with the expansion above for large second parameter(large aT)

$$Q(a,b,T) \sim \frac{\sqrt{\pi}}{a} \left[\frac{1}{2} E_1(b^2T^2)\ \mathrm{erfc(aT)} - \frac{1}{2aT\sqrt{\pi}} \sum_{k=0}^{\infty} (-1)^k \frac{(1/2)_k}{(aT)^{2k}} E_{k+3/2}(X) \right], \quad X = T^2(a^2 + b^2)\,.$$

In all of these expansions, the truncation error is bounded in magnitude by the next term.

Folder 12

Evaluation of $\mathbf{I} = \int \mathbf{x e^{(a^2-b^2)x} erfc}\left(\mathbf{a}\sqrt{\mathbf{x}} + \frac{\mathbf{c}}{\sqrt{\mathbf{x}}}\right)\mathbf{d}\,x$

Donald E. Amos, July 2001

From Beck et al., p. 426 #3, we have

$$J \equiv \int e^{(a^2-b^2)x} \operatorname{erfc}\left(a\sqrt{x} + \frac{c}{\sqrt{x}}\right)dx$$

$$= \frac{e^{(a^2-b^2)x}}{a^2-b^2}\operatorname{erfc}\left(a\sqrt{x} + \frac{c}{\sqrt{x}}\right) + \frac{e^{-2ac}}{2}\left[-f(\sqrt{x},a,b,c) + f(\sqrt{x},-a,b,-c)\right]$$

where $f(\sqrt{x},a,b,c) = \dfrac{e^{2bc}}{(a^2-b^2)}\left(1+\dfrac{a}{b}\right)\operatorname{erfc}\left(b\sqrt{x} + \dfrac{c}{\sqrt{x}}\right)$

The procedure is to let $b=\sqrt{y}$, differentiate wrt y and then replace y by b^2. This gets the x multiplier under the integral sign.

Let $b=\sqrt{y}$, $\quad X(\sqrt{x},a,c) = a\sqrt{x} + c/\sqrt{x}$

$$J(y) = e^{a^2x}\left[\frac{e^{-xy}}{a^2-y}\right]\operatorname{erfc}[X(\sqrt{\mathrm{x}},a,c)] + \frac{e^{-2ac}}{2}[-f(\sqrt{x},a,\sqrt{y},c) + f(\sqrt{x},-a,\sqrt{y},-c)]$$

and

$$f(\sqrt{x},a,\sqrt{y},c) = \left[\frac{e^{2c\sqrt{y}}}{a^2-y}\left(1+\frac{a}{\sqrt{y}}\right)\right]\operatorname{erfc}[X(\sqrt{x},\sqrt{y},c)]$$

$$\frac{d}{dy}\left[\frac{e^{-xy}}{a^2-y}\right] = \frac{-xe^{-xy}(a^2-y)+e^{-xy}}{(a^2-y)^2} = \left[-x+\frac{1}{a^2-y}\right]\frac{e^{-xy}}{(a^2-y)}$$

$$\frac{d}{dy}\left[\frac{e^{2c\sqrt{y}}}{a^2-y}\left(1+\frac{a}{\sqrt{y}}\right)\right] = \frac{2ce^{2c\sqrt{y}}}{2\sqrt{y}}\cdot\frac{1}{a^2-y}\left(1+\frac{a}{\sqrt{y}}\right)$$
$$+e^{2c\sqrt{y}}\left[\frac{-a}{2y^{3/2}(a^2-y)} + \left(1+\frac{a}{\sqrt{y}}\right)\frac{1}{(a^2-y)^2}\right]$$

$$\frac{d}{dy}\operatorname{erfc}[X(\sqrt{x},\sqrt{y},c)] = -\sqrt{\frac{x}{\pi y}}e^{-X^2(\sqrt{x},\sqrt{y},c)}$$

$I = -\left.\frac{dJ}{dy}\right|_{y=b^2}$. Now

$$\frac{dJ}{dy} = \frac{e^{(a^2-y)x}}{a^2-y}\left[-x+\frac{1}{a^2-y}\right]\operatorname{erfc}[X(\sqrt{x},a,c)]$$
$$+\frac{e^{-2ac}}{2}\left[-\frac{df}{dy}(\sqrt{x},a,\sqrt{y},c)+\frac{df}{dy}(\sqrt{x},-a,\sqrt{y},-b)\right]$$

$$\frac{df}{dy}(\sqrt{x},a,\sqrt{y},c) = \left\{\frac{-e^{2c\sqrt{y}}}{a^2-y}\left(1+\frac{a}{\sqrt{y}}\right)\cdot\sqrt{\frac{x}{\pi y}}e^{-X^2(\sqrt{x},\sqrt{y},c)}\right.$$
$$\left.+\left(\frac{ce^{2c\sqrt{y}}}{\sqrt{y}(a^2-y)}\left(1+\frac{a}{\sqrt{y}}\right)+\frac{e^{2c\sqrt{y}}}{a^2-y}\left[\frac{-a}{2y^{3/2}}+\left(1+\frac{a}{\sqrt{y}}\right)\cdot\frac{1}{a^2-y}\right]\right)\operatorname{erfc}[X(\sqrt{x},\sqrt{y},c)]\right\}$$

$$\frac{df}{dy}(\sqrt{x},a,\sqrt{y},c) = \frac{e^{2c\sqrt{y}}}{a^2-y^2}\left\{-\sqrt{\frac{x}{\pi y}}e^{-X^2(\sqrt{x},\sqrt{y},c)}\left(1+\frac{a}{\sqrt{y}}\right)\right.$$
$$\left.+\left(\left[\frac{c}{\sqrt{y}}+\frac{1}{a^2-y}\right]\left(1+\frac{a}{\sqrt{y}}\right)-\frac{a}{2y^{3/2}}\right)\operatorname{erfc}[X(\sqrt{x},\sqrt{y},c)]\right\}$$

$$X(\sqrt{x},\sqrt{y},c) = \sqrt{y}\sqrt{x}+c/\sqrt{x}$$

$$I = -\left.\frac{dJ}{dy}\right|_{y=b^2} = -\frac{e^{(a^2-b^2)x}}{a^2-b^2}\left[-x+\frac{1}{a^2-b^2}\right]\operatorname{erfc}[X(\sqrt{x},a,c)]$$
$$+\frac{e^{-2ac}}{2}[F(\sqrt{x},a,b,c)-F(\sqrt{x},-a,b,-c)]$$

$$F(\sqrt{x},a,b,c) = \left.\frac{df}{dy}\right|_{y=b^2} = \frac{e^{2bc}}{a^2-b^2}\left\{\frac{-\sqrt{x}}{b\sqrt{\pi}}e^{-X^2(\sqrt{x},b,c)}\left(1+\frac{a}{b}\right)\right.$$
$$\left.+\left[\left(\frac{c}{b}+\frac{1}{a^2-b^2}\right)\left(1+\frac{a}{b}\right)-\frac{a}{2b^3}\right]\operatorname{erfc}[X(\sqrt{x},b,c]\right\}$$

$$X(\sqrt{x},a,c) = a\sqrt{x}+c/\sqrt{x}$$

Folder 13

Evaluation of

$$\mathbf{I_{13}(a,b,T)=\int_T^\infty \frac{e^{-a^2w^2}erf(bw)}{w^3}dw=\frac{1}{2}\int_0^t\ e^{-a^2/u}erf(b/\sqrt{u})du}$$

$$\mathbf{I^c_{13}(a,b,T)=\int_T^\infty \frac{e^{-a^2w^2}erfc(bw)}{w^3}dw=\frac{1}{2}\int_0^t\ e^{-a^2/u}erfc(b/\sqrt{u})du}$$

$$\mathbf{a>0,\quad b>0,\quad \mathit{T}=\frac{1}{\sqrt{\mathit{t}}},\quad t>0}$$

Donald E. Amos, December 2001

Summary

These integrals are generalizations of those found in Folders 10 and 11. The computational forms are:

$$I_{13}(a,b,T)=\frac{e^{-a^2T^2}}{2T^2}erf(bT)+\frac{b}{T}ierfc(T\sqrt{a^2+b^2})\text{-}a^2P(a,b,T)$$

$$=\frac{erf(bT)}{2T^2}E_2(a^2T^2)+\frac{b}{T}ierfc[T(\sqrt{a^2+b^2}]-\frac{a^2b}{\sqrt{\pi}}Q(b,a,T)$$

$$I^c_{13}(a,b,T)=\frac{e^{-a^2T^2}}{2T^2}erfc(bT)\text{-}\frac{a^2}{2}E_1(a^2T^2)\text{-}\frac{b}{T}ierfc[T\sqrt{a^2+b^2}]+a^2P(a,b,T)$$

$$=\frac{1}{2T^2}E_2(a^2T^2)erfc(bT)\text{-}\frac{b}{T}ierfc[T\sqrt{a^2+b^2}]+\frac{a^2b}{\sqrt{\pi}}Q(b,a,T)$$

$$I_{13}(a,b,T)+I^c_{13}(a,b,T)=\frac{1}{2T^2}E_2(a^2T^2)$$

where P(a,b,T) and Q(b,a,T) are computed in Folder 11. Folder 10 has a discussion of the computation of *i*erfc(*x*) which is implemented in subroutine DIERFC.

Derivation of I_{13} Formulas

Integrate by parts:

$$u=\text{erf}(bw) \qquad\qquad dv=\frac{e^{-a^2w^2}}{w^3}dw$$

$$du=\frac{2b}{\sqrt{\pi}}e^{-b^2w^2}dw \qquad\qquad v=-\int_w^\infty \frac{e^{-a^2x^2}}{x^3}dx=-\frac{1}{2w^2}E_2(a^2w^2)$$

Then with $E_2(x) = e^{-x} - xE_1(x)$, we have

$$I_{13}(a,b,T) = \frac{\mathrm{erf}(bT)}{2T^2} E_2(a^2T^2) + \frac{b}{\sqrt{\pi}} \int_T^\infty \frac{e^{-(a^2+b^2)w^2}}{w^2} dw - \frac{a^2 b}{\sqrt{\pi}} \int_T^\infty e^{-b^2w^2} E_1(a^2w^2) dw$$

$$= \frac{\mathrm{erf}(bT)}{2T^2} E_2(a^2T^2) + \frac{b}{2T\sqrt{\pi}} E_{3/2}[T^2(a^2+b^2)] - \frac{a^2 b}{\sqrt{\pi}} Q(b,a,T)$$

where

$$Q(a,b,T) = \int_T^\infty e^{-a^2w^2} E_1(b^2w^2) dw, \quad E_{3/2}(x) = 2\sqrt{\pi}\, i\mathrm{erfc}(\sqrt{x}).$$

Now, Q is computed in Folder 11 and is related to P by the equations

$$P(a,b,T) = \int_T^\infty \frac{e^{-a^2w^2}\,\mathrm{erf}(bw)}{w} dw$$

$$P(a,b,T) = \frac{1}{2} E_1(a^2T^2)\,\mathrm{erf}(bT) + \frac{b}{\sqrt{\pi}} Q(b,a,T).$$

Therefore, I_{13} can also be represented in terms of P by

$$I_{13}(a,b,T) = \frac{e^{-a^2T^2}}{2T^2} \mathrm{erf}(bT) + \frac{b}{T} i\mathrm{erfc}(T\sqrt{a^2+b^2}) - a^2 P(a,b,T)$$

where both P and Q are computed in Folder 11 and $i\mathrm{erfc}(x)$ is described in Folder 10 and implemented in subroutine DIERFC. For $\mathrm{I}_{13}^{c}(\mathrm{a},\mathrm{b},\mathrm{T})$ we use erfc(x)=1-erf(x) to get

$$I_{13}^c(a,b,T) = \frac{1}{2T^2} E_2(a^2T^2) - I_{13}(a,b,T)$$

and from above,

$$I_{13}^c(a,b,T) = \frac{1}{2T^2} E_2(a^2T^2) erfc(bT) - \frac{b}{2T\sqrt{\pi}} E_{3/2}[T^2(a^2+b^2)] + \frac{a^2 b}{\sqrt{\pi}} Q(b,a,T)$$

$$= \frac{1}{2T^2} E_2(a^2T^2) erfc(bT) - \frac{b}{T} ierfc[T\sqrt{a^2+b^2}] + \frac{a^2 b}{\sqrt{\pi}} Q(b,a,T)$$

$$I_{13}^c(a,b,T) = \frac{1}{2T^2} E_2(a^2T^2) - \frac{e^{-a^2T^2}}{2T^2} erf(bT) - \frac{b}{T} ierfc[T\sqrt{a^2+b^2}] + a^2 P(a,b,T)$$

$$= \frac{e^{-a^2T^2}}{2T^2} erfc(bT) - \frac{a^2}{2} E_1(a^2T^2) - \frac{b}{T} ierfc[T\sqrt{a^2+b^2}] + a^2 P(a,b,T)$$

Folder 14

Evaluation of

$$\mathbf{I_{14}(a,b,c,T) = \int_T^\infty e^{-c^2w^2} \frac{erf(aw)erf(bw)}{w^2} dw = \frac{1}{2}\int_0^t e^{-c^2/u} \frac{erf(a/\sqrt{u})erf(b/\sqrt{u})}{\sqrt{u}} du}$$

$$\mathbf{I_{14}^c(a,b,c,T) = \int_T^\infty e^{-c^2w^2} \frac{erfc(aw)erfc(bw)}{w^2} dw = \frac{1}{2}\int_0^t e^{-c^2/u} \frac{erfc(a/\sqrt{u})erfc(b/\sqrt{u})}{\sqrt{u}} du}$$

$$\mathbf{a>0, \quad b>0, \quad \boldsymbol{T} = \frac{1}{\sqrt{\boldsymbol{t}}}, \quad t>0}$$

Donald E. Amos, December 2001

Derivation of Formula:

Integrate I_{14} by parts

$$u = \frac{e^{-c^2w^2}}{w^2} \qquad\qquad dv = \mathrm{erf}(aw)\mathrm{erf}(bw)dw$$

$$du = \left[\frac{-2c^2e^{-c^2w^2}}{w} - \frac{2e^{-c^2w^2}}{w^3}\right]dw \qquad v(w) = I_2(a,b,w) = \int_0^w erf(ax)erf(bx)dx$$

Here $I_2(a,b,w)$ is evaluated in Folder 9

$$I_2(a,b,w) = w\,\mathrm{erf}(aw)\,\mathrm{erf}(bw) + \frac{e^{-a^2w^2}}{a\sqrt{\pi}}\mathrm{erf}(bw) + \frac{e^{-b^2w^2}}{b\sqrt{\pi}}\mathrm{erf}(aw) - \frac{\sqrt{a^2+b^2}}{ab\sqrt{\pi}}\mathrm{erf}(w\sqrt{a^2+b^2}).$$

Now,

$$I_{14}(a,b,c,T) = \frac{-e^{-c^2T^2}I_2(a,b,T)}{T^2} + 2c^2\int_T^\infty \frac{e^{-c^2w^2}I_2(a,b,w)}{w}dw + 2\int_T^\infty \frac{e^{-c^2w^2}I_2(a,b,w)}{w^3}dw.$$

We now evaluate each of these integrals:

$$\int_T^\infty \frac{e^{-c^2w^2}I_2(a,b,w)}{w}dw = \int_T^\infty e^{-c^2w^2}\,\mathrm{erf}(aw)\,\mathrm{erf}(bw)dw + \frac{1}{a\sqrt{\pi}}\int_T^\infty \frac{e^{-(a^2+c^2)w^2}\,\mathrm{erf}(bw)}{w}dw$$

$$+\frac{1}{b\sqrt{\pi}}\int_T^\infty \frac{e^{-(b^2+c^2)w^2}\,\mathrm{erf}(aw)dw}{w} - \frac{\sqrt{a^2+b^2}}{ab\sqrt{\pi}}\int_T^\infty \frac{e^{-c^2w^2}\,\mathrm{erf}(w\sqrt{a^2+b^2})}{w}dw$$

$$\int_T^\infty \frac{e^{-c^2w^2} I_2(a,b,w)}{w} dw = I_3(a,b,c,T) + \frac{1}{a\sqrt{\pi}} P(\sqrt{a^2+c^2},b,T) + \frac{1}{b\sqrt{\pi}} P(\sqrt{b^2+c^2},a,T)$$

$$-\frac{\sqrt{a^2+b^2}}{ab\sqrt{\pi}} P(c,\sqrt{a^2+b^2},T)$$

where $I_3(a,b,c,T)$ is treated in Folder 7 and the P function is treated in Folder 11.

To proceed, we evaluate

$$\int_T^\infty e^{-c^2w^2} \frac{I_2(a,b,w)}{w^3} dw = \int_T^\infty e^{-c^2w^2} \frac{\mathrm{erf}(aw)\,\mathrm{erf}(bw)}{w^2} dw + \frac{1}{a\sqrt{\pi}} \int_T^\infty e^{-(a^2+c^2)w^2} \frac{\mathrm{erf}(bw)}{w^3} dw$$

$$+ \frac{1}{b\sqrt{\pi}} \int_T^\infty e^{-(b^2+c^2)w^2} \frac{\mathrm{erf}(aw)}{w^3} dw - \frac{\sqrt{a^2+b^2}}{ab\sqrt{\pi}} \int_T^\infty e^{-c^2w^2} \frac{\mathrm{erf}(w\sqrt{a^2+b^2})}{w^3} dw$$

$$= I_{14}(a,b,c,T) + \frac{1}{a\sqrt{\pi}} I_{13}(\sqrt{a^2+c^2},b,T) + \frac{1}{b\sqrt{\pi}} I_{13}(\sqrt{b^2+c^2},a,T) - \frac{\sqrt{a^2+b^2}}{ab\sqrt{\pi}} I_{13}(c,\sqrt{a^2+b^2},T)$$

Notice that $I_{14}(a,b,c,T)$, the unknown, appears on the right and also on the right side of the basic equation. Here, $I_{13}(a,b,c,T)$ is the function of Folder 13 in Chapter 3. We solve for $I_{14}(a,b,c,T)$ and obtain:

$$I_{14}(a,b,c,T) = \frac{I_2(a,b,T)\; e^{-c^2T^2}}{T^2} +$$

$$-2c^2\left[I_3(a,b,c,T) + \frac{1}{a\sqrt{\pi}} P(\sqrt{a^2+c^2},b,T) + \frac{1}{b\sqrt{\pi}} P(\sqrt{b^2+c^2},a,T) - \frac{\sqrt{a^2+b^2}}{ab\sqrt{\pi}} P(c,\sqrt{a^2+b^2},T)\right]$$

$$-2\left[\frac{1}{a\sqrt{\pi}} I_{13}(\sqrt{a^2+c^2},b,T) + \frac{1}{b\sqrt{\pi}} I_{13}(\sqrt{b^2+c^2},a,T) - \frac{\sqrt{a^2+b^2}}{ab\sqrt{\pi}} I_{13}(c,\sqrt{a^2+b^2},T)\right]$$

where

$$I_2(a,b,T) = T\,\mathrm{erf}(aT)\,\mathrm{erf}(bT) + \frac{e^{-a^2T^2}}{a\sqrt{\pi}} \mathrm{erf}(bT) + \frac{e^{-b^2T^2}}{b\sqrt{\pi}} \mathrm{erf}(aT) - \frac{\sqrt{a^2+b^2}}{ab\sqrt{\pi}} \mathrm{erf}(T\sqrt{a^2+b^2}),$$

For I_{14}^c, we can use erfc(x)=1-erf(x) and get

$$I_{14}^c(a,b,c,T) = \int_T^\infty \frac{e^{-c^2w^2}}{w^2} dw - \int_T^\infty \frac{e^{-c^2w^2}}{w^2} erf(aw) dw - \int_T^\infty \frac{e^{-c^2w^2}}{w^2} erf(bw) dw + I_{14}(a,b,c,T)$$

and with $E_{3/2}(x^2) = 2\sqrt{\pi}\, ierfc(x)$ we have a representation

$$I_{14}^{c}(a,b,c,T)=\frac{\sqrt{\pi}}{T}ierfc(cT)-I_1(c,a,T)-I_1(c,b,T)+I_{14}(a,b,c,T).$$

But integration by parts gives a more computationally appealing form similar to that for $I_{14}(a,b,c,T)$. Then,

$$I_{14}^{c}(a,b,c,T)=\int_T^\infty e^{-c^2w^2}\,\frac{erfc(aw)erfc(bw)}{w^2}dw$$

and

$$u=\frac{e^{-c^2w^2}}{w^2} \qquad\qquad dv=\text{erfc}(aw)\text{erfc}(bw)dw$$

$$du=\left[\frac{-2c^2e^{-c^2w^2}}{w}-\frac{2e^{-c^2w^2}}{w^3}\right]dw \qquad\qquad v(w)=-I_2^c(a,b,w)$$

where $I_2^c(a,b,w)$ is computed in Folder 9, Chapter3. Then

$$I_{14}^{c}(a,b,c,T)=\frac{e^{-c^2T^2}I_2^c(a,b,T)}{T^2}-2c^2\int_T^\infty \frac{e^{-c^2w^2}I_2^c(a,b,w)}{w}dw-2\int_T^\infty \frac{e^{-c^2w^2}I_2^c(a,b,w)}{w^3}dw$$

and from Folder 9,

$$I_2^c(a,b,w)=-\,werfc(aw)erfc(bw)+\frac{e^{-a^2w^2}}{a\sqrt{\pi}}erfc(bw)+\frac{e^{-b^2w^2}}{b\sqrt{\pi}}erfc(aw)-\frac{\sqrt{a^2+b^2}}{ab\sqrt{\pi}}erfc(w\sqrt{a^2+b^2})$$

Consequently,

$$I_{14}^{c}(a,b,c,T)=\frac{e^{-c^2T^2}I_2^c(a,b,T)}{T^2}-2c^2\int_T^\infty \frac{e^{-c^2w^2}I_2^c(a,b,w)}{w}dw-2\int_T^\infty \frac{e^{-c^2w^2}I_2^c(a,b,w)}{w^3}dw.$$

We need to evaluate these last two integrals.

$$\int_T^\infty \frac{e^{-c^2w^2}I_2^c(a,b,w)}{w}dw=-\int_T^\infty e^{-c^2w^2}erfc(aw)erfc(bw)dw+\frac{1}{a\sqrt{\pi}}\int_T^\infty \frac{e^{-(a^2+c^2)w^2}erfc(bw)}{w}dw$$

$$+\frac{1}{b\sqrt{\pi}}\int_T^\infty \frac{e^{-(b^2+c^2)w^2}erfc(aw)dw}{w}-\frac{\sqrt{a^2+b^2}}{ab\sqrt{\pi}}\int_T^\infty \frac{e^{-c^2w^2}erfc(w\sqrt{a^2+b^2})}{w}dw$$

$$\int_T^\infty \frac{e^{-c^2w^2}I_2^c(a,b,w)}{w}dw=\;-I_3^c(a,b,c,T)+\frac{1}{a\sqrt{\pi}}P^c(\sqrt{a^2+c^2},b,T)+\frac{1}{b\sqrt{\pi}}P^c(\sqrt{b^2+c^2},a,T)$$

$$-\frac{\sqrt{a^2+b^2}}{ab\sqrt{\pi}}P^c(c,\sqrt{a^2+b^2},T)$$

where $I_3^c(a,b,c,T)$ is treated in Folder7.

$$\int_T^\infty e^{-c^2w^2}\,\frac{I_2^c(a,b,w)}{w^3}\,dw = -\int_T^\infty e^{-c^2w^2}\,\frac{erfc(aw)erfc(bw)}{w^2}\,dw + \frac{1}{a\sqrt{\pi}}\int_T^\infty e^{-(a^2+c^2)w^2}\,\frac{erfc(bw)}{w^3}\,dw$$

$$+\frac{1}{b\sqrt{\pi}}\int_T^\infty e^{-(b^2+c^2)w^2}\,\frac{erfc(aw)}{w^3}\,dw - \frac{\sqrt{a^2+b^2}}{ab\sqrt{\pi}}\int_T^\infty e^{-c^2w^2}\,\frac{erfc(w\sqrt{a^2+b^2})}{w^3}\,dw$$

$$= -I_{14}^c(a,b,c,T) + \frac{1}{a\sqrt{\pi}}I_{13}^c(\sqrt{a^2+c^2},b,T) + \frac{1}{b\sqrt{\pi}}I_{13}^c(\sqrt{b^2+c^2},a,T) - \frac{\sqrt{a^2+b^2}}{ab\sqrt{\pi}}I_{13}^c(c,\sqrt{a^2+b^2},T)$$

As before, we solve for $I_{14}^c(a,b,c,T)$ to obtain

$$I_{14}^c(a,b,c,T) = -\frac{I_2^c(a,b,T)\;e^{-c^2T^2}}{T^2} +$$

$$+2c^2\left[-\mathrm{I}_3^c(\mathrm{a,b,c,T}) + \frac{1}{\mathrm{a}\sqrt{\pi}}\mathrm{P}^c(\sqrt{\mathrm{a}^2+\mathrm{c}^2},\mathrm{b,T}) + \frac{1}{\mathrm{b}\sqrt{\pi}}\mathrm{P}^c(\sqrt{\mathrm{b}^2+\mathrm{c}^2},\mathrm{a,T}) - \frac{\sqrt{\mathrm{a}^2+\mathrm{b}^2}}{\mathrm{ab}\sqrt{\pi}}\mathrm{P}^c(\mathrm{c},\sqrt{\mathrm{a}^2+\mathrm{b}^2},\mathrm{T})\right]$$

$$+2\left[\frac{1}{\mathrm{a}\sqrt{\pi}}\mathrm{I}_{13}^c(\sqrt{\mathrm{a}^2+\mathrm{c}^2},\mathrm{b,T}) + \frac{1}{\mathrm{b}\sqrt{\pi}}\mathrm{I}_{13}^c(\sqrt{\mathrm{b}^2+\mathrm{c}^2},\mathrm{a,T}) - \frac{\sqrt{\mathrm{a}^2+\mathrm{b}^2}}{\mathrm{ab}\sqrt{\pi}}\mathrm{I}_{13}^c(\mathrm{c},\sqrt{\mathrm{a}^2+\mathrm{b}^2},\mathrm{T})\right]$$

where, from Folder 11,

$$P^c(a,b,T) = \frac{1}{2}E_1(a^2T^2)erfc(bT) - \frac{b}{\sqrt{\pi}}Q(b,a,T)\,.$$

Folder 15

Folders 3 and 6 Revisited

$$\boldsymbol{I_6(a,b,T) = \int_T^\infty \frac{\mathrm{erf}(aw)\mathrm{erf}(bw)}{w^2}dw = \frac{1}{2}\int_0^t \frac{\mathrm{erf}(a/\sqrt{u})\mathrm{erf}(b/\sqrt{u})}{\sqrt{u}}du}$$

$$\mathbf{I_6^c(a,b,T) = \int_T^\infty \frac{erfc(aw)erfc(bw)}{w^2}dw = \frac{1}{2}\int_0^t \frac{erfc(a/\sqrt{u})erfc(b/\sqrt{u})}{\sqrt{u}}du}$$

$$\boldsymbol{T = \frac{1}{\sqrt{t}}}$$

Donald E. Amos, December 2001

Summary

The formulae for the computation of P and Q in Folder 11 were obtained from the derivations in Folder 6. In this folder, we restate the results of Folder 6 for I_6 in terms of P and Q. Folder 3 presents the quadrature procedure for $I_6(a,b,T)$. The results are

$$\mathrm{I_6(a,b,T) = \frac{erf(aT)erf(bT)}{T} + \frac{2a}{\sqrt{\pi}}P(a,b,T) + \frac{2b}{\sqrt{\pi}}P(b,a,T)}$$

$$I_6(a,b,T) = \mathrm{erf}(aT)\left[\frac{\mathrm{erf}(bT)}{T} + \frac{b}{\sqrt{\pi}}E_1(b^2T^2)\right] + \frac{2a}{\sqrt{\pi}}P(a,b,T) + \frac{2ab}{\pi}Q(a,b,T)$$

$$\mathrm{I_6(a,b,T) = \frac{erf(aT)erf(bT)}{T} + \frac{1}{\sqrt{\pi}}\left[aE_1(a^2T^2)erf(bT) + bE_1(b^2T^2)erf(aT)\right] + \frac{2ab}{\pi}[Q(a,b,T)+Q(b,a,T)]}$$

and

$$\mathrm{I_6^c(a,b,T) = \frac{erfc(aT)erfc(bT)}{T} - \frac{2a}{\sqrt{\pi}}P^c(a,b,T) - \frac{2b}{\sqrt{\pi}}P^c(b,a,T)}$$

$$I_6^c(a,b,T) = \mathrm{erfc(aT)}\left[\frac{\mathrm{erfc(bT)}}{\mathrm{T}} - \frac{\mathrm{b}}{\sqrt{\pi}}\mathrm{E_1(b^2T^2)}\right] - \frac{\mathrm{2a}}{\sqrt{\pi}}\mathrm{P^c(a,b,T)} + \frac{\mathrm{2ab}}{\pi}\mathrm{Q(a,b,T)}$$

$$\mathrm{I_6^c(a,b,T) = \frac{erfc(aT)erfc(bT)}{T} - \frac{1}{\sqrt{\pi}}\left[aE_1(a^2T^2)erfc(bT) + bE_1(b^2T^2)erfc(aT)\right] + \frac{2ab}{\pi}[Q(a,b,T)+Q(b,a,T)]}.$$

Derivation

Integrate $I_6(a,b,T)$ by parts:

$$u = \mathrm{erf}(aw)\mathrm{erf}(bw) \qquad dv = \frac{dw}{w^2}$$

$$du = \left[\frac{2a}{\sqrt{\pi}}e^{-a^2w^2}\,\mathrm{erf}(bw) + \frac{2b}{\sqrt{\pi}}e^{-b^2w^2}\,\mathrm{erf}(aw)\right]dw \qquad v = -\frac{1}{w}$$

Then

$$I_6(a,b,T) = \frac{\mathrm{erf}(aT)\,\mathrm{erf}(bT)}{T} + \frac{2a}{\sqrt{\pi}}P(a,b,T) + \frac{2b}{\sqrt{\pi}}P(b,a,T)$$

where

$$P(a,b,T) = \int_T^\infty e^{-a^2w^2}\frac{\mathrm{erf}(bw)}{w}dw \quad \text{and} \quad Q(a,b,T) = \int_T^\infty e^{-a^2w^2}E_1(b^2w^2)dw$$

are computed in Folder 11.

In Folder 6 we derived the expression

$$I_6(a,b,T) = \mathrm{erf}(aT)\left[\frac{\mathrm{erf}(bT)}{T} + \frac{b}{\sqrt{\pi}}E_1(b^2T^2)\right] + \frac{2a}{\sqrt{\pi}}P(a,b,T) + \frac{2ab}{\pi}Q(a,b,T)$$

with a similar expression for a and b reversed since I_6 is symmetric in a and b. From Folder 11 we have the two results:

$$P(a,b,T) = \frac{1}{2}E_1(a^2T^2)\,\mathrm{erf}(bT) + \frac{b}{\sqrt{\pi}}Q(b,a,T),\ \mathrm{P}^c(\mathrm{a,b,T}) = \frac{1}{2}\mathrm{E}_1(\mathrm{a}^2\mathrm{T}^2)\mathrm{erfc(bT)} - \frac{\mathrm{b}}{\sqrt{\pi}}\mathrm{Q(b,a,T)}.$$

If we eliminate Q between these last two expressions, we get the formula for I_6 in terms of $P(a,b,T)$ and $P(b,a,T)$ which was derived from the integration by parts.

We can also write the results in terms of Q:

$$\mathrm{I}_6(\mathrm{a,b,T}) = \frac{\mathrm{erf(aT)erf(bT)}}{\mathrm{T}} + \frac{1}{\sqrt{\pi}}\left[\mathrm{aE}_1(\mathrm{a}^2\mathrm{T}^2)\mathrm{erf(bT)}+\mathrm{bE}_1(\mathrm{b}^2\mathrm{T}^2)\mathrm{erf(aT)}\right] + \frac{\mathrm{2ab}}{\pi}[\mathrm{Q(a,b,T)+Q(b,a,T)}]$$

With $erfc(x) = 1 - erf(x)$ we also have

$$I_6^c(a,b,T = \frac{1}{T} - J_2(a,T) - J_2(b,T) + I_6(a,b,T)$$

where

$$J_2(a,T) = \int_T^\infty \frac{erf(aw)}{w^2}dw = \frac{erf(aT)}{T} + \frac{a}{\sqrt{\pi}}E_1(a^2T^2).$$

Then

$$\mathrm{I}_6^c(\mathrm{a,b,T}) = \frac{\mathrm{erfc(aT)erfc(bT)}}{\mathrm{T}} - \frac{1}{\sqrt{\pi}}\left[\mathrm{aE}_1(\mathrm{a}^2\mathrm{T}^2)\mathrm{erfc(bT)}+\mathrm{bE}_1(\mathrm{b}^2\mathrm{T}^2)\mathrm{erfc(aT)}\right] + \frac{\mathrm{2ab}}{\pi}[\mathrm{Q(a,b,T)+Q(b,a,T)}]$$

and, in terms of P^c of Folder 11,

$$\mathrm{I}_6^c(\mathrm{a,b,T}) = \frac{\mathrm{erfc(aT)erfc(bT)}}{\mathrm{T}} - \frac{\mathrm{2a}}{\sqrt{\pi}}\mathrm{P}^c(\mathrm{a,b,T}) - \frac{\mathrm{2b}}{\sqrt{\pi}}\mathrm{P}^c(\mathrm{b,a,T}).$$

If we eliminate Q(b,a,T) from the expression above, we get

$$I_6^c(a,b,T) = \mathrm{erfc(aT)}\left[\frac{\mathrm{erfc(bT)}}{\mathrm{T}} - \frac{\mathrm{b}}{\sqrt{\pi}}\mathrm{E}_1(\mathrm{b}^2\mathrm{T}^2)\right] - \frac{\mathrm{2a}}{\sqrt{\pi}}\mathrm{P}^c(\mathrm{a,b,T}) + \frac{\mathrm{2ab}}{\pi}\mathrm{Q(a,b,T)}$$

which corresponds to a similar expression for $I_6(a,b,T)$ which is shown above and derived in Folder 6.

Folder 16

Accurate Evaluation of

$$\boldsymbol{F(X)=\int_0^X \frac{\mathrm{erf}(x)}{x}dx,\quad G(X)=\int_X^\infty \frac{\mathrm{erfc}(x)}{x}dx,\quad \int_0^X e^{-a^2x^2}\ln x dx,\quad \int_X^\infty e^{-a^2x^2}\ln x dx}$$

$$\boldsymbol{a>0,\quad X>0}$$

Donald E. Amos, January 2002

Summary

The Formula

$$F(X)=\frac{\gamma}{2}+\ln 2X+G(X) \qquad\qquad (\gamma = \text{Euler Constant})$$

is derived and the asymptotics for these integrals are developed for $X\to 0$ and $X\to\infty$. Notice also that

$$F(aX)=\int_0^X \frac{\mathrm{erf}(ax)}{x}dx\,, \qquad\qquad G(aX)=\int_X^\infty \frac{\mathrm{erfc}(ax)}{x}dx\,.$$

Integration of $F(aX)$ and $G(aX)$ by parts gives [Beck, et al, p. 422, Table F.2 #5,6]

$$\int_0^X e^{-a^2x^2}\ln x dx=\frac{\sqrt{\pi}}{2a}[\mathrm{erf}(aX)\ln X-F(aX)]$$

$$\int_X^\infty e^{-a^2x^2}\ln x\, dx=\frac{\sqrt{\pi}}{2a}[\mathrm{erfc}(aX)\ln X+G(aX)]$$

which sum to

$$\int_0^\infty e^{-a^2x^2}\ln x dx=-\frac{\sqrt{\pi}}{2a}\left[\frac{\gamma}{2}+\ln(2a)\right]$$

Manipulation of these expressions further gives the relation

$$\int_0^\infty e^{-pt}\frac{\ln t}{\sqrt{t}}dt=-\frac{\sqrt{\pi}}{\sqrt{p}}[\gamma+\ln(4p)], \qquad\qquad \left[\frac{\partial E_\nu(a)}{\partial\nu}\right]_{\nu=1/2}=-2\sqrt{\frac{\pi}{a}}G(\sqrt{a})$$

A computational algorithm is given for each of the integrals $F(X)$ and $G(X)$. While $G(X)$ is computed in Folder 6, the worst errors are $O(10^{-8})$; on the other hand, the worst errors from the algorithms presented here are $O(10^{-13})$. The series in the formula (Folder 6)

$$G(X)=\frac{1}{2}E_1(X^2)-\frac{1}{2\pi}\sum_{k=0}^{\infty}\frac{C_k}{(k+1/2)}E_{k+3/2}(X^2), \qquad C_k=\frac{(1/2)_k}{k!}$$

is manipulated to increase the rate of convergence.

<u>Derivations</u> In Folder 11, we derived the formula

$$P(a,b,0)=\int_0^{\infty} e^{-a^2w^2}\frac{\operatorname{erf}(bw)}{w}dw=\ln\left[\frac{b+\sqrt{a^2+b^2}}{a}\right].$$

Consequently

$$\int_0^{T} e^{-a^2w^2}\frac{\operatorname{erf}(bw)}{w}dw=\int_0^{\infty} e^{-a^2w^2}\frac{\operatorname{erf}(bw)}{w}dw-\int_T^{\infty} e^{-a^2w^2}\frac{\operatorname{erf}(bw)}{w}dw$$

$$=\ln\left[\frac{b+\sqrt{a^2+b^2}}{a}\right]-P(a,b,T)$$

where formulas for $P(a,b,T)$ were developed in Folder 11 along with the asymptotics for $a\to 0$,

$$P(a,b,T)=-\frac{\gamma}{2}-\ln(aT)-\int_{bT}^{\infty}\frac{\operatorname{erfc}(x)}{x}dx+O(a^2)$$

Consequently, for $a\to 0$ we have

$$F(bT)=\frac{\gamma}{2}+\lim_{a\to 0}\left\{\ln(aT)+\ln\left[\frac{b}{a}\left(1+\sqrt{1+\frac{a^2}{b^2}}\right)\right]\right\}+G(bT)$$

$$=\frac{\gamma}{2}+\ln(2bT)+G(bT)$$

or

$$F(X)=\frac{\gamma}{2}+\ln(2X)+G(X), \qquad X>0.$$

We also note, by integration by parts, the relations

$$F(aT)=\int_0^{T}\frac{\operatorname{erf}(ax)}{x}dx=\operatorname{erf}(aT)\ln T-\frac{2a}{\sqrt{\pi}}\int_0^{T} e^{-a^2w^2}\ln w\,dw$$

or

$$\int_0^T e^{-a^2w^2} \ln w dw = \frac{\sqrt{\pi}}{2a}[\operatorname{erf}(aT)\ln T - F(aT)]$$

and similarly for $G(aT)$,

$$\int_T^\infty e^{-a^2w^2} \ln w dw = \frac{\sqrt{\pi}}{2a}[\operatorname{erfc}(aT)\ln T + G(aT)].$$

Adding these two relations together gives

$$\int_0^\infty e^{-a^2w^2} \ln w dw = \frac{\sqrt{\pi}}{2a}[\ln T + G(aT) - F(aT)] = -\frac{\sqrt{\pi}}{2a}\left[\frac{\gamma}{2} + \ln(2a)\right].$$

A change of variables $w^2 = t$, $a^2 = p$ gives the Laplace transform

$$\int_0^\infty e^{-pt} \frac{\ln t}{\sqrt{t}} dt = -\frac{\sqrt{\pi}}{\sqrt{p}}[\gamma + \ln(4p)], \qquad p > 0.$$

We also note that if we change variables $w = \sqrt{x/a}$ in

$$I(a,T) = \int_T^\infty e^{-a^2w^2} \ln w dw = \frac{\sqrt{\pi}}{2a}[\operatorname{erfc}(aT)\ln T + G(aT)],$$

$$= \frac{1}{4\sqrt{a}}\int_{aT^2}^\infty e^{-ax} \frac{\ln x}{\sqrt{x}} dx - \frac{\ln a}{4\sqrt{a}}\int_{aT^2}^\infty \frac{e^{-ax}}{\sqrt{x}} dx$$

we get, with $T = 1/\sqrt{a}$,

$$I\left(a, \frac{1}{\sqrt{a}}\right) = \frac{1}{4\sqrt{a}}\int_1^\infty e^{-ax} \frac{\ln x}{\sqrt{x}} dx - \frac{\sqrt{\pi}\ln a}{4a}\operatorname{erfc}(\sqrt{a})$$

But,

$$\int_1^\infty e^{-ax} \frac{\ln x}{\sqrt{x}} dx = -\left[\frac{\partial}{\partial \nu} E_\nu(a)\right]_{\nu=1/2}$$

Therefore, setting $T = 1/\sqrt{a}$ in the expression for $I(a,T)$ above, we have

$$\left[\frac{\partial E_\nu(a)}{\partial \nu}\right]_{\nu=1/2} = \frac{-2\sqrt{\pi}}{\sqrt{a}} G(\sqrt{a})$$

Asymptotics for $F(X)$ and $G(X)$

If we take X large, $w \ge X$, $X \to \infty$

$$\mathrm{erfc}(w) = \frac{e^{-w^2}}{w\sqrt{\pi}} \sum_{k=0}^{N} \frac{(-1)^k (1/2)_k}{w^{2k}} + R_N, \qquad |R_N| \le \frac{e^{-w^2}}{w\sqrt{\pi}} \frac{(1/2)_{N+1}}{w^{2N+2}}$$

and integrate, we get

$$G(X) = \frac{1}{\sqrt{\pi}} \sum_{k=0}^{N} (-1)^k \left(1/2\right)_k \int_X^\infty \frac{e^{-w^2}}{w^{2k+2}} dw + W_N,$$

$$= \frac{1}{2X\sqrt{\pi}} \sum_{k=0}^{N} \frac{(-1)^k (1/2)_k}{X^{2k}} E_{k+3/2}(X^2) + W_N, \quad |W_N| \le \frac{1}{2X\sqrt{\pi}} \frac{(1/2)_{N+1}}{X^{2N+2}} E_{N+5/2}(X^2)$$

To first terms,

$$G(X) = \frac{1}{2X\sqrt{\pi}} \left[\frac{e^{-X^2}}{X^2} + O\left(\frac{e^{-X^2}}{X^4} \right) \right] \qquad \text{for } X \to \infty$$

$$F(X) = \frac{\gamma}{2} + \ln(2X) + \frac{e^{-X^2}}{2X^3\sqrt{\pi}} + O\left(\frac{e^{-X^2}}{X^5} \right) \qquad \text{for } X \to \infty$$

For $X \to 0$, we have

$$\mathrm{erf}(w) = \frac{2w}{\sqrt{\pi}} \sum_{k=0}^{\infty} \frac{(-1)^k w^{2k}}{k!(2k+1)}$$

and

$$F(X) = \frac{2X}{\sqrt{\pi}} \sum_{k=0}^{\infty} \frac{(-1)^k X^{2k}}{k!(2k+1)^2} \qquad \text{for } X \to 0$$

$$G(X) = F(X) - \frac{\gamma}{2} - \ln(2X) \qquad \text{for } X \to 0$$

Computational Forms

$X \le 1$: The simplest expression for small X is

$$F(X) = \frac{2X}{\sqrt{\pi}} \sum_{k=0}^{\infty} \frac{(-1)^k X^{2k}}{k!(2k+1)^2} \qquad X \le 1$$

which converges rapidly and we compute the tail by

$$G(X) = -\frac{\gamma}{2} - \ln(2X) + F(X), \qquad X \le 1$$

<u>$X > 1$</u>: For $X > 1$ we develop the computational procedure for $G(X)$ and use the basic relation

$$F(X) = \frac{\gamma}{2} + \ln(2X) + G(X), \qquad X > 1$$

to compute $F(X)$ for $X > 1$. We have a rapidly terminating asymptotic expression for $G(X)$ for X large ($X \ge 7$) but no good expression for (1,7]. The strategy here is to develop Chebyshev series on the interval (1,7] and use the asymptotic expression for $G(X)$ for $X \to \infty$ for $X \ge 7$. To keep the Chebyshev coefficients relatively small, we divide [1,7] into subintervals [1,3], [3,5] and [5,7] and approximate

$$f(x) = x^3 e^{x^2} G(x).$$

This form removes the dominant asymptotic behavior for sizable x and makes $f(x)$ smoother and more nearly constant which produces accurate approximations in fewer terms. The computation of $G(x)$ is done by a double precision quadrature with DGAUS8. The Chebyshev algorithm is described in the next section.

<u>ALGORITHM I FOR $G(X)$</u>

$$G(X) = \frac{2X}{\sqrt{\pi}} \sum_{k=0}^{\infty} \frac{(-1)^k X^{2k}}{k!(2k+1)^2} - \frac{\gamma}{2} - \ln(2X) \qquad X \le 1$$

$$= \text{Chebyshev Expansions} \qquad 1 < X \le 7$$

$$= \frac{1}{2X\sqrt{\pi}} \sum_{k=0}^{N} \frac{(-1)^k (1/2)_k}{X^{2k}} E_{k+3/2}(X^2) \qquad 7 < X < \infty$$

<u>ALGORITHM II FOR $F(X)$</u>

$$F(X) = \frac{2X}{\sqrt{\pi}} \sum_{k=0}^{\infty} \frac{(-1)^k X^{2k}}{k!(2k+1)^2}, \qquad 0 \le X \le 1$$

$$= \frac{\gamma}{2} + \ln(2X) + G(X), \qquad 1 < X < \infty$$

where $G(X)$ is computed from Algorithm I on $1 < X < \infty$. Since the error in the asymptotic expansion for $X \geq 7$ is less than the next term, we terminate when a term relative to the accumulated sum is less than 0.5×10^{-14}. $X \geq 7$ guarantees that the sum terminates with this relative error in less than 20 terms.

<u>The Chebyshev Algorithm for $G(X)$</u> In reference [1], a procedure for computing Chebyshev coefficients a_r, $r = 0, n$ in the approximation of $f(x)$ by

$$S_n(x) = \frac{a_o}{2} + \sum_{r=1}^{n} a_r T_r(x) \qquad\qquad -1 \leq x \leq 1$$

is given. This high accuracy computation of $f(x)$ for the generation of the a_r coefficients is accomplished for

$$f(x) = x^3 e^{x^2} G(x)$$

by DGAUS8 quadrature on $G(X)$ with a relative error 0.5×10^{-14} on each of the intervals [1,3], [3,5], [5,7]. The algorithm for summing the $S_n(x)$ representation above is

$$b_{n+1} = b_{n+2} = 0$$

$$b_r = 2xb_{r+1} - b_{r+2} + a_r, \qquad\qquad r = n, n-1, \ldots, 0$$

and

$$S_n(x) = xb_1 - b_2 + \frac{a_o}{2} = \frac{1}{2}(b_o - b_2)$$

then, numerically,

$$G(x) = \frac{e^{-x^2}}{x^3} S_n(x).$$

Notice that this recurrence evaluates S_n in just n multiplications, not $O(n^2)$ multiplications by direct computation of each T_r or $2n$ by using the T_r recurrence. For an interval $A \leq X \leq B$, $-1 \leq x \leq 1$ is mapped onto $[A,B]$ by

$$X = \frac{(B-A)}{2}x + \frac{B+A}{2} \qquad \text{or} \qquad x = [2X - (B+A)]/(B-A).$$

The Chebyshev coefficients are given at the end of this Folder in the form of FORTRAN DATA statements for each interval $[A,B]$. n is selected for each interval so that the relative errors are $O(10^{-13})$.

[1] Clenshaw, C.W., *Mathematical Tables*, Vol. 5, Chebyshev Series for Mathematical Functions, NPL Physical Laboratory, Dept. of Scientific and Industrial Research, Her Majesty's Stationery Office, 1962, Reprinted 1963.

Acceleration of the Convergence of the $G(X)$ Series

In Folder 6, we developed the representation

$$G(X) = \int_X^\infty \frac{\text{erfc}(x)}{x}\,dx = \frac{1}{2}E_1(X^2) - \frac{1}{2\pi}\sum_{k=0}^{\infty} \frac{C_k}{(k+1/2)} E_{k+3/2}(X^2)$$

$$C_k = \frac{(1/2)_k}{k!} \sim \frac{k^{-1/2}}{\sqrt{\pi}} \quad \text{for } k \to \infty$$

and by one application of the recurrence

$$E_{k+3/2}(X^2) = \frac{e^{-x^2} - X^2 E_{k+1/2}(X^2)}{(k+1/2)}$$

we obtained

$$G(X) = \frac{1}{2}E_1(X^2) - \frac{S_1}{2\pi}e^{-X^2} + S_{10}\left(\frac{X^2}{2\pi}\right)E_{1/2}(X^2) + \frac{X^2}{2\pi}\sum_{k=0}^{\infty} S_{1,k+1}E_{k+3/2}(X^2)$$

where

$$S_1 = \sum_{k=0}^{\infty} \frac{C_k}{(k+1/2)^2} = 2\pi \ln 2, \quad E_{1/2}(X^2) = \sqrt{\pi}\,\frac{\text{erfc}(X)}{X}, \quad S_{1,k} = \frac{C_k}{(k+1/2)^2}, \quad k \geq 0.$$

Notice that the convergence of the series was enhanced from $O(k^{-3/2})$ to $O(k^{-5/2})$ by the application of the recurrence. Splitting out the first term ($k = 0$) produced

$$S_{10}\left(\frac{X^2}{2\pi}\right)E_{1/2}(X^2), \qquad S_{10} = \frac{C_o}{(1/2)^2} = 4$$

while re-indexing the remainder of the series got the index on E back to $k + 3/2$ from $k + 1/2$. Now we see that we can reapply the recurrence, split out the first term of the series and re-index the series to keep the index on E at $k + 3/2$. We compute the result after n applications of the recurrence. Let

$$
\begin{aligned}
H(X) &= \sum_{k=0}^{\infty} \frac{C_k}{(k+1/2)} E_{k+3/2}(X^2) \\
&= e^{-X^2} \sum_{k=1}^{n} (-1)^{k-1} S_k X^{2k-2} - X\sqrt{\pi}\,\mathrm{erfc}(X) \sum_{k=1}^{n} (-1)^{k-1} S_{k,o} X^{2k-2} \\
&\quad + (-X^2)^n \sum_{k=0}^{\infty} S_{n,k+1} E_{k+3/2}(X^2)
\end{aligned}
$$

where

$$
S_{1,k} = \frac{C_k}{(k+1/2)^2}, \qquad S_{n+1,k} = \frac{S_{n,k+1}}{(k+1/2)}, \qquad k \geq 0, \qquad S_n = \sum_{k=0}^{\infty} S_{n,k}, \qquad n \geq 1
$$

Explicitly,

$$
S_{n,k} = \frac{(1/2)_k}{(n+k-1)!(n+k-1/2)^2}, \qquad k \geq 0, \qquad n \geq 1
$$

Notice that the terms $S_{n,k} = O(k^{-n-3/2})$ and the convergence is enhanced with a truncation error estimate at N terms of

$$
T_N \approx \int_N^{\infty} \frac{dk}{k^{n+3/2}} = \frac{1}{(n+1/2)N^{n+1/2}}
$$

While it seems that one should take n large, the polynomial evaluations in X^2 must be considered. Thus for $X \leq 1$ one can expect no adverse effects from the polynomial evaluations. However, as X gets large, one can expect losses of significance by differences of large powers. Thus, there is a tradeoff to be made when using this formula for computation.

There is one last manipulation which will reduce losses of significance when X is large. Notice that by splitting out $S_{n,o}$ from S_n,

$$
S_n = S_{no} + W_n, \qquad W_n = \sum_{k=1}^{\infty} S_{n,k}, \qquad n \geq 1
$$

the $S_{k,o}$ sums combine to give

$$
e^{-X^2} \sum_{k=1}^{n} W_k (-X^2)^{k-1} + \sqrt{\pi}\, i\mathrm{erfc}(X) \sum_{k=1}^{n} S_{k,o} (-X^2)^{k-1}
$$

since $\displaystyle D \equiv \frac{e^{-X^2}}{\sqrt{\pi}} - X\mathrm{erfc}(X) = i\mathrm{erfc}(X).$

By replacing this difference with $i$erfc(X), we avoid losses of significance when X is large because $i$erfc(X) can be computed accurately. More precisely, $D = O(e^{-X^2}/X^2)$ which shows that we lose 2 significant digits for X as low as 10. $S_{n,o}$, W_n and S_n are tabulated at the end of this folder. The number of terms of W_n was computed from the truncation error estimate T_N divided by $S_{n,1}$ (a lower bound on W_n).

The final result is

$$H(X) = e^{-X^2}\sum_{k=1}^{n} W_k(-X^2)^{k-1} + \sqrt{\pi}ierfc(X)\sum_{k=1}^{n} S_{k,0}(-X^2)^{k-1} + (-X^2)^n \sum_{k=0}^{\infty} S_{n,k+1}E_{k+3/2}(X^2)$$

and

$$G(X) = \frac{1}{2}E_1(X) - \frac{1}{2\pi}H(X)\,.$$

The estimates

$$S_{n,k} = \frac{(1/2)_k}{[1\cdot 2\ldots n](n+1)\ldots(n+k-1)(n+k-1/2)^2}, \qquad n \geq 1, \qquad k \geq 0$$

$$S_{n,k} \leq \frac{(1/2)_k}{n!k!(k+1/2)^2}, \qquad n \geq 1$$

$$W_n = \sum_{k=1}^{\infty} S_{n,k} \leq \frac{1}{n!}\sum_{k=0}^{\infty} \frac{(1/2)_k}{k!(k+1/2)^2} = \frac{2\pi \ln 2}{n!}$$

show that the terms of $S_{n,o}$ and W_n go down like a small constant divided by $n!$. Therefore, in order to keep $W_k X^{2k-2}$ and $S_{k,o}X^{2k-2}$ from growing excessively and losing significant digits, we need $X^2 < n/e$ or $X < \sqrt{n/e}$. The error estimate T_N for $n = 6$ with $N = 100$ terms of the series gives an acceptable truncation error $O(10^{-13})$ since

$$E_{k+3/2}(X^2) \sim \frac{e^{-X^2}}{X^2 + k + 3/2} \quad \text{for } k \to \infty$$

acts more like a scale factor for small k. The degradation of the relative error for $X > \sqrt{n/e}$ with $N = 125$ and $6 \leq n \leq 10$ was confirmed numerically, though decimal place accuracies were retained.

Nevertheless, this formula still has theoretical value and shows how to accelerate the convergence of series containing exponential integrals.

CHEBYSHEV COEFFICIENTS FOR G(X)

The following coefficients were computed in multiple precision arithmetic using the MP package developed by David M. Smith and published as Algortihm 693 in the ACM Trans Math Software, Vol. 17,
No. 2, June, 1991, pp 273-283.

```
C     CHEBYSHEV COEFFICIENTS FOR INTERVAL (1,3]
               DATA (COE(I,1), I=1,24)/
     1      0.183936201622926800D+00,      0.602481282921201200D-01,
     2     -0.119525542040734300D-01,      0.163048183862326200D-02,
     3     -0.916761486984451800D-04,     -0.300245124943356100D-04,
     4      0.136020773297374100D-04,     -0.357151573002627300D-05,
     5      0.743773093967212500D-06,     -0.131331376113140100D-06,
     6      0.196950958343117800D-07,     -0.233463166825802300D-08,
     7      0.141611588786531800D-09,      0.304836446286498300D-10,
     8     -0.158873383783807400D-10,      0.471064076171147000D-11,
     9     -0.117254478451858600D-11,      0.268672719869292500D-12,
     A     -0.589487969074253200D-13,      0.126554771091031300D-13,
     B     -0.269424003935865400D-14,      0.573660397938811300D-15,
     C     -0.122809738646140600D-15,      0.265136030774818600D-16/
C     CHEBYSHEV COEFFICIENTS FOR INTERVAL (3,5]
                  DATA (COE(I,2), I=1,18)/
     1      0.249911966143169100D+00,      0.137407615980377800D-01,
     2     -0.211076375427525400D-02,      0.274149770820318000D-03,
     3     -0.315160070674318100D-04,      0.324067726462838300D-05,
     4     -0.293779756592443700D-06,      0.221272952269653000D-07,
     5     -0.108159106183739300D-08,     -0.393712006838136900D-10,
     6      0.211188488938613800D-10,     -0.394325155773269300D-11,
     7      0.574358099846202900D-12,     -0.736872227670002800D-13,
     8      0.868815019030334600D-14,     -0.959589743808029100D-15,
     9      0.100178090811985800D-15,     -0.991516948990920300D-17/
C     CHEBYSHEV COEFFICIENTS FOR INTERVAL (5,7)
                  DATA (COE(I,3), I=1,14)/
     1      0.267032022166309300D+00,      0.465764615292050200D-02,
     2     -0.529406889059992700D-03,      0.522543466857855500D-04,
     3     -0.471680853238733800D-05,      0.398010531483489200D-06,
     4     -0.317099625639186200D-07,      0.239340951085099200D-08,
     5     -0.170820332342142600D-09,      0.114336576815315400D-10,
     6     -0.703928286525811000D-12,      0.380529299822365100D-13,
     7     -0.156156454824444000D-14,      0.110122861980293900D-16/
             DATA EULER /0.5772156649015328610D0/
             DATA RTPI  /1.772453850905516027D0/
```

S(n,0), W(n), S(n)
IN G(X) FORMULAE WITH
ACCELERATED CONVERGENCE OF THE SERIES
LIMITED TO 3,000,000 TERMS

n	SUM	# TERMS
1	0.4000000000000000E+01	3000000
	0.3551721805347100E+00	
	0.4355172180534709E+01	
2	0.4444444444444444E+00	2512000
	0.5879072135855803E-01	
	0.5032351658030025E+00	
3	0.8000000000000000E-01	57000
	0.9336136051502611E-02	
	0.8933613605150262E-01	
4	0.1360544217687075E-01	8000
	0.1342179865788014E-02	
	0.1494762204265876E-01	
5	0.2057613168724280E-02	2000
	0.1730351809922372E-03	
	0.2230648349716517E-02	
6	0.2754820936639118E-03	829
	0.2006068029221810E-04	
	0.2955427739561299E-03	
7	0.3287310979618672E-04	447
	0.2104387364280908E-05	
	0.3497749716046763E-04	
8	0.3527336860670194E-05	282
	0.2011131912775020E-06	
	0.3728450051947696E-05	
9	0.3432745647278519E-06	198
	0.1762558608336856E-07	
	0.3609001508112205E-06	
10	0.3053442573294836E-07	151
	0.1425116586260281E-08	
	0.3195954231920864E-07	

Accurate Values for n=1

	2*PI*LN(2) FOR S(1)
1	0.4355172180607204E+01
	2*PI*LN(2)-4 FOR W(1)
1	0.355172180607204E+01

Folder 17

Notes on the Evaluation of

$$\boldsymbol{I(a,b,X)=\int_X^\infty e^{-at-b/t}\,dt\,,\qquad a>0,\ \ b>0,\ \ X>0}$$

Donald E. Amos, February 2002

Summary

We derive series Expansions I and II for I(a,b,X) with Expansion I expressing the behavior for large X and Expansion II the behavior for small X. In addition, the series of reference [1, Eq. 2.168] for the generalized incomplete gamma function

$$\Gamma(\alpha,x;b)=\int_x^\infty t^{\alpha-1}e^{-t-b/t}\,dt$$

is extended from non-integer α to the integer case for $\alpha=1$ and is shown to be identical to Expansion I for a=1. That is, $I(a,b,X)=I(1,ab,aX)/a=\Gamma(1,aX;ab)/a$

Expansion I

Use the series for $e^{-b/t}$ and integrate

$$\begin{aligned}I&=\sum_{k=0}^\infty \frac{(-1)^k}{k!}b^k\int_X^\infty \frac{e^{-at}}{t^k}\,dt\\&=X\sum_{k=1}^\infty \frac{(-1)^k}{k!}\left(\frac{b}{X}\right)^k E_k(aX)+\int_X^\infty e^{-at}\,dt\\&=\frac{e^{-aX}}{a}+X\sum_{k=1}^\infty \frac{(-1)^k}{k!}\left(\frac{b}{X}\right)^k E_k(aX)\end{aligned}$$

This series converges for all *a*, *b* and *X*, but its usage is best for $b/X\le 2$ where significant digits can be assured. (i.e. Large *X* expansion).

Special Case, $X=0$ and evaluation of a Laplace Transform

$$I(a,b,0)=\int_0^\infty e^{-at-b/t}\,dt=2\sqrt{\frac{b}{a}}K_1(2\sqrt{ab})$$

Expansion II Notice

$$I(a,b,X)=I(a,b,0)-\int_0^X e^{-at-b/t}\,dt,\qquad\qquad t=\frac{1}{v},$$

$$I(a,b,X) = 2\sqrt{\frac{b}{a}}K_1(2\sqrt{ab}) - \int_{1/X}^{\infty} \frac{e^{-bv-a/v}}{v^2}dv$$

$$= 2\sqrt{\frac{b}{a}}K_1(2\sqrt{ab}) - X\sum_{k=0}^{\infty} \frac{(-1)^k}{k!}(aX)^k E_{k+2}\left(\frac{b}{X}\right)$$

for the small X expansion. Again, $aX \le 2$ gives rapid convergence and retention of significant digits.

Analytic Integrand We change the variables to convert the integrand to an analytic function which is suitable for further manipulation and evaluation by an accurate quadrature.

Let $t = \sqrt{\frac{b}{a}}u.$ Then

$$I(a,b,X) = \sqrt{\frac{b}{a}}\int_{X\sqrt{a/b}}^{\infty} e^{-\sqrt{ab}(u+1/u)}du$$

Now let $u = e^v$,

$$I(a,b,X) = \sqrt{\frac{b}{a}}\int_{\ln(X\sqrt{a/b})}^{\infty} e^{-2\sqrt{ab}\cosh v+v}dv$$

Comments: The expansions above cover the cases where X is large and X is small. The restrictions $b/X \le 2$ and $aX \le 2$ can be violated if one is willing to accept some losses of significance and accept absolute accuracy. A quadrature with DGAUS8 will always produce significant digits since the integrand is positive everywhere.

Comments on 2.168 of Chaudhry and Zubair: Equation 2.168 has both positive powers of X and negative powers of X when $m > n$ and $m < n$ respectively and becomes indeterminant for both large and small X. The best results are obtained *near* the minimum of the expression

$$f(x) = ax + b/x \qquad \text{or} \qquad x = \sqrt{\frac{b}{a}}$$

where $b/x = \sqrt{ab}$ and $ax = \sqrt{ab}$ and $\sqrt{ab}$ is not large.

Evaluation of Equation 2.163 of Chaudhry and Zubair: Equation 2.163 is the start of the evaluation of

$$\Gamma(\alpha,x:b) = \int_x^{\infty} t^{\alpha-1}\exp(-t-b/t)dt = \frac{1}{2\pi i}\sum_{n=0}^{\infty}\frac{(-b)^n}{n!}\int_{c-i\infty}^{c+i\infty}\frac{\Gamma(s)x^{\alpha-n-s}}{s+n-\alpha}ds$$

which for $\alpha = 1$ is $I(1, b, x)$. Notice that a change of variables in $I(a,b,X)$ can reduce the 3 parameters down to 2 with $t = v/a$.

Now, when α is *not* an integer, the integrand has simple poles at $s = -m$ from $\Gamma(s)$ and $s = \alpha - n$ from $s+n-\alpha$ and Chaudhry and Zubair compute this case. However if $\alpha > 0$ is an integer, there are double poles in the integrand at some of the negative integers from $s+n-\alpha$ and $\Gamma(s)$. We consider the case where $\alpha = 1$ to represent $I(1, b, x)$.

<u>$\alpha = 1$</u>

Equation 2.164 gets the correct residues for the integral from the pole of $\Gamma(s)$ when $s = -m$ is not equal to $s = 1-n$, $n \geq 1$. Therefore we concentrate on getting the residues for the case $n = 0$ where the denominator has a simple pole and the cases where $s = 1-n$, $n \geq 1$ where the integrand has a double pole. Then we have

$$\frac{x^{1-n}}{2\pi i}\int_{c-i\infty}^{c+i\infty}\frac{\Gamma(s)x^{-s}}{s+n-1}ds \equiv \left[\sum_{\substack{m=0\\ m\neq n-1}}^{\infty}\frac{(-1)^m x^m}{m!(n-m-1)} + \begin{pmatrix}\textbf{Residues from}\\ s=1 \textbf{ and } s=1-n\end{pmatrix}\right]x^{1-n}$$

<u>Residue for $s = 1$ (n = 0)</u> This is a simple pole since the poles of $\Gamma(s)$ lie in the left half plane and $s = 0$. Then

$$\text{Residue} = \lim_{s\to 1}(s-1)\cdot\frac{\Gamma(s)x^{-s}}{s-1} = \frac{1}{x}$$

and the term to be added above is $\left(\frac{1}{x}\right)\cdot x = 1$

<u>Residues for $s = 1-n$, $n \geq 1$</u> Let $s = 1-n+t$ and let $t \to 0$ to find the residues:

$$\frac{\Gamma(s)x^{-s}}{s+n-1} = \frac{\Gamma(1-n+t)}{t}x^{n-1}e^{-t\ln x}$$

It is clear that there is a double pole at $t = 0$. Now we apply

$$\Gamma(z)\Gamma(-z) = \frac{-\pi}{z\sin\pi z}$$

to make the expansion about $t = 0$

$$\Gamma(1-n+t) = \frac{\pi}{(n-1-t)\Gamma(n-1-t)\sin\pi(1-n+t)} = \frac{\pi(-1)^{n-1}}{\Gamma(n-t)\sin\pi t}$$

and

$$\frac{\Gamma(s)x^{-s}}{s+n-1} = \frac{\pi(-1)^{n-1}e^{-t\ln x}x^{n-1}}{\Gamma(n-t)\cdot t\cdot \sin\pi t}$$

In this form, the double pole at $t = 0$ is apparent. One could make the power series expansions for $e^{-t\ln x}$, $\Gamma(n-t)$, and $\sin\pi t$ and manipulate to get the power series in t^{-2}, t^{-1}, t^{0}, … but the residue is more easily calculated from the formula

$$\begin{array}{c}\text{Residue}\\ (\text{Double pole at } z=z_o)\end{array} = \frac{2p'(z_o)}{q''(z_o)} - \frac{2}{3}\frac{p(z_o)q'''(z_o)}{[q''(z_o)]^2}$$

for ratios $p(z)/q(z)$. Now, we expand the numerator and denominators in powers of t and identify the required derivatives from the coefficients since $z_o = 0$. Now

$$p(t) = e^{-t\ln x} = 1\text{-}t(\ln x)+O(t^2)$$

Then

$$p(0) = 1, \qquad \frac{p'(0)}{1!} = -\ln x$$

For the denominator,

$$q(t) = \Gamma(n-t)\cdot t\cdot \sin\pi t$$

and

$$\Gamma(n-t) = \Gamma(n)-\Gamma'(n)t+O(t^2)$$

$$t\sin\pi t = \pi t^2 - \frac{\pi^3 t^4}{3!} + O(t^6).$$

Then

$$q(t) = \pi\Gamma(n)t^2-\pi\Gamma'(n)t^3+O(t^4)$$

Therefore

$$\frac{q''(0)}{2!} = \pi\Gamma(n), \qquad \frac{q'''(0)}{3!} = -\pi\Gamma'(n)$$

Now the residue at $s = 1-n$ is

$$\begin{matrix}\text{Residue} \\ (s=1-n)\end{matrix} = \left[\frac{-2\ln x}{2\pi\Gamma(n)} + \frac{2}{3}\cdot\frac{6\pi\Gamma'(n)}{4\pi^2\Gamma^2(n)}\right]\cdot \pi(-x)^{n-1} = \frac{(-x)^{n-1}}{\Gamma(n)}[-\ln x + \psi(n)]$$

The term to be added from the residue is

$$(\text{residue})x^{1-n} = \frac{(-1)^{n-1}}{\Gamma(n)}[-\ln x + \psi(n)]$$

Therefore the final form after summing on n is

$$\Gamma(1, x : b) = x\sum_{n=0}^{\infty} \frac{(-1)^n}{n!}\left(\frac{b}{x}\right)^n \sum_{\substack{m=0 \\ m\neq n-1}}^{\infty} \frac{(-x)^m}{m!(n-m-1)} + 1 + \sum_{n=1}^{\infty} \frac{b^n}{n!(n-1)!}[\ln x - \psi(n)]$$

Theoretically, one could take the limit as $\alpha \to 1$ in 2.168, but it is clear that the residue approach is easier.

Notice also that (Abramowitz and Stegen, Chapter 5)

$$\sum_{\substack{m=0 \\ m\neq n-1}}^{\infty} \frac{(-x)^m}{m!(n-m-1)} = E_n(x) - \frac{(-x)^{n-1}}{(n-1)!}[-\ln x + \psi(n)], \qquad n \geq 1$$

Then

$$\begin{aligned}\Gamma(1,x:b) = & -x\sum_{m=0}^{\infty} \frac{(-x)^m}{m!(m+1)} + x\sum_{n=1}^{\infty} \frac{(-1)^n}{n!}\left(\frac{b}{x}\right)^n E_n(x) \\ & - x\sum_{n=1}^{\infty} \frac{(-1)^n b^n}{n! x^n}\cdot\frac{(-x)^{n-1}}{(n-1)!}[-\ln x + \psi(n)] + 1 + \sum_{n=1}^{\infty} \frac{b^n}{n!(n-1)!}[\ln x - \psi(n)] \\ = & -x\left[\frac{1}{x}\int_0^x e^{-t}dt\right] + x\sum_{n=1}^{\infty} \frac{(-1)^n}{n!}\left(\frac{b}{x}\right)^n E_n(x) + 1 \\ = & \; e^{-x} + x\sum_{n=1}^{\infty} \frac{(-1)^n}{n!}\left(\frac{b}{x}\right)^n E_n(x)\end{aligned}$$

which is Expansion I with $a = 1$.

Reference

[1] Chaudhry, M.A. and Zubair, S.M., *On a Class of Incomplete Gamma Functions with Applications*, Chapman & Hall/CRD, Boca Raton, 2001.

Folder 18

Evaluation of

$$\boldsymbol{E_\nu(x) = \int_1^\infty \frac{e^{-xt}}{t^\nu}dt, \quad G_\nu(x) = \int_1^\infty \frac{E_\nu(xt)}{t^\nu}dt = -\frac{\partial E_\nu(x)}{\partial \nu} = \int_1^\infty \frac{e^{-xt}\ln t}{t^\nu}dt, \quad x > 0}$$

For Integer and Half Odd Integer Orders with Application to

$$\boldsymbol{I_n(b,T) = \int_T^\infty \frac{e^{-b^2x^2}\ln x}{x^n}dx, \qquad n = 0, 1, 2, \ldots, \qquad T > 0}$$

Donald E. Amos, April 2002

Summary

Folder 18 is composed of subfolders 18a and 18b. Folder 18a provides the documentation necessary to compute K member sequences $E_{N+\alpha}(x)$, …, $E_{N+\alpha+K-1}(x)$, $G_{N+\alpha}(x)$, …, $G_{N+\alpha+K-1}(x)$ for $\alpha = 0$ or $\alpha = ½$. The resulting codes are designated by DEXINT, DHEXINT, DGEXINT, DGHEXINT for $E_n(x)$, $n \ge 1$; $E_{n+1/2}(x)$, $n \ge 0$; $G_n(x)$, $n \ge 1$; and $G_{n+1/2}(x)$, $n \ge 0$ respectively. Emphasis on stable recurrence is foremost in the development. Since $G_\nu(x)$ is closely related to $E_\nu(x)$, order of magnitude estimates (bounds) for $G_\nu(x)$ similar to those for $E_\nu(x)$ are derived.

$G_{1/2}(x)$ is also related to G(x) of Folder 16 by

$$G_{1/2}(x) = 2\sqrt{\frac{\pi}{x}}G(\sqrt{x}) = -\left[\frac{\partial E_\nu(x)}{\partial \nu}\right]_{\nu=1/2} = \int_1^\infty \frac{e^{-xt}\ln t}{\sqrt{t}}dt\ .$$

Folder 18b shows how to apply the results of Folder 18a to generate either single values or sequences for the function $I_n(b,T)$.

Folder 18a

Introduction

The relation

$$\int_x^\infty \frac{E_\mu(t)}{t^\nu}\,dt = \frac{1}{x^{\nu-1}}\,\frac{E_\nu(x)-E_\mu(x)}{\mu-\nu}, \qquad \mu \neq \nu$$

or equivalently

$$\int_1^\infty \frac{E_\mu(xt)}{t^\nu}\,dt = \frac{E_\nu(x)-E_\mu(x)}{\mu-\nu}, \qquad \mu \neq \nu$$

can be verified by differentiation w.r.t. x. We have used this relation in previous folders for $\mu \neq \nu$. Notice that if $\mu \to \nu$, we get the case where $\mu = \nu$ on the left, and

$$G_\nu(x) \equiv \int_1^\infty \frac{E_\nu(xt)}{t^\nu}\,dt = \frac{-\partial E_\nu(x)}{\partial \nu} = \int_1^\infty \frac{e^{-xt}\ln t}{t^\nu}\,dt$$

or

$$\int_x^\infty \frac{E_\nu(t)}{t^\nu}\,dt = \frac{1}{x^{\nu-1}}\,G_\nu(x), \qquad \mu = \nu$$

It is the purpose of this folder to provide the documentation for the computation of $G_\nu(x)$ for integer and half odd integer orders and $E_\nu(x)$ for half odd integer orders. The plan is to review the computation of $E_n(x)$, $n = 1, 2, \ldots$ from [1] and show how the analysis can be modified to include the computation of these new functions. It follows that the corresponding code, denoted by EXPINT in the ACM Collected Algorithms [2] and EXINT or DEXINT in the SLATEC Library, can also be modified to incorporate appropriate changes and/or additions.

The beauty of this process is that only new initial values need be supplied to get sequences of these new functions, and the coding structure of DEXINT remains (almost) unchanged to construct new codes. The (double precision) codes for $E_n(x)$, $G_n(x)$, $n \geq 1$ and $E_{n+1/2}(x)$, $G_{n+1/2}(x)$, $n \geq 0$ are called DEXINT, DGEXINT, DHEXINT, DGHEXINT respectively. (H = half odd order routine). K member sequences $\{n+\alpha, \ldots, n+\alpha+K-1\}$ $\alpha = 0$ or ½ as well as exponential scaling are options in each of these subroutines.

Computational feasibility is the main reason why we consider only integer and half odd integer orders. The Miller procedure, which is used to compute these functions for larger $x > 2$, can be applied for general orders. However the convergence is poor for $x < 2$ which results in large amounts of computation. Therefore, we need formulas to cover the region $x \leq 2$ and these are obtained fairly easily for the integer and half odd integer cases.

What, of course, makes these routines worthwhile is that the integer and half odd integer orders arise in many applications.

<u>Basic Properties of $E_\nu(x)$ and $G_\nu(x)$, $\nu > 1$</u>

The basic inequalities for $E_\nu(x)$ [3, Chapter 5]

$$E_\nu(x) = \int_1^\infty \frac{e^{-xt}}{t^\nu} dt, \qquad \nu > 0$$

can be found in most handbooks. These inequalities come from the relations

$$\nu E_{\nu+1}(x) + xE_\nu(x) = e^{-x}$$

$$E_{\nu+1}(x) < E_\nu(x) .$$

Replacement on the left gives

$$E_{\nu+1}(x) \le \frac{e^{-x}}{x+\nu} \text{ and } E_\nu(x) > \frac{e^{-x}}{x+\nu}.$$

Combining these we have [3, p. 229]

$$\frac{e^{-x}}{x+\nu} < E_\nu(x) \le \frac{e^{-x}}{x+\nu-1} \qquad \nu \ge 1$$

We can generate similar inequalities for $G_\nu(x)$,

$$G_\nu(x) = \int_1^\infty \frac{E_\nu(xt)}{t^\nu} dt , \qquad \nu > 0.$$

We take the recurrence above for $E_\nu(xt)$ divide by $t^{\nu+1}$ and integrate to get

$$\nu G_{\nu+1}(x) + xG_\nu(x) = E_{\nu+1}(x)$$

and note

$$G_{\nu+1}(x) = \int_1^\infty \frac{E_{\nu+1}(xt)}{t^{\nu+1}} < \int_1^\infty \frac{E_\nu(xt)}{t^{\nu+1}} dt < \int_1^\infty \frac{E_\nu(xt)}{t^\nu} dt = G_\nu(x), \qquad \nu > 0$$

Then, repeating the analysis above with E_ν replaced by G_ν we get

$$\frac{e^{-x}}{(x+\nu)(x+\nu+1)} < \frac{E_{\nu+1}(x)}{x+\nu} < G_\nu(x) \le \frac{E_\nu(x)}{x+\nu-1} \le \frac{e^{-x}}{(x+\nu-1)^2} \quad , \qquad \nu \ge 1$$

We also have

$$G'_\nu(x) = -G_{\nu-1}(x), \qquad \nu > 1$$

from $E'_\nu(x) = -E_{\nu-1}(x)$ and

$$G_\nu(0) = 1/(\nu-1)^2 \qquad \nu > 1$$

from $E_\nu(0) = 1/(\nu-1)$, $\nu > 1$.

Since the recurrence operator

$$L\{y\} = \nu y_{\nu+1}(x)+xy_\nu(x)$$

is used for both $\{E_\nu(x)\}$ and $\{G_\nu(x)\}$ sequences, we expect that the recurrence properties for $\{G_\nu(x)\}$ mimic those of $\{E_\nu(x)\}$because it is the homogeneous solutions of $L\{h\} = 0$ which determine stability and instability of the recurrence. That is, the amplification factors $h_{\nu+1}/h_\nu = -x/\nu$, $h_\nu/h_{\nu+1} = -\nu/x$ apply for forward recurrence and backward recurrence respectively. Thus, homogeneous solutions (which are introduced by an error in the computation, usually a rounding error) do not grow if we recur away from x because these ratios are less than 1. A more rigorous analysis on the coupled system

$$\begin{pmatrix} \nu & -1 \\ 0 & \nu \end{pmatrix}\begin{pmatrix} G_{\nu+1} \\ E_{\nu+1} \end{pmatrix} + \begin{pmatrix} x & 0 \\ 0 & x \end{pmatrix}\begin{pmatrix} G_\nu \\ E_\nu \end{pmatrix} = \begin{pmatrix} 0 \\ e^{-x} \end{pmatrix}$$

confirms this observation since the eigenvalues of the iteration matrices for forward and backward recurrence are $-x/\nu$ and $-\nu/x$ respectively.

General Description

The real problem is to compute an initial value to start the recurrence. Since we are interested in K member sequences

$$E_N(x),\ \ldots,\ E_{N+K\text{-}1},\ \ G_N(x),\ \ldots,\ G_{N+K\text{-}1},\ \ E_{N+/12}(x),\ \ldots,\ E_{N+K\text{-}1/2}(x),\ \ G_{N+1/2}(x),\ \ldots,\ G_{N+K\text{-}1/2}(x),$$

we need to know, for stable recurrence, where these sequences lie with respect to x. Let $N_x = [x+1/2]$ or $[x]+1/2$ be the integer or half odd integer closest to x. Then we have the following cases:

Case I: N_x lies to the left of all indices. In this case we generate the first member, a, of the sequence and recur forward.

Case II: N_x lies in the index set. In this case we generate the function for $a = N_x$ and recur forward and backward from the index a to complete the set.

Case III: N_x lies to the right of the index set. In this case we generate the function for the last index, a, and recur backward.

These cases have been incorporated into the computation of integer orders $E_{N+k}(x)$, $k = 0, K-1$ in reference [1] which resulted in the ACM Algorithm 556 for the code EXPINT [2] and the SLATEC library codes EXINT and DEXINT for single and double precision arithmetic.

Code modifications depend on observing that the overall code structure and recurrences are almost the same as those for DEXINT. This holds true for both integer and half odd integer orders. The major changes or additions come about in the computation of initial values at index a to start the recurrences. Unfortunately, one formula for $E_a(x)$ or $G_a(x)$ which covers all x with an acceptable amount of computation does not seem to exist. Specifically, the x variable is partitioned into two parts, $x \le XCUT$ and $x > XCUT$ with $XCUT = 2$ where acceptable formulas or procedures are available.

$x \le XCUT = 2$ The general procedure in [1] and in the code DEXINT uses the power series for $E_n(x)$ for $x \le 2$ and all $n \ge 1$. The problem for $G_n(x)$ is that we do not have a simple representation to cover all n. But we do have recurrence from $a = 1$ and $a = 2$. The same is true for $E_\nu(x)$ and $G_\nu(x)$ for ν a half odd order. $N_x = [x+0.5]$ or $[x]+0.5$ is close to $a = 1$ or 2 for the integer orders and $a = ½$ for the half odd orders, and there is at most one step where recurrence in the "wrong" or unstable direction occurs (i.e., $1 \le x \le 2$, $a = ½$). One step does not generally create an excessive error. Then with an initial index $\nu = a$ we recur forward with

$$\nu E_{\nu+1}(x) + xE_\nu(x) = e^{-x} \qquad \nu \ge a$$

to generate $E_{\nu+1}(x)$ and generate $G_{\nu+1}(x)$ from

$$\nu G_{\nu+1}(x) + xG_\nu(x) = E_{\nu+1}(x)$$

at each step in the same loop. The starting formulas for each code are:

DEXINT: For any integer $a \ge 1$, [1], [3]

$$E_a(x) = \frac{(-x)^{a-1}}{(a-1)!}[-\ln x + \psi(a)] - \sum_{\substack{m=0 \\ m \ne a-1}}^{\infty} \frac{(-x)^m}{(m-a+1)m!}$$

DHEXINT: $a = ½$ and $a = 3/2$

$$E_{1/2}(x) = \sqrt{\pi}\,\mathrm{erfc}\sqrt{x})/\sqrt{x}, \qquad E_{3/2}(x) = 2\sqrt{\pi}\,\mathrm{ierfc}(\sqrt{x})$$

DGEXINT: $a = 1$ and $a = 2$,

$$E_1(x) = -\gamma - \ln x - \sum_{n=1}^{\infty} \frac{(-x)^n}{nn!}$$

$$E_2(x) = x[\gamma + \ln x - 1] + 1 - \sum_{n=2}^{\infty} \frac{(-x)^n}{(n-1)n!}$$

$$G_1(x) = \frac{\pi^2}{12} + \frac{1}{2}E_1^2(x) - (\gamma + \ln x)\sum_{n=1}^{\infty} \frac{(-x)^n}{nn!} + \sum_{n=1}^{\infty} \frac{(-x)^n}{n^2 n!} - \sum_{k=2}^{\infty} \frac{(-x)^k}{k} A_k$$

$$G_2(x) = 1 - \frac{\pi^2}{12}x - \frac{x}{2}(\gamma + \ln x)^2 + x(\gamma + \ln x) - x - x\sum_{n=1}^{\infty} \frac{(-x)^n}{n^2(n+1)n!}$$

$$+ x\sum_{n=2}^{\infty} \frac{(-x)^n}{n(n+1)} A_n - \frac{x}{2}\sum_{n=2}^{\infty} \frac{(-x)^n}{n+1} D_n$$

where

$$A_n = \frac{1}{n!}\sum_{k=1}^{n-1} \frac{C_k^n}{n-k}, \qquad D_n = \frac{1}{n!}\sum_{k=1}^{n-1} \frac{C_k^n}{k(n-k)}, \qquad n \geq 2$$

and C_k^n are binomial coefficients.

See the sections on Computation of $G_1(x)$, $x \leq 2$ and Computation of $G_2(x)$, $x \leq 2$ for the derivations.

DGHEXINT: $a = ½$ and $a = 3/2$

$$E_{1/2}(x) = \sqrt{\pi}\,\mathrm{erfc}(\sqrt{x})/\sqrt{x}, \qquad G_{1/2}(x) = 2\sqrt{\frac{\pi}{x}}G(\sqrt{x})$$

$$E_{3/2}(x) = 2\sqrt{\pi}\,\mathrm{ierfc}(\sqrt{x}), \qquad G_{3/2}(x) = 2[E_{3/2}(x) - xG_{1/2}(x)]$$

where $G(x)$ is described in Folder 16 and computed from double precision function DGERFC. ierfc(x) is computed from double precision function DIERFC.

Miller Algorithm for $x > XCUT = 2$

The procedure in [1] for $x > XCUT$ uses the confluent hypergeometric function $U(a,a,x)$ to represent

$$E_a(x) = e^{-x}x^{a-1}U(a,a,x)$$

where a is an integer, $a > 2$. However there is nothing in the theory that restricts a to be an integer. Thus we apply the theory for all cases considered here. The remainder of the computation for error estimates, etc. remains the same. That is, the error estimates for a given tolerance produce a starting value M for backward recurrence on

$$y_{m+1}(x)-(c_m+xd_m)y_m(x)-a_m y_{m-1}(x) = 0$$

where

$$a_m = \frac{m(a+m-1)}{(m+a/2)} d_m, \qquad c_m = 2\left(m+\frac{a}{2}\right)d_m, \qquad d_m = \frac{1}{m+a/2+1}$$

and, explicitly,

$$y_m(x) = A_m U(m+a,a,x), \qquad A_m = \frac{(1)_m (a)_m}{(1+a/2)_m}, \qquad m \geq 0$$

In particular, $y_0(x) = U(a,a,x)$ and $E_a(x) = e^{-x}x^{a-1}y_0(x)$.

Since the normal procedure for backward recurrence starts with an index M and an asymptotic (~) estimate of the ratio y_{M+1}/y_M, the $\{\tilde{y}_m^M\}$ sequence is only determined to within a multiplicative constant. A normalizating relation is then used to pin down the constant of proportionality, C_M. In this case, the recurrence

$$aE_{a+1}(x)+xE_a(x) = e^{-x}$$

is used to normalize the $\{\tilde{y}_m^M\}$ sequence where $E_{a+1}(x)$ is computed in terms of $\tilde{y}_0^M$ and $\tilde{y}_1^M$ from the contiguous relation for $U(a+1, a+1, x)$ (this relation is presented below). The result is

$$E_a(x) \doteq C_M \tilde{y}_0^M, \qquad C_M = \frac{e^{-x}}{a[\tilde{y}_0^M - (1+a/2)\tilde{y}_1^M / a] + x\tilde{y}_0^M}$$

where $\tilde{y}_0^M$ and $\tilde{y}_1^M$ are computed by backward recurrence from M.

<u>Computation of $G_a(x)$ for $x > XCUT$</u>

Now we show how to compute $G_a(x)$ for $x > 2$. Define w_m by

$$w_m = \int_1^\infty \frac{e^{-xt}}{t} U(m+a,a,xt)dt \quad , \quad m \ \geq \ 0 \quad .$$

With $m = 0$ we have

$$w_o = \int_1^\infty \frac{e^{-xt}}{t} U(a,a,xt)dt = \frac{1}{x^{a-1}} \int_1^\infty \frac{E_a(xt)}{t^a} dt = \frac{1}{x^{a-1}} G_a(x)$$

Thus, to get w_m we integrate the y_m recurrence

$$y_{m+1}(xt)-(c_m+xtd_m)y_m(xt)+a_m y_{m-1}(xt) = 0$$

$$y_m(xt) = A_m U(m+a,a,xt)$$

with the factor $\exp(-xt)/t$ to get

$$A_{m+1}w_{m+1} - c_m A_m w_m - d_m A_m x\int_1^\infty e^{-xt}U(m+a,a,xt)dt + a_m A_{m-1}w_{m-1} = 0$$

Now, we manipulate the integral to get it in terms of w_m and w_{m+1}. Integration by parts

$$u = U(m+a,a,xt) \qquad dv = e^{-xt}dt$$

$$du = -(m+a)xU(m+a+1,a+1,xt)dt \qquad v = -\frac{e^{-xt}}{x}$$

leads to

$$\int_1^\infty e^{-xt}U(m+a,a,xt)dt = \frac{e^{-x}}{x}U(m+a,a,x) - (m+a)\int_1^\infty e^{-xt}U(m+a+1,a+1,xt)dt.$$

The contiguous relation [1], [3,13.4.18]

$$zU(b,c+1,z) = (c-b)U(b,c,z)+U(b-1,c,z)$$

with $b = m+a+1$, $c = a$ and $z = xt$ gives

$$U(m+a+1,a+1,xt) = \frac{-(m+1)}{xt}U(m+a+1,a,xt) + \frac{U(m+a,a,xt)}{xt}$$

and

$$\int_1^\infty e^{-xt}\psi(m+a,a,xt)dt = \frac{e^{-x}}{x}U(m+a,a,x) - (m+a)\left[-\frac{(m+1)}{x}w_{m+1} + \frac{w_m}{x}\right].$$

(Notice that for $m = 0$, $U(a+1,a+1,x)$ is obtained in terms of y_0 and y_1 from which C_M in the previous section is derived.)

Then

$$A_{m+1}w_{m+1} - c_m A_m w_m - d_m A_m x\left[\frac{e^{-x}}{x}U(m+a,a,x) + \frac{(m+a)(m+1)}{x}w_{m+1} - \frac{(m+a)}{x}w_m\right]$$
$$+ a_m A_{m-1}w_{m-1} = 0$$

and combining these terms, we have

$$[A_{m+1}-d_mA_m(m+a)(m+1)]w_{m+1}-[c_mA_m-d_mA_m(m+a)]w_m+a_mA_{m-1}w_{m-1} = d_mA_me^{-x}U(m+a,a,x)$$

The recurrence

$$A_{m+1} = A_m(m+a)(m+1)d_m$$

can be derived from the definition of A_{m+1} and

$$y_m = A_mU(m+a,a,x).$$

Then, with these two relations, we get

$$0 \cdot w_{m+1}-A_m[c_m-d_m(m+a)]w_m+a_mA_{m-1}w_{m-1} = d_my_me^{-x}.$$

This is a two-term, non-homogeneous recurrence. Let $v_m = A_mw_me^{-x}$ and notice that $v_0 = w_0e^{-x}$. Using the definition of c_m, d_m and a_m, we get

$$v_{m-1} = \frac{m+a/2}{m+a-1}\left[v_m + \frac{y_m}{m}\right] \quad \text{or} \quad v_m = \frac{m+a/2+1}{m+a}\left[v_{m+1} + \frac{y_{m+1}}{m+1}\right].$$

It is clear that the amplification factor $(m+a/2+1)/(m+a)$, for $a > 2$, is less than 1 on backward recurrence and the recurrence is numerically stable. The beauty of this formula is that the $\{\tilde{v}_m^M\}$ sequence can be calculated in the same loop with the $\{\tilde{y}_m^M\}$ sequence giving not only the starting value $E_a(x)$ but also the starting value $G_a(x)$ for the recurrences

$$\nu E_{\nu+1}(x)+xE_\nu(x) = e^{-x}, \qquad \nu G_\nu(x)+xG_\nu(x) = E_{\nu+1}(x).$$

There are two details to add to make the analysis complete. The $\{\tilde{y}_m^M\}$ sequence is generated by starting with an asymptotic (~) estimate of the ratio $\tilde{r}_M = \tilde{y}_{M+1}^M / \tilde{y}_M^M$. One can see that this determines not only the $\{\tilde{y}_m^M\}$ sequence up to a multiplicative constant but also the $\{\tilde{v}_m^M\}$ $(\tilde{v}_{M+1}^M = 0)$ sequence up to the same multiplicative constant because each $\tilde{v}_m^M$ is a linear combination of previously calculated $\tilde{y}_k^M$'s, $k \geq m$. Thus, it is not surprising that C_M normalizes both sequences and

$$G_a(x) \doteq C_M \tilde{v}_0^M .$$

The second item to be addressed is the convergence of the series for $\tilde{v}_0^M$ (generated by backward recurrence from index M). This process is more slowly convergent than that for $\tilde{y}_0^M$. To remedy this, we simply decrease the requested tolerance from TOL = REL to TOL = REL/PTOL, PTOL = 1.0×10^{10}, in the DG routines. This gets the required convergence for the worst case when x is close to $XCUT$ at the tightest tolerance REL = 0.5×10^{-15}. This increases the value of M from about 50 to 120. For larger x's, the values of M are much less. TABLE I shows a comparison of the M for $E_a(x)$ in DHEXINT and the M for $G_a(x)$ in DGHEXINT starting with $a = 2.5$, the smallest value a could have in these routines (recall $a = N_x = [x]+0.5$ and $x > 2$). This empirical procedure using PTOL seems to work well for other values of REL also.

TABLE I

Typical values of M for a relative error REL = 0.5×10^{-15} at index a:

DGHEXINT, DHEXINT		
$a = [x] + 0.5$		
x	$G_a(x)$	$E_a(x)$
2.0	117	46
3.0	84	34
4.0	67	27
5.0	57	23
6.0	50	21
7.0	45	19
8.0	41	17
9.0	38	16
10.0	36	15
20.0	25	11

DGEXINT, DEXINT		
$a = [x + 0.5]$		
x	$G_a(x)$	$E_a(x)$
2.0	117	46
3.0	84	34
4.0	67	28
5.0	57	24
6.0	50	22
7.0	45	20
8.0	41	19
9.0	38	18
10.0	36	17
20.0	11	12

Numerical Experiments

In order to verify the correctness of the code and assess an overall accuracy, each routine was compared with a highly accurate quadrature using DQUAD8 (DGAUS8). The integrands for the quadrature used DEXINT for the integer orders and DHEXINT for the half odd orders. A relative error tolerance of 0.5×10^{-14} was requested from DQUAD8. The results showed worst overall relative differences of $O(10^{-13})$ which always occurred in the interval $1.5 \le x \le 2$.

Explicit Expressions for v_0 and $\tilde{v}_0$

We solve the recurrence relation as a difference equation

$$v_m = \frac{m+a/2+1}{m+a}\left[v_{m+1} + \frac{y_{m+1}}{m+1}\right] \qquad \text{or} \qquad v_{m+1} = \frac{m+a}{m+a/2+1}v_m - \frac{y_{m+1}}{m+1}$$

for the explicit solution by variation of parameters. The homogeneous equation

$$h_{m+1} = \frac{m+a}{m+a/2+1} h_m \qquad \text{has a solution} \qquad h_m = C\frac{\Gamma(m+a)}{\Gamma(m+a/2+1)}$$

and we express v_m as $v_m = h_m V_m$ to get

$$h_{m+1}V_{m+1} = h_m \frac{m+a}{m+a/2+1} V_m - \frac{y_{m+1}}{m+1} \quad \text{or} \quad V_{m+1} - V_m = -\frac{y_{m+1}}{(m+1)h_{m+1}}.$$

Now we sum on m to get

$$V_{m+1} - V_0 = -\sum_{k=0}^{m} \frac{y_{k+1}}{(k+1)h_{k+1}}$$

and we use the relations $V_m = v_m / h_m$ and $v_m = A_m e^{-x} w_m$ to show that $V_m \to 0$ as $m \to \infty$. w_m is defined by

$$w_m = \int_1^\infty \frac{e^{-xt}}{t} U(m+a,a,xt)dt \quad , \quad m \geq 0$$

and the definition of the confluent hypergeometric function $U(m+a,a,xt)$ yields

$$U(m+a,a,xt) = \frac{1}{\Gamma(m+a)} \int_0^\infty e^{-xtu} u^{m+a-1} /(1+u)^{m+1} du .$$

Then,

$$\left|U(m+a,a,xt)\right| \leq \frac{1}{\Gamma(m+a)} \int_0^\infty e^{-xtu} u^{a-1} du = \frac{\Gamma(a)}{\Gamma(m+a)} \cdot \frac{1}{(xt)^a}$$

and

$$\left|w_m\right| \leq \frac{\Gamma(a)}{\Gamma(m+a)} \cdot \int_1^\infty \frac{e^{-xt}}{x^a t^{a+1}} dt = \frac{\Gamma(a)}{x^a \Gamma(m+a)} E_{a+1}(x).$$

We also have from the previous development, $v_m = A_m e^{-x} w_m$ and

$$\left|v_m\right| \leq \frac{\Gamma(m+1)\Gamma(m+a)\Gamma(1+a/2)}{\Gamma(a)\Gamma(m+a/2+1)} e^{-x} \frac{\Gamma(a)E_{a+1}(x)}{x^a\Gamma(m+a)} = \frac{\Gamma(m+1)}{\Gamma(m+a/2+1)} \frac{\Gamma(1+a/2)}{x^a} e^{-x} E_{a+1}(x) = O(m^{-a/2})$$

using $\Gamma(z+a)/\Gamma(z+b) \sim z^{a-b}$, $z \to \infty$. Then,

$$V_m = \frac{v_m}{h_m} \leq \frac{\Gamma(m+1)\Gamma(1+a/2)}{x^a \Gamma(m+a/2+1)} e^{-x} E_{a+1}(x) \cdot \frac{\Gamma(m+a/2+1)}{C\Gamma(m+a)} = \frac{\Gamma(m+1)}{C\Gamma(m+a)} \frac{\Gamma(1+a/2)}{x^a} e^{-x} E_{a+1}(x) = O(m^{-a+1})$$

Thus, $V_m \to 0$ as $m \to \infty$ for a>1. Therefore,

$$V_0 = \frac{1}{C} \sum_{k=0}^\infty \frac{y_{k+1}\Gamma(k+a/2+2)}{(k+1)\Gamma(k+a+1)}$$

and

$$v_0 = V_0 h_0 = \frac{\Gamma(a)}{\Gamma(1+a/2)} \sum_{k=0}^\infty \frac{y_{k+1}\Gamma(k+a/2+2)}{(k+1)\Gamma(k+a+1)}.$$

If we truncate or recur backward on the recurrence from m=M with $\tilde{v}_M^M = 0$, we get

$$\tilde{v}_0^M = \frac{\Gamma(a)}{\Gamma(1+a/2)} \sum_{k=0}^{M} \frac{\tilde{y}_{k+1}^M \Gamma(k+a/2+2)}{(k+1)\Gamma(k+a+1)} = \sum_{k=0}^{M} \frac{\tilde{y}_{k+1}^M (1+a/2)_{k+1}}{(k+1)(a)_{k+1}}$$

and we see that the normalization factor C_M for the sequence $\{\tilde{y}_k^M\}$ also normalizes $\tilde{v}_0^M$ to produce the the computational form for $G_a(x)$:

$$G_a(x) \doteq C_M \tilde{v}_0^M .$$

From the analysis above, we have an estimate for the truncation error of the series for

$$v_0 = V_0 h_0 = \frac{\Gamma(a)}{\Gamma(1+a/2)} \sum_{k=0}^{\infty} \frac{y_{k+1} \Gamma(k+a/2+2)}{(k+1)\Gamma(k+a+1)} = \sum_{k=0}^{\infty} \frac{y_{k+1} (1+a/2)_{k+1}}{(k+1)(a)_{k+1}}$$

as $O(M^{-a/2})$. But a new and possibly better estimate can be obtained by expressing y_{k+1} in terms of its hypergeometric function $U(a,b,x)$ and estimating the terms of the series by using the *uniform asymptotic expansion* of Olver, being asymptotic in a with an error which is uniformly bounded in x. This uniform expression is presented on page 80 in the classic reference

Slater, L.J., Confluent Hypergeometric Functions, Cambridge University Press, 1960.

We start by expressing the terms of the series in terms of $U(k+a,a,x)$, using

$$y_k = A_k U(k+a,a,x), \qquad A_k = \frac{(1)_k (a)_k}{(1+a/2)_k} .$$

Then

$$\frac{\Gamma(a)}{\Gamma(1+a/2)} \frac{y_{k+1}\Gamma(k+a/2+2)}{(k+1)\Gamma(k+a+1)} = \frac{(1)_{k+1}(a)_{k+1}}{(1+a/2)_{k+1}} \frac{U(k+a+1,a,x)}{k+1} \frac{(1+a/2)_{k+1}}{(a)_{k+1}} = \Gamma(k+1)U(k+a+1,a,x)$$

Now the first term of the uniform asymptotic expression for U as expressed by Slater is

$$U(u^2/4+b/2,b,z^2) \sim \frac{2^{2-b} u^{b-1} e^{z^2/2}}{z^b \Gamma(u^2/4+b/2)} \cdot z K_{b-1}(uz)$$

where $K_{b-1}(uz)$ is the modified Bessel function of the second kind. We take

$$u^2/4+b/2 = k+a+1 \qquad b=a \qquad z=\sqrt{x} \qquad u = 2\sqrt{k+a/2+1}$$

which gives

$$\Gamma(k+1)U(k+a+1,a,x) \sim \frac{\Gamma(k+1) 2^{2-a} 2^{a-1} (k+a/2+1)^{(a-1)/2}}{x^{a/2}\Gamma(k+a+1)} \cdot e^{x/2}\sqrt{x} K_{a-1}(2\sqrt{x(k+a/2+1)})$$

and using $\Gamma(z+a)/\Gamma(z+b) \sim z^{a-b}$, $z \to \infty$ we have

$$\Gamma(k+1)U(k+a+1,a,x) \sim \frac{2}{x^{(a-1)/2}} \frac{(k+a/2+1)^{(a-1)/2}}{k^a} \cdot e^{x/2}\sqrt{x} K_{a-1}(2\sqrt{x(k+a/2+1)}) .$$

Now, the argument of the K Bessel function is large for k large and we use the first term of the large argument expansion

$$K_\nu(z) \sim \sqrt{\frac{\pi}{2z}} e^{-z} \ , \qquad z \to \infty$$

to obtain

$$\Gamma(k+1)U(k+a+1,a,x) \approx \frac{2}{x^{(a-1)/2}}\left(1+\frac{1+a/2}{k}\right)^{a/2} \frac{e^{x/2}}{k^{a/2}} \sqrt{\frac{\pi}{4\sqrt{x(k+a/2+1)}}} \frac{e^{-2\sqrt{x(k+a/2+1)}}}{\sqrt{k+a/2+1}}$$

or

$$\Gamma(k+1)U(k+a+1,a,x) \approx \frac{\sqrt{\pi}}{x^{(2a-1)/4}}\left(1+\frac{1+a/2}{k}\right)^{a/2} \cdot e^{x/2} \frac{e^{-2\sqrt{x(k+a/2+1)}}}{(k+a/2+1)^{3/4} k^{a/2}}$$

A truncation error estimate for the series terminating at index M can be calculated from the integral

$$\frac{\sqrt{\pi}}{x^{(2a-1)/4}} e^{x/2} \int_M^\infty \left(1+\frac{1+a/2}{k}\right)^{a/2} \frac{e^{-2\sqrt{x(k+a/2+1)}}}{(k+a/2+1)^{3/4} k^{a/2}} dk \ .$$

Then, creating upper bounds by replacement of k with M, we have

$$\frac{\sqrt{\pi}}{x^{(2a-1)/4}} \frac{e^{x/2}}{M^{a/2}} \left(1+\frac{1+a/2}{M}\right)^{a/2} \int_M^\infty \frac{e^{-2\sqrt{x(k+a/2+1)}}}{(k+a/2+1)^{3/4}} dk$$

and

$$= \frac{\pi\sqrt{2}}{x^{a/2}} \frac{e^{x/2}}{M^{a/2}} \left(1+\frac{1+a/2}{M}\right)^{a/2} erfc\left(X^{1/4}\sqrt{2}\right) \approx \frac{\sqrt{\pi}\, e^{x/2}}{x^{a/2} M^{a/2}} \left(1+\frac{1+a/2}{M}\right)^{a/2} \frac{e^{-2\sqrt{X}}}{X^{1/4}}$$

where $X = x(M + a/2 + 1)$.

The worst case where M is largest occurs for x=2 and a=2 (see TABLE I). For M=117, this expression evaluates to $O(10^{-15})$ which is in very good agreement with the result from TABLE I.

<u>Computation of $G_1(x)$, $x \le 2$</u>

The code for integer orders $G_n(x)$, n = 1,2…is denoted by DGEXINT. In order to compute a sequence for $x \le 2$ we need to compute $G_1(x)$ to start the forward recurrence. We start the derivation by noting the Laplace transform [4],

$$\int_0^\infty e^{-st} E_\nu(t)dt = \sum_{n=0}^\infty \frac{(-s)^r}{r+\nu} \qquad\qquad 0 \le s \le 1, \qquad \nu > 0$$

where ν need not be an integer. Let $t = xw$. Then

$$x\int_0^\infty e^{-sxw} E_\nu(xw)dw = \sum_{r=0}^\infty \frac{(-s)^r}{r+\nu}.$$

Integrate s on $[0,s]$ to obtain

$$\int_0^\infty \frac{1-e^{-sxw}}{w} E_\nu(xw)dw = \sum_{r=0}^\infty \frac{(-1)^r s^{r+1}}{(r+\nu)(r+1)}.$$

Now, break the interval $[0,\infty]$ into $[0,1]$, $[1,\infty]$ and use the power series for $\exp(-sxw)$ on $[0,1]$:

$$-\sum_{r=1}^\infty \frac{(-x)^r s^r}{r!} \int_0^1 w^{r-1} E_\nu(xw)dw + \int_1^\infty \frac{E_\nu(xw)}{w} dw - \int_1^\infty \frac{e^{-sxw}}{w} E_\nu(xw)dw$$

$$= \sum_{r=0}^\infty \frac{(-1)^r s^{r+1}}{(r+\nu)(r+1)}$$

Notice that if $\nu = 1$ we have $G_1(x)$ on the left along with other terms. To get $G_2(x)$ we can integrate s on $[0,s]$ again to get w^2 in the denominator, etc.

For $\nu = 1$ and $s = 1$, we can compute $G_1(x)$. Now

$$E_1(xw) = -\gamma - \ln(xw) - \sum_{n=1}^\infty \frac{(-x)^n w^n}{nn!}$$

and for $r \geq 1$

$$\int_0^1 w^{r-1} E_1(xw)dw = -\frac{(\gamma + \ln x)}{r} - \int_0^1 w^{r-1} \ln w dw - \sum_{n=1}^\infty \frac{(-x)^n}{n(n+r)n!} \quad , \qquad r \geq 1$$

Also,

$$\int_0^1 w^{r-1} \ln w dw = \left[\frac{w^r \ln w}{r} - \frac{w^r}{r^2} \right]_0^1 = -\frac{1}{r^2} \quad , \qquad r \geq 1$$

and

$$\int_0^1 w^{r-1} E_1(xw)dw = -\frac{(\gamma + \ln x)}{r} + \frac{1}{r^2} - \sum_{n=1}^\infty \frac{(-x)^n}{n(n+r)n!}$$

Notice also that the right side for $\nu = 1$ and $s = 1$ is found in tables:

$$\sum_{r=0}^{\infty} \frac{(-1)^r}{(r+1)^2} = \frac{\pi^2}{12}$$

Then, for $\nu = 1$ and $s = 1$,

$$-\int_1^{\infty} \frac{e^{-xw}}{w} E_1(xw)dw = \left.\frac{E_1^2(xw)}{2}\right|_1^{\infty} = \frac{-E_1^2(x)}{2}$$

and summing on r gives

$$G_1(x) = \frac{\pi^2}{12} + \frac{1}{2}E_1^2(x) - (\gamma + \ln x)\sum_{r=1}^{\infty} \frac{(-x)^r}{rr!} + \sum_{r=1}^{\infty} \frac{(-x)^r}{r^2 r!} - S(x)$$

where

$$S(x) = \sum_{r=1}^{\infty} \frac{(-x)^r}{r!} \sum_{n=1}^{\infty} \frac{(-x)^n}{n(n+r)n!}$$

We sum the double series by taking left sloping diagonals in the (r,n) lattice on $[1,\infty)\times[1,\infty)$ with the substitution $n+r = k$

$$S(x) = \sum_{k=2}^{\infty} \frac{(-x)^k}{k} A_k\,, \qquad A_k = \sum_{r=1}^{k-1} \frac{1}{r!(k-r)!(k-r)}, \qquad k \ge 2$$

or

$$A_k = \frac{1}{k!}\sum_{r=1}^{k-1} C_r^k \cdot \frac{1}{k-r}, \qquad k \ge 2$$

Now, A_k is bounded by a partial binomial sum

$$A_k < \frac{1}{k!}\sum_{r=1}^{k-1} C_r^k = \frac{1}{k!}[2^k - 2] < \frac{2^k}{k!}$$

and this series converges as well as the other series in the representation of $G_1(x)$. The final form is

$$G_1(x) = \frac{\pi^2}{12} + \frac{1}{2}E_1^2(x) - (\gamma + \ln x)\sum_{r=1}^{\infty} \frac{(-x)^r}{rr!} + \sum_{r=1}^{\infty} \frac{(-x)^r}{r^2 r!} - \sum_{k=2}^{\infty} \frac{(-x)^k}{k} A_k$$

Computation of $G_2(x)$, $x \le 2$

We have three options for computing $G_2(x)$:

Option 1: The first option involves using one step of the recurrence starting with $\nu = 1$:

$$E_2(x) = e^{-x} - xE_1(x), \qquad G_2(x) = E_2(x) - xG_1(x)$$

where $G_1(x)$ is computed in the previous section.

Option 2: We integrate

$$G_\nu'(x) = -G_{\nu-1}(x)$$

for $\nu = 2$ to get

$$G_2(x) = G_2(0) - \int_0^x G_1(t)dt, \ \ G_\nu(0) = (\nu-1)^{-2}, \ \ G_2(0) = 1.$$

The required integrals using $G_1(x)$ from the previous section gives

$$\int_0^x E_1^2(t)dt = \int_0^x \left[(\gamma + \ln t)^2 + 2(\gamma + \ln t)\sum_{n=1}^{\infty} \frac{(-t)^n}{nn!} + \sum_{n=2}^{\infty} (-t)^n D_n\right]dt$$

$$\int_0^x (\gamma + \ln t)^2 dt = x(\gamma + \ln x)^2 - 2x(\gamma + \ln x) + 2x$$

$$\int_0^x t^n(\gamma + \ln t)dt = \frac{x^{n+1}}{n+1}[\gamma + \ln x] - \frac{x^{n+1}}{(n+1)^2}, \qquad n \ge 0$$

$$D_n = \sum_{k=1}^{n-1} \frac{1}{kk!(n-k)(n-k)!} = \frac{1}{n!}\sum_{k=1}^{n-1} \frac{C_k^n}{k(n-k)}, \qquad n \ge 2$$

Then,

$$\begin{aligned} G_2(x) = 1 - \frac{\pi^2}{12}x - \frac{1}{2}\Bigg[& x(\gamma + \ln x)^2 - 2x(\gamma + \ln x) + 2x + 2\sum_{n=1}^{\infty} \frac{(-1)^n}{nn!}\int_0^x t^n(\gamma + \ln t)dt \\ & + x\sum_{n=2}^{\infty} \frac{(-x)^n}{(n+1)} D_n\Bigg] + \sum_{n=1}^{\infty} \frac{(-1)^n}{nn!}\int_0^x t^n(\gamma + \ln t)dt \\ & - x\sum_{n=1}^{\infty} \frac{(-x)^n}{n^2 n!(n+1)} + x\sum_{n=2}^{\infty} \frac{(-x)^n}{n(n+1)} A_n \end{aligned}$$

and finally

$$G_2(x) = 1 - \frac{\pi^2}{12}x - \frac{x}{2}(\gamma + \ln x)^2 + x(\gamma + \ln x) - x - x\sum_{n=1}^{\infty} \frac{(-x)^n}{n^2(n+1)n!}$$

$$+ x\sum_{n=2}^{\infty} \frac{(-x)^n}{n(n+1)} A_n - \frac{x}{2}\sum_{n=2}^{\infty} \frac{(-x)^n}{(n+1)} D_n$$

where C_k^n is a binomial coefficient and

$$A_n = \frac{1}{n!}\sum_{k=1}^{n-1} \frac{C_k^n}{n-k}, \qquad D_n = \frac{1}{n!}\sum_{k=1}^{n-1} \frac{C_k^n}{k(n-k)}, \qquad n \geq 2.$$

<u>Option 3</u>: In this option, we continue to manipulate the general form used to compute $G_1(x)$ by setting $\nu = 2$:

$$-\sum_{r=1}^{\infty} \frac{(-x)^n}{r!} s^r \int_0^1 w^{r-1} E_2(xw)dw + \int_1^{\infty} \frac{E_2(xw)}{w} dw - \int_1^{\infty} \frac{e^{-sxw}}{w} E_2(xw)dw$$

$$= \sum_{r=0}^{\infty} \frac{(-1)^r s^{r+1}}{(r+2)(r+1)}$$

Notice that if we integrate s on $[0,s]$ we get

$$-\sum_{r=1}^{\infty} \frac{(-x)^r}{r!} \frac{s^{r+1}}{r+1} \int_0^1 w^{r-1} E_2(xw)dw + s\int_1^{\infty} \frac{E_2(xw)}{w} dw - \frac{1}{x}\int_1^{\infty} \frac{1 - e^{-sxw}}{w^2} E_2(xw)dw$$

$$= \sum_{r=0}^{\infty} \frac{(-1)^r s^{r+2}}{(r+1)(r+2)^2}$$

and $-G_2(x)/x$ appears as a term. To get computable quantities, we set $s = 1$, and use

$$E_2(x) = x[\gamma - 1 + \ln x] + 1 - \sum_{n=2}^{\infty} \frac{(-x)^n}{(n-1)n!}.$$

The integrals become

$$\int_0^1 w^{r-1} E_2(xw)dw = x(\gamma - 1 + \ln x)/(r+1) + x\int_0^1 w^r \ln w dw + \frac{1}{r} - \sum_{n=2}^{\infty} \frac{(-x)^n}{(n-1)(n+r)n!}$$

$$\int_1^{\infty} \frac{E_2(xw)}{w} dw = E_1(x) - E_2(x) \qquad \text{(Introduction, } \mu = 2, \nu = 1\text{)}$$

and with $E_2(x) = e^{-x} - xE_1(x)$,

$$\int_1^\infty \frac{e^{-xw}}{w^2}E_2(xw)dw = \int_1^\infty \frac{e^{-2xw}}{w^2}dw - x\int_1^\infty \frac{e^{-xw}}{w}E_1(xw)dw$$

$$= E_2(2x) + x\left.\frac{E_1^2(xw)}{2}\right|_1^\infty = E_2(2x) - \frac{x}{2}E_1^2(x)$$

The right side can be evaluated explicitly by writing

$$\sum_{r=0}^\infty \frac{(-1)^r}{(r+1)(r+2)^2} = \sum_{r=0}^\infty \frac{(-1)^r}{r+2}\left[\frac{1}{r+1} - \frac{1}{r+2}\right] = \sum_{r=0}^\infty (-1)^r\left[\frac{1}{r+1} - \frac{1}{r+2}\right] - \sum_{r=0}^\infty \frac{(-1)^r}{(r+2)^2}$$

and observing that

$$\ln(1+x) = x\sum_{k=0}^\infty \frac{(-1)^k x^k}{k+1}, \qquad \ln 2 = \sum_{k=0}^\infty \frac{(-1)^k}{k+1} = 1 - \sum_{r=0}^\infty \frac{(-1)^r}{(r+2)}, \qquad -1<x\le 1$$

and

$$\sum_{r=0}^\infty \frac{(-1)^r}{(r+2)^2} = \sum_{k=1}^\infty \frac{(-1)^{k-1}}{(k+1)^2} = 1 - \left(1 + \sum_{k=1}^\infty \frac{(-1)^k}{(k+1)^2}\right) = 1 - \frac{\pi^2}{12}$$

Then the right side is

$$\sum_{r=0}^\infty \frac{(-1)^r}{(r+1)(r+2)^2} = 2\ln 2 - 2 + \frac{\pi^2}{12}$$

Putting this all together, we have

$$G_2(x) = E_2(2x) - \frac{x}{2}E_1^2(x) + x[E_1(x) - E_2(x)] - x^2(\gamma - 1 + \ln x)\sum_{r=1}^\infty \frac{(-x)^r}{(r+1)^2 r!}$$

$$+ x^2\sum_{n=1}^\infty \frac{(-x)^r}{(r+1)^3 r!} - x\sum_{n=1}^\infty \frac{(-x)^r}{r(r+1)r!} + x\sum_{r=1}^\infty \frac{(-x)^r}{(r+1)r!}\sum_{n=2}^\infty \frac{(-x)^n}{(n-1)(n+r)n!} - x\left[2\ln 2 - 2 + \frac{\pi^2}{12}\right].$$

The double sum can be manipulated by letting $n+r = k$ and summing in the (r,n) lattice on $[1,\infty)\times[2,\infty)$ along left sloping diagonals:

$$\sum_{r=1}^\infty \frac{(-x)^r}{(r+1)r!}\sum_{n=2}^\infty \frac{(-x)^n}{(n-1)(n+r)n!} = \sum_{k=3}^\infty \frac{(-x)^k}{k}B_k$$

where

$$B_k = \sum_{r=1}^{k-2} \frac{1}{(k-r-1)(r+1)r!(k-r)!} = \frac{1}{k!}\sum_{r=1}^{k-2} \frac{C_r^k}{(k-r-1)(r+1)}$$

Notice that

$$B_k < \frac{2^k - 2 - k}{k!} < \frac{2^k}{k!}, \qquad\qquad k \ge 3$$

and the convergence is as good as the other terms in the series. Then, the final answer is

$$G_2(x) = E_2(2x) - \frac{x}{2}E_1^2(x) + x[E_1(x) - E_2(x)] - x^2(\gamma - 1 + \ln x)\sum_{n=1}^{\infty} \frac{(-x)^n}{(n+1)^2 n!}$$

$$+ x^2 \sum_{n=1}^{\infty} \frac{(-x)^n}{(n+1)^3 n!} - x\sum_{n=1}^{\infty} \frac{(-x)^n}{(n+1)nn!} + x\sum_{k=3}^{\infty} \frac{(-x)^k}{k} B_k - x\left[2\ln 2 - 2 + \frac{\pi^2}{12}\right]$$

where $E_1(x)$ and $E_2(x)$ are computed from their series. Computationally, all sums can be generated in one loop with modifications to the factor $(-x)^n/n!$

Numerical Results These three methods of generating $G_2(x)$ were tested against a quadrature of $E_2(xt)/t^2$ on $[1,\infty)$ with DQUAD8 (DGAUS8) using DEXINT for E_2 with a tolerance of 0.5×10^{-14}. OPTION 2 was slightly better than OPTIONS 1 or 3. The differences amounted to about one more digit of relative error, with an overall worst (relative) error being 0.66×10^{-13} in the interval $1.5 \le x \le 2$.

Folder 18b

Application of $G_\nu(x)$ to

$$\mathbf{I_n(b,T) = \int_T^\infty \frac{e^{-b^2x^2}\ln x}{x^n}dx, \quad \boldsymbol{n} = 0, 1, 2, \ldots}$$

Explicit formula Change the variable $x = T\sqrt{v}$ to get

$$I_n(b,T) = \frac{1}{2T^{n-1}}\int_1^\infty \frac{e^{-b^2T^2v}}{v^{(n+1)/2}}\ln(T\sqrt{v})dv$$

$$= \frac{\ln T}{2T^{n-1}}E_{(n+1)/2}(b^2T^2) + \frac{1}{4T^{n-1}}\int_1^\infty \frac{e^{-b^2T^2v}}{v^{(n+1)/2}}\ln v dv$$

$$= \frac{\ln T}{2T^{n-1}}E_{(n+1)/2}(b^2T^2) + \frac{1}{4T^{n-1}}G_{(n+1)/2}(b^2T^2)$$

For $n = 0$ the results are computed in Folder 16:

$$I_0(b,T) = \int_T^\infty e^{-b^2x^2}\ln x dx = \frac{\sqrt{\pi}}{2b}[\operatorname{erfc}(bT)\ln T + G(bT)]$$

For even integers we get the G_ν function of half odd orders:

$$I_{2n}(b,T) = \frac{\ln T}{2T^{2n-1}}E_{n+1/2}(b^2T^2) + \frac{1}{4T^{2n-1}}G_{n+1/2}(b^2T^2), \quad G_{1/2}(x) = 2\sqrt{\frac{\pi}{x}}G(\sqrt{x}) \quad , \qquad n \geq 0$$

For odd integers, we get the G_ν functions of integer orders,

$$I_{2n+1}(b,T) = \frac{\ln T}{2T^{2n}}E_{n+1}(b^2T^2) + \frac{1}{4T^{2n}}G_{n+1}(b^2T^2), \qquad n \geq 0$$

<u>Recurrence for $I_n(b,T)$</u> Integrate I_n by parts:

$$I_n = \int_T^\infty x\frac{e^{-b^2x^2}}{x^{n+1}}\ln x dx \ ,$$

$$u = \frac{\ln x}{x^{n+1}} \qquad\qquad dv = xe^{-b^2x^2}dx$$

$$du = \left[\frac{1}{x^{n+2}} - \frac{(n+1)}{x^{n+2}}\ln x\right]dx \qquad\qquad v = -\frac{1}{2b^2}e^{-b^2x^2}$$

Then,

$$I_n(b,T) = \frac{\ln T}{2b^2} \cdot \frac{e^{-b^2T^2}}{T^{n+1}} + \frac{1}{2b^2}\int_T^\infty \frac{e^{-b^2x^2}}{x^{n+2}}dx - \frac{(n+1)}{2b^2}\int_T^\infty \frac{e^{-b^2x^2}}{x^{n+2}}\ln x dx$$

$$= \frac{\ln T}{2b^2} \cdot \frac{e^{-b^2T^2}}{T^{n+1}} + \frac{1}{4b^2T^{n+1}}\int_1^\infty \frac{e^{-b^2T^2v}}{v^{(n+3)/2}}dv - \frac{(n+1)}{2b^2}I_{n+2}(b,T)$$

or

$$\frac{(n+1)}{2b^2}I_{n+2} + I_n = \frac{\ln T}{2b^2} \cdot \frac{e^{-b^2T^2}}{T^{n+1}} + \frac{1}{4b^2T^{n+1}}E_{(n+3)/2}(b^2T^2)$$

and this shows that the even and odd sequences are decoupled. Also, the amplification factors

$$\frac{2b^2}{n+1} \text{ or } \frac{n+1}{2b^2}$$

for forward or backward recurrence show that a stable recurrence occurs by recursion away from $N_b = [2b^2]$ or N_b+1 since one of these is an even index and one is odd. Thus, we choose one of these indices for an odd or even sequence and choose Case I, II or III in the section on General Description in Folder 18a. Then, with $N_x = N_b$ or N_b+1 we compute $I_a(b,T)$ from the explicit formula to start the recurrence where a is defined in one of the Cases I, II or III depending on the relation of a to N_x. Notice also that for T=1,

$$I_n(b,1) = \frac{1}{4}G_{(n+1)/2}(b^2) = -\frac{1}{4}\left[\frac{\partial E_\nu(b^2)}{\partial \nu}\right]_{\nu=(n+1)/2}.$$

References

[1] Amos, D.E., "Computation of Exponential Integrals", *ACM Trans. Math Software*, **6**, No. 3, September 1980, pp. 365-377.

[2] Amos, D.E., "Algorithm 556, Exponential Integrals", *ACM Trans. Math Software*, **6**, No. 3, September 1980, pp. 420-428.

[3] Abramowitz, M. and Stegun, I.A., Handbook of Mathematical Functions *(AMS 55)*, National Bureau of Standards, Washington, D.C., Dec. 1965.

[4] Miller, G.F., "Tables of Generalized Exponential Integrals", *Mathematical Tables*, **3**, National Physical Laboratory, Her Majesty's Stationery Office, 1960.

Folder 19

Evaluation of

$$\mathbf{I_{19}(a,b,T)=\int_0^T \frac{erf(aw)erf(bw)}{w}dw=\frac{1}{2}\int_t^\infty \frac{erf(a/\sqrt{\tau})erf(b/\sqrt{\tau})}{\tau}d\tau}$$

$$\mathbf{I_{19}^c(a,b,T)=\int_T^\infty \frac{erfc(aw)\,erfc(bw)}{w}dw=\frac{1}{2}\int_0^t \frac{erfc(a/\sqrt{\tau})erfc(b/\sqrt{\tau})}{\tau}d\tau}$$

$$\boldsymbol{a>0,\qquad b>0,\qquad T=1/\sqrt{t},\qquad t>0}$$

Donald E. Amos, April 2002

Summary

A computational form for I_{19}^{c} for $a \le b$ is derived. The case for $a > b$ is obtained by exchanging a and b since I_{19}^{c} is symmetric in a and b. The result is

$$\begin{aligned}\mathrm{I}_{19}^{c}(a,b,T) &= G(bT)\,\mathrm{erfc}(aT) - \frac{a}{2\pi\sqrt{a^2+b^2}}\sum_{k=0}^{\infty} C_k\left(\frac{a^2}{a^2+b^2}\right)^k G_{k+3/2}(X)\\ &+\frac{a}{\sqrt{\pi}}\ln\left[\frac{b^2}{a^2+b^2}\right] I_5(a,b,T) \qquad\qquad a\le b\\ &+\frac{a}{2\pi\sqrt{a^2+b^2}}\sum_{k=1}^{\infty}\left(\frac{a^2}{a^2+b^2}\right)^k E_{k+3/2}(X)\sum_{m=1}^{k}\frac{C_{k-m}}{m}, \qquad X=T^2(a^2+b^2),\end{aligned}$$

where $G(X)$ is computed from DGERFC (Folder 16), $I_5(a,b,T)$ is computed from INTEGI5 (Folder 5), $C_k = (1/2)_k/k!$, $k \ge 0$, and $G_{k+3/2}(X)$ and $E_{k+3/2}(X)$ are generated from DGHEXINT (Folder 18). The convergence of the series is rapid since $a^2/(a^2+b^2)$ is at most ½, and (Folder 18)

$$E_{k+3/2}(X)\le\frac{e^{-X}}{(k+1/2+X)},\quad G_{k+3/2}(X)\le\frac{e^{-X}}{(k+1/2+X)^2},\quad C_k\sim\frac{1}{\sqrt{\pi k}},\quad k\to\infty$$

$$X=T^2(a^2+b^2).$$

For $\mathrm{I}_{19}(a,b,T)$ we get

$$\mathrm{I}_{19}(a,b,T)=F(aT)-[G(bT)-\mathrm{I}_{19}^{c}(a,b,T)]_{T\to 0}+[G(bT)-\mathrm{I}_{19}^{c}(a,b,T)],\qquad a\le b$$

$$\mathrm{I}_{19}(a,b,T)=F(bT)-[G(aT)-\mathrm{I}_{19}^{c}(b,a,T)]_{T\to 0}+[G(aT)-\mathrm{I}_{19}^{c}(b,a,T)],\qquad a> b$$

and the limit for $T\to 0$ is computable from I_{19}^{c} above.

Analysis For I_{19}^c

Since I_{19}^c is symmetric in *a* and *b*, we consider the case where $a \le b$ and exchange the variables in the final formula for the case a>b.

Integrate by parts:

$$u = \operatorname{erfc}(ax) \qquad\qquad dv = \frac{\operatorname{erfc}(bx)}{x}dx$$

$$du = -\frac{2a}{\sqrt{\pi}}e^{-a^2x^2}dx \qquad\qquad v = -G(bx) = -\int_1^\infty \frac{\operatorname{erfc}(bxw)}{w}dw$$

where $G(x)$ is defined in Folder 6 and analyzed in Folder 16 for a computational code DGERFC. Then

$$I_{19}^c(a,b,T) = G(bT)\operatorname{erfc}(aT) - \frac{2a}{\sqrt{\pi}}\int_T^\infty e^{-a^2x^2}dx\int_1^\infty \frac{\operatorname{erfc}(bxw)}{w}dw$$

$$= G(bT)\operatorname{erfc}(aT) - \frac{2a}{\sqrt{\pi}}\int_1^\infty \frac{I_5(a,bw,T)}{w}dw$$

where

$$I_5(a,b,T) = \frac{1}{2\sqrt{\pi}}\sum_{k=0}^\infty C_k \frac{a^{2k}}{(a^2+b^2)^{k+1/2}}E_{k+3/2}(X), \qquad a \le b,$$

is the *I* function of Folder 5 for $a \le b$ with $X = T^2(a^2+b^2)$ and $C_k = (1/2)_k/k!$

Then,

$$I_{19}^c(a,b,T) = G(bT)\operatorname{erfc}(aT) - \frac{a}{\pi}\sum_{k=0}^\infty C_k a^{2k}\int_1^\infty \frac{E_{k+3/2}[T^2(a^2+b^2w^2)]}{w[a^2+b^2w^2]^{k+1/2}}dw$$

and $a \le b \le bw$ since $w \ge 1$. Change variables, $\nu = T^2(a^2+b^2w^2)$ or $w = (\sqrt{\nu - a^2T^2})/bT$. Then

$$U_k \equiv \int_1^\infty \frac{E_{k+3/2}[T^2(a^2+b^2w^2)]}{w[a^2+b^2w^2]^{k+1/2}}dw = \frac{T^{2k+1}}{2}\int_X^\infty \frac{E_{k+3/2}(\nu)}{\nu^{k+1/2}(\nu - a^2T^2)}d\nu$$

$$= \frac{T^{2k+1}}{2}\sum_{m=0}^\infty (a^2T^2)^m \int_X^\infty \frac{E_{k+3/2}(\nu)}{\nu^{k+3/2+m}}d\nu, \qquad X = T^2(a^2+b^2).$$

Now, from Folder 18

$$\int_X^\infty \frac{E_\mu(t)}{t^\nu}dt = \frac{1}{X^{\nu-1}}\frac{[E_\nu(X)-E_\mu(X)]}{\mu-\nu} \qquad\qquad \mu\neq\nu$$

$$\int_X^\infty \frac{E_\nu(t)}{t^\nu}dt = \frac{1}{X^{\nu-1}}G_\nu(X) \qquad\qquad \mu=\nu$$

and

$$U_k = \frac{1}{2}\left[T^{2k+1}\int_X^\infty \frac{E_{k+3/2}(v)}{v^{k+3/2}}dv + \sum_{m=1}^\infty \frac{(a^2)^m}{(a^2+b^2)^{k+1/2+m}}\frac{E_{k+3/2+m}(X)-E_{k+3/2}(X)}{(-m)}\right]$$

$$= \frac{1}{2}\frac{G_{k+3/2}(X)}{(a^2+b^2)^{k+1/2}} + \frac{1}{2}\frac{E_{k+3/2}(X)}{(a^2+b^2)^{k+1/2}}\sum_{m=1}^\infty \frac{1}{m}\left(\frac{a^2}{a^2+b^2}\right)^m$$

$$-\frac{1/2}{(a^2+b^2)^{k+1/2}}\sum_{m=1}^\infty \frac{1}{m}\left(\frac{a^2}{a^2+b^2}\right)^m E_{k+3/2+m}(X)$$

Now,

$$-\ln(1-x) = \sum_{m=1}^\infty \frac{x^m}{m} \qquad\qquad |x|<1$$

and

$$U_k = \frac{1}{2\sqrt{a^2+b^2}}\left\{\frac{G_{k+3/2}(X)}{(a^2+b^2)^k} - \frac{\ln[b^2/(a^2+b^2)]}{(a^2+b^2)^k}E_{k+3/2}(X)\right.$$

$$\left.-\frac{1}{(a^2+b^2)^k}\sum_{m=1}^\infty \frac{1}{m}\left(\frac{a^2}{a^2+b^2}\right)^m E_{k+3/2+m}(X)\right\}$$

The sum on *k* yields

$$\mathrm{I}_{19}^{c}(\mathrm{a},\mathrm{b},\mathrm{T}) = \mathrm{G}(\mathrm{bT})\,\mathrm{erfc}(\mathrm{aT}) - \frac{\mathrm{a}}{2\pi\sqrt{\mathrm{a}^2+\mathrm{b}^2}}\sum_{\mathrm{k}=0}^\infty \mathrm{C}_\mathrm{k}\left(\frac{\mathrm{a}^2}{\mathrm{a}^2+\mathrm{b}^2}\right)^\mathrm{k}\mathrm{G}_{\mathrm{k}+3/2}(\mathrm{X})$$

$$+\frac{a}{2\pi}\frac{\ln[b^2/(a^2+b^2)]}{\sqrt{a^2+b^2}}\sum_{k=0}^\infty C_k\left(\frac{a^2}{a^2+b^2}\right)^k E_{k+3/2}(X)$$

$$+\frac{a}{2\pi\sqrt{a^2+b^2}}\sum_{k=0}^\infty C_k\left(\frac{a^2}{a^2+b^2}\right)^k\sum_{m=1}^\infty \frac{1}{m}\left(\frac{a^2}{a^2+b^2}\right)^m E_{m+k+3/2}(X)$$

where G(x) is computed in Folder 16 and $G_{k+3/2}(x)$ is computed in Folder 18. Notice that (from the previous page)

$$\frac{1}{\sqrt{a^2+b^2}}\sum_{k=0}^\infty C_k\left(\frac{a^2}{a^2+b^2}\right)^k E_{k+3/2}(X) = 2\sqrt{\pi}I_5(a,b,T).$$

Further, for the double sum, let $m + k = n$ and sum in the (m,k) lattice along left sloping diagonals. Then

$$\sum_{k=0}^{\infty} C_k\left(\frac{a^2}{a^2+b^2}\right)^k \sum_{m=1}^{\infty} \frac{1}{m}\left(\frac{a^2}{a^2+b^2}\right)^m E_{m+k+3/2}(X) = \sum_{n=1}^{\infty} \left(\frac{a^2}{a^2+b^2}\right)^n E_{n+3/2}(X) \sum_{m=1}^{n} \frac{C_{n-m}}{m}$$

Finally with $X = T^2(a^2+b^2)$, $C_k = (1/2)_k/k!$, $k \geq 0$,

$$\mathrm{I}_{19}^{c}(a,b,T) = G(bT)\,\mathrm{erfc}(aT) - \frac{a}{2\pi\sqrt{a^2+b^2}} \sum_{k=0}^{\infty} C_k\left(\frac{a^2}{a^2+b^2}\right)^k G_{k+3/2}(X)$$
$$+\frac{a}{\sqrt{\pi}} \ln\left[\frac{b^2}{a^2+b^2}\right] I_5(a,b,T) \qquad a \leq b$$
$$+\frac{a}{2\pi\sqrt{a^2+b^2}} \sum_{n=1}^{\infty} \left(\frac{a^2}{a^2+b^2}\right)^n E_{n+3/2}(X) \sum_{m=1}^{n} \frac{C_{n-m}}{m}$$

Notice that $E_{n+3/2}(X)$ and $G_{k+3/2}(X)$ sequences can be generated from DGHEXINT in the same CALL. Notice also that $a^2/(a^2+b^2)$ is at most ½ and the series converges rapidly since (Folder 18)

$$G_{k+3/2}(X) \leq \frac{e^{-X}}{(k+1/2+X)^2}, \quad E_{n+3/2}(X) \leq \frac{e^{-X}}{(n+1/2+X)} \quad \text{and} \quad C_k \sim \frac{1}{\sqrt{\pi k}} \quad \text{for } k \to \infty.$$

$G(bT)$ is computed from DGERFC (Folder 16). For errors $O(10^{-16})$ we need no more than 50 terms.

Numerical experiments show that I_{19}^{c} evaluated in the form

$$\mathrm{I}_{19}^{c}(a,b,T) = G(bT)\,\mathrm{erfc}(aT) + \frac{a}{\sqrt{\pi}} \ln\left[\frac{b^2}{a^2+b^2}\right] \mathrm{I}_5(a,b,T)$$
$$+\frac{a}{2\pi\sqrt{a^2+b^2}}\left\{-G_{3/2}(X) + \sum_{k=1}^{\infty}\left(\frac{a^2}{a^2+b^2}\right)^k \left[E_{k+3/2}(X)\sum_{m=1}^{k}\frac{C_{k-m}}{m} - C_k G_{k+3/2}(X)\right]\right\}$$

gives slightly better results presumably because powers $[a^2/(a^2+b^2)]^k$ reduce the magnitude of terms where losses of significance might occur.

<u>Analysis For</u> $\mathbf{I_{19}}$

We start with

$$\mathrm{I}_{19}(a,b,T) = \int_0^T \frac{\mathrm{erf}(aw)\,\mathrm{erf}(bw)}{w}\,dw = \int_0^T \frac{\mathrm{erf}(aw)}{w}\,dw - \int_0^T \frac{\mathrm{erf}(aw)\,\mathrm{erfc}(bw)}{w}\,dw$$
$$= F(aT) - \left[\int_0^{\infty} \frac{erf(aw)\,erfc(bw)}{w}\,dw - \int_T^{\infty} \frac{erf(aw)\,erfc(bw)}{w}\,dw\right]$$

where F(aT) can be computed from DFERF (Folder 16). Now,

$$\int_T^\infty \frac{\mathrm{erf}(aw)\mathrm{erfc}(bw)}{w}dw = \int_T^\infty \frac{\mathrm{erfc}(bw)}{w}dw - \int_T^\infty \frac{\mathrm{erfc}(aw)\mathrm{erfc}(bw)}{w}dw$$
$$= G(bT) - I_{19}^c(a,b,T)$$

and we get

$$I_{19}(a,b,T) = F(aT) - \left[\ G(bT) - I_{19}^c(a,b,T)\ \right]_{T\to 0} + \left[\ G(bT) - I_{19}^c(a,b,T)\ \right] \qquad a \le b$$

since the series for I_{19}^c converges best for $a \le b$. Then exchange a and b for a>b since I_{19} is symmetric in a and b,

$$I_{19}(a,b,T) = F(bT) - \left[\ G(aT) - I_{19}^c(b,a,T)\ \right]_{T\to 0} + \left[\ G(aT) - I_{19}^c(b,a,T)\ \right] \qquad a>b.$$

and the series for $I_{19}^c(b,a,T)$ converges best for a>b. Now we compute the limit to get the cancellation of the logarithmic contributions from G and I_{19}^c. Now,

$$G(bT) - I_{19}^c(a,b,T) = G(bT) - G(bT)erfc(aT) - R(a,b,T) = G(bT)erf(aT) - R(a,b,T)$$

where R is the regular part of I_{19}^c,

$$R(a,b,T) = \frac{a}{\sqrt{\pi}}\ln\left[\frac{b^2}{a^2+b^2}\right]I_5(a,b,T) - \frac{a}{2\pi\sqrt{a^2+b^2}}\sum_{k=0}^{\infty} C_k\left(\frac{a^2}{a^2+b^2}\right)^k G_{k+3/2}(X) \qquad a \le b$$

$$+\frac{a}{2\pi\sqrt{a^2+b^2}}\sum_{n=1}^{\infty}\left(\frac{a^2}{a^2+b^2}\right)^n E_{n+3/2}(X)\sum_{m=1}^{n}\frac{C_{n-m}}{m}\ , \qquad X = T^2(a^2+b^2)\,,$$

and $C_k = (1/2)_k/k!$, $G_{k+3/2}(0) = \dfrac{1}{(k+1/2)^2}$, $E_{n+3/2}(0) = \dfrac{1}{(n+1/2)}$, $I_5(a,b,0) = \dfrac{1}{a\sqrt{\pi}}\tan^{-1}\dfrac{a}{b}$.

Then the difference at T=0 is

$$\left[\ G(bT) - I_{19}^c(a,b,T)\ \right]_{T\to 0} = -R(a,b,0)\,, \qquad a \le b$$

since from Folder16,

$$G(bT)erf(aT) = erf(aT)\left[\ F(bT) - \frac{\gamma}{2} - \ln(2bT)\ \right] \to 0 \quad as \quad T \to 0\,.$$

For a>b we simply exchange a and b in the formulas since both I_{19} and I_{19}^c are symmetric in a and b.

Folder 20

Evaluation of

$$\mathbf{I_{20}(a,b,T) = \int_T^\infty e^{-a^2x^2}\,erf(bx)\ln x\,dx, \qquad I_{20}^c(a,b,T) = \int_T^\infty e^{-a^2x^2}\,erfc(bx)\ln x\,dx,}$$

$$\boldsymbol{a > 0,\quad b > 0,\quad T \geq 0}$$

Donald E. Amos, April 2002

Summary

The analysis for I_{20} and I_{20}^c is divided into Case I and Case II. Case I presents formulas for $a \leq b$. Case II presents formulas for $a > b$ which, in some cases, reduce the computation to Case I. The results are:

Case I, $a \leq b$

For I_{20}, we get

$$I_{20}(a,b,T) = \frac{\sqrt{\pi}}{2a}[\mathrm{erfc}(aT)\ln T + G(aT)] - I_{20}^c(a,b,T)$$

$$\mathrm{I}_{20}(\mathrm{a,b,T}) = \mathrm{J}_5(\mathrm{a,b,T})\ln\mathrm{T} + \frac{\sqrt{\pi}}{2\mathrm{a}}\mathrm{G(aT)} - \frac{1}{2\mathrm{a}\sqrt{\pi}}\mathrm{E}_1(\mathrm{X})\tan^{-1}\left(\frac{\mathrm{a}}{\mathrm{b}}\right) + \frac{\mathrm{S(a,b,T)}}{4\sqrt{\pi}\sqrt{\mathrm{a}^2+\mathrm{b}^2}}$$

$$I_{20}^c(a,b,T) = I_5(a,b,T)\ln T + \frac{1}{2a\sqrt{\pi}}E_1(X)\tan^{-1}\left(\frac{a}{b}\right) - \frac{S(a,b,T)}{4\sqrt{\pi}\sqrt{a^2+b^2}} \qquad a \leq b$$

where $I_5(a,b,T)$ and $J_5(a,b,T)$ are the I and J functions of Folder 5 (INTEGI5, and INTEGJ5) and

$$S(a,b,T) = \sum_{k=0}^{\infty} \frac{C_k}{(k+1/2)}\left(\frac{a^2}{a^2+b^2}\right)^k E_{k+3/2}(X)$$

$$X = T^2(a^2+b^2), \qquad C_k = (1/2)_k/k!, \qquad k \geq 0.$$

Notice that $a^2/(a^2+b^2) \leq ½$ and the convergence of the S series is better than $O(2^{-k}k^{-3/2})$.

Case II, $a > b$ The reflexive relation

$$I_{19}^c(a,b,T) = -\ln T\,\mathrm{erfc}(aT)\,\mathrm{erfc}(bT) + \frac{2a}{\sqrt{\pi}}I_{20}^c(a,b,T) + \frac{2b}{\sqrt{\pi}}I_{20}^c(b,a,T)$$

is derived from I_{19}^c of Folder 19 and used to compute for $a > b$:

$$I_{20}^{c}(a,b,T)=\frac{\sqrt{\pi}}{2a}[I_{19}^{c}(a,b,T)+\mathrm{erfc}(aT)\,\mathrm{erfc}(bT)\ln T]-\frac{b}{a}I_{20}^{c}(b,a,T) \qquad a>b$$

where $\mathrm{I}_{20}^{\mathrm{c}}(\mathrm{b,a,T})$ fits into Case I because the first parameter is smaller than the second parameter. $\mathrm{I}_{19}^{\mathrm{c}}(\mathrm{a,b,T})$ is computed with subroutine INTEGI19. We also derive the results for $\mathrm{I}_{20}^{\mathrm{c}}(\mathrm{a,b,0})$ and $\mathrm{I}_{20}(\mathrm{a,b,0})$ to make the result

$$I_{20}(a,b,T)=I_{20}(a,b,0)-\frac{\sqrt{\pi}}{2a}[\mathrm{erf}(aT)\mathrm{erf}(bT)\ln T-I_{19}(a,b,T)]+\frac{b}{a}\left[I_{20}(b,a,0)-I_{20}(b,a,T)\right]$$

useful for a>b. The special cases for a=0 and b=0 are also evaluated.

<u>Analysis for $\mathbf{I}_{20}^{\mathbf{c}}$</u>

<u>Case I, $a \le b$</u>. We integrate I_{20}^{c} by parts:

$$u = \ln x \qquad dv = e^{-a^2x^2}\,\mathrm{erfc}(bx)\,dx$$

$$du = dx/x \qquad v = -I_5(a,b,x)$$

then

$$\mathrm{I}_{20}^{\mathrm{c}}(\mathrm{a,b,T})=\mathrm{I}_5(\mathrm{a,b,T})\ln\mathrm{T}+\int_{\mathrm{T}}^{\infty}\frac{\mathrm{I}_5(\mathrm{a,b,x})}{\mathrm{x}}\mathrm{dx}$$

where $I_5(a,b,T)$ is the I function of Folder 5. For $a \le b$, we have (from Folder 5),

$$I_5(a,b,x)=\frac{1}{2\sqrt{\pi}\sqrt{a^2+b^2}}\sum_{k=0}^{\infty}C_k\left(\frac{a^2}{a^2+b^2}\right)^k E_{k+3/2}[x^2(a^2+b^2)], \qquad a\le b$$

$$C_k = (1/2)_k/k! \qquad k\ge 0.$$

Then,

$$\mathrm{I}_{20}^{\mathrm{c}}(\mathrm{a,b,T})=\mathrm{I}_5(\mathrm{a,b,T})\ln\mathrm{T}+\frac{1}{2\sqrt{\pi}\sqrt{\mathrm{a}^2+\mathrm{b}^2}}\sum_{\mathrm{k}=0}^{\infty}\mathrm{C}_{\mathrm{k}}\left(\frac{\mathrm{a}^2}{\mathrm{a}^2+\mathrm{b}^2}\right)^{\mathrm{k}}\mathrm{V}_{\mathrm{k}}$$

where

$$V_k=\int_T^{\infty}\frac{E_{k+3/2}[x^2(a^2+b^2)]}{x}dx.$$

Now, change variables $v = x^2(a^2+b^2)$ or $x=\sqrt{v}/\sqrt{a^2+b^2}$ to get

$$V_k = \frac{1}{2}\int_X^\infty \frac{E_{k+3/2}(v)}{v}\,dv, \qquad X = T^2(a^2+b^2).$$

From Folder 18, we have

$$\int_X^\infty \frac{E_\mu(t)dt}{t^\nu} = \frac{1}{X^{\nu-1}}\frac{E_\nu(X)-E_\mu(X)}{\mu-\nu} \qquad \mu \neq \nu$$

which leads to

$$V_k = \frac{1}{2}\frac{E_1(X)-E_{k+3/2}(X)}{(k+1/2)}, \qquad k \geq 0.$$

Then,

$$\mathrm{I}_{20}^{c}(\mathrm{a,b,T}) = \mathrm{I}_5(\mathrm{a,b,T})\ln\mathrm{T} + \frac{\mathrm{E}_1(\mathrm{X})}{4\sqrt{\pi}\sqrt{\mathrm{a}^2+\mathrm{b}^2}}\sum_{\mathrm{k}=0}^{\infty}\frac{\mathrm{C_k}}{\mathrm{k}+1/2}\left(\frac{\mathrm{a}^2}{\mathrm{a}^2+\mathrm{b}^2}\right)^{\mathrm{k}} - \frac{1}{4\sqrt{\pi}\sqrt{a^2+b^2}}\cdot S(a,b,T)$$

where

$$S(a,b,T) = \sum_{k=0}^{\infty}\frac{C_k}{k+1/2}\left(\frac{a^2}{a^2+b^2}\right)^k E_{k+3/2}(X), \qquad X = T^2(a^2+b^2)$$

$$C_k = (1/2)_k/k!$$

Now, we have from Folder 6,

$$\sum_{k=0}^{\infty}\frac{C_k}{k+1/2}x^k = \frac{2\sin^{-1}\sqrt{x}}{\sqrt{x}} = \frac{2}{\sqrt{x}}\tan^{-1}\sqrt{\frac{x}{1-x}}, \qquad |x| \leq 1$$

and

$$\mathrm{I}_{20}^{c}(\mathrm{a,b,T}) = \mathrm{I}_5(\mathrm{a,b,T})\ln\mathrm{T} + \frac{\mathrm{E}_1(\mathrm{X})}{2\mathrm{a}\sqrt{\pi}}\tan^{-1}\left(\frac{\mathrm{a}}{\mathrm{b}}\right) - \frac{\mathrm{S(a,b,T)}}{4\sqrt{\pi}\sqrt{\mathrm{a}^2+\mathrm{b}^2}}, \qquad a \leq b$$

Notice that $a^2/(a^2+b^2) \leq 1/2$ and the convergence of the S series is better than $O(2^{-k}k^{-3/2})$. Thus, for a truncation error $O(10^{-16})$ we can expect to use no more than 50 terms.

Case II, $a > b$. For this case, we note that if we integrate by parts,

$$\mathrm{I}_{19}^{c}(\mathrm{a,b,T}) = \int_{\mathrm{T}}^{\infty}\frac{\mathrm{erfc(aw)erfc(bw)}}{\mathrm{w}}\mathrm{dw},$$

we get a reflexive relation. Let

$u = \mathrm{erfc}(aw)\mathrm{erfc}(bw)$ $\qquad$ $dv = dw/w$

$$du = \left[-\frac{2a}{\sqrt{\pi}}e^{-a^2w^2}\ \mathrm{erfc}(bw) - \frac{2b}{\sqrt{\pi}}e^{-b^2w^2}\ \mathrm{erfc}(aw)\right]dw \qquad v = \ln w$$

Then,

$$\mathrm{I}_{19}^{c}(a,b,T) = -\,\mathrm{erfc}(aT)\mathrm{erfc}(bT)\ln T + \frac{2a}{\sqrt{\pi}}\mathrm{I}_{20}^{c}(a,b,T) + \frac{2b}{\sqrt{\pi}}\mathrm{I}_{20}^{c}(b,a,T)$$

or

$$\mathrm{I}_{20}^{c}(a,b,T) = \frac{\sqrt{\pi}}{2a}[\mathrm{I}_{19}^{c}(a,b,T) + \mathrm{erfc}(aT)\mathrm{erfc}(bT)\ln T] - \frac{b}{a}\mathrm{I}_{20}^{c}(b,a,T).$$

Now, I_{20}^{c} (b,a,T) fits into Case I because the first parameter is smaller than the second parameter. I_{19}^{c} (a,b,T) is computed from subroutine INTEGI19.

Analysis for $\mathbf{I}_{20}$

Since erf(x)=1-erfc(x) we can write from Folder 16,

$$\mathrm{I}_{20}(a,b,T) = \int_T^\infty e^{-a^2x^2}\ \ln x\,dx - \mathrm{I}_{20}^{c}(a,b,T) = \frac{\sqrt{\pi}}{2a}[\mathrm{erfc}(aT)\ln T + G(aT)] - \mathrm{I}_{20}^{c}(a,b,T)$$

but an analysis similar to the one above can also be carried out in terms of J_5.

Case I, $a \le b$. We integrate I_{20} by parts:

$u = \ln x$ $\qquad$ $dv = e^{-a^2x^2}\,\mathrm{erf}(bx)dx$

$du = dx/x$ $\qquad$ $v = -J_5(a,b,x)$.

Then

$$\mathrm{I}_{20}(a,b,T) = J_5(a,b,T)\ln T + \int_T^\infty \frac{J_5(a,b,x)}{x}dx$$

where $J_5(a,b,x)$ is the J function of Folder 5. For $a \le b$, we have (from Folder 5),

$$J_5(a,b,x) = \frac{\sqrt{\pi}}{2a}\mathrm{erfc}(ax) - I_5(a,b,x)$$

$$I_5(a,b,x) = \frac{1}{2\sqrt{\pi}\sqrt{a^2+b^2}}\sum_{k=0}^{\infty} C_k\left(\frac{a^2}{a^2+b^2}\right)^k E_{k+3/2}[x^2(a^2+b^2)], \qquad a \le b$$

$$C_k = (1/2)_k/k! \qquad k \ge 0.$$

Then, since the analysis for $I_5(a,b,x)$ is the same as that for I_{20}^{c},

$$I_{20}(a,b,T) = J_5(a,b,T)\ln T + \frac{\sqrt{\pi}}{2a}G(aT) - \frac{1}{2a\sqrt{\pi}}E_1(X)\tan^{-1}\left(\frac{a}{b}\right) + \frac{S(a,b,T)}{4\sqrt{\pi}\sqrt{a^2+b^2}}$$

where G(aT) is computed in Folder 16.

Case II, $a > b$. For this case, we note that if we integrate by parts,

$$I_{19}(a,b,T) = \int_0^T \frac{\operatorname{erf}(aw)\operatorname{erf}(bw)}{w}dw,$$

we get a reflexive relation. Let

$u = \operatorname{erf}(aw)\operatorname{erf}(bw)$ $\qquad\qquad dv = dw/w$

$$du = \left[\frac{2a}{\sqrt{\pi}}e^{-a^2w^2}\operatorname{erf}(bw) + \frac{2b}{\sqrt{\pi}}e^{-b^2w^2}\operatorname{erf}(aw)\right]dw \qquad v = \ln w$$

Then, with

$$\int_0^T e^{-a^2w^2} erf(bw)\ln w dw = \int_0^\infty e^{-a^2w^2} erf(bw)\ln w dw - \int_T^\infty e^{-a^2w^2} erf(bw)\ln w dw$$
$$= \left[I_{20}(a,b,0) - I_{20}(a,b,T) \right]$$

we get

$$I_{19}(a,b,T) = \operatorname{erf}(aT)\operatorname{erf}(bT)\ln T - \frac{2a}{\sqrt{\pi}}\left[I_{20}(a,b,0) - I_{20}(a,b,T) \right] - \frac{2b}{\sqrt{\pi}}\left[I_{20}(b,a,0) - I_{20}(b,a,T) \right]$$

or

$$I_{20}(a,b,T) = I_{20}(a,b,0) - \frac{\sqrt{\pi}}{2a}[\operatorname{erf}(aT)\operatorname{erf}(bT)\ln T - I_{19}(a,b,T)] + \frac{b}{a}\left[I_{20}(b,a,0) - I_{20}(b,a,T)\right].$$

The evaluations of $I_{20}(a,b,0)$ and $I_{20}(b,a,0)$ are considered next. Similar reflexive relations also occur in Folders 5 and 7 and are used in a similar way. Numerical experiments show that this formula gives absolute accuracies down to magnitudes on the order of unit round-off (increasing T).

Special Cases $T \to 0$

Case I, $a \le b$, $T \to 0$

$$I_{20}^c(a,b,T) = I_5(a,b,T)\ln T + \frac{1}{2a\sqrt{\pi}}E_1(X)\tan^{-1}\left(\frac{a}{b}\right) - \frac{S(a,b,T)}{4\sqrt{\pi}\sqrt{a^2+b^2}},\ X = T^2(a^2+b^2),\ a \le b.$$

Now $I_5(a,b,T) = \dfrac{1}{a\sqrt{\pi}}\tan^{-1}\dfrac{a}{b} + O(T)$, $E_{k+3/2}(0) = \dfrac{1}{k+1/2}$,

$$E_1(X) = -\gamma - 2\ln T - \ln(a^2+b^2) + O(T), \qquad S(a,b,0) = \sum_{k=0}^{\infty} \frac{C_k}{(k+1/2)^2}\left(\frac{a^2}{a^2+b^2}\right)^k$$

and

$$\mathrm{I}^{c}_{20}(\mathrm{a},\mathrm{b},0)=\frac{1}{2\mathrm{a}\sqrt{\pi}}[-\gamma-\ln(\mathrm{a}^2+\mathrm{b}^2)]\tan^{-1}\frac{\mathrm{a}}{\mathrm{b}}-\frac{\mathrm{S}(\mathrm{a},\mathrm{b},0)}{4\sqrt{\pi}\sqrt{\mathrm{a}^2+\mathrm{b}^2}}, \qquad a\le b$$

For I_{20} we have

$$I_{20}(a,b,T)=\frac{\sqrt{\pi}}{2a}[\mathrm{erfc}(aT)\ln T+G(aT)]-I^{c}_{20}(a,b,T)$$

and from Folder 16, we have

$$\mathrm{G(aT)}=\mathrm{F(aT)}-\frac{\gamma}{2}-\ln(2\mathrm{aT})$$

which gives

$$I_{20}(a,b,T)=\frac{\sqrt{\pi}}{2a}\left[F(aT)-\frac{\gamma}{2}-\ln(2a)-erf(aT)\ln T\right]-I^{c}_{20}(a,b,T)$$

$$I_{20}(a,b,0)=\frac{\sqrt{\pi}}{2a}[-\frac{\gamma}{2}-\ln(2a)]-I^{c}_{20}(a,b,0), \qquad a\le b.$$

<u>Case II, $a>b$</u>, $\mathrm{T}\to 0$

For a>b, we get

$$I^{c}_{20}(a,b,0)=\frac{\sqrt{\pi}}{2a}[I^{c}_{19}(a,b,T)+erfc(aT)erfc(bT)\ln T]_{T\to 0}-\frac{b}{a}I^{c}_{20}(b,a,0), \qquad \text{a>b}$$

From Folder 19 and G above,

$$I^{c}_{19}(a,b,T)=G(bT)erfc(aT)+R(a,b,T)=G(aT)erfc(bT)+R(b,a,T) \quad , \qquad \text{a>b}$$

$$I^{c}_{20}(a,b,0)=\frac{\sqrt{\pi}}{2a}[-\frac{\gamma}{2}-\ln(2a)+R(b,a,0)]-\frac{b}{a}I^{c}_{20}(b,a,0), \qquad \text{a>b}$$

where R from Folder 19 is given by

$$R(a,b,T)=-\frac{a}{2\pi\sqrt{a^2+b^2}}\sum_{k=0}^{\infty}C_k\left(\frac{a^2}{a^2+b^2}\right)^k G_{k+3/2}(X)+\frac{a}{\sqrt{\pi}}\ln\left[\frac{b^2}{a^2+b^2}\right]I_5(a,b,T)$$
$$+\frac{a}{2\pi\sqrt{a^2+b^2}}\sum_{n=1}^{\infty}\left(\frac{a^2}{a^2+b^2}\right)^n E_{n+3/2}(X)\sum_{m=1}^{n}\frac{C_{n-m}}{m}, \quad X=T^2(a^2+b^2), \quad \text{a>b}$$

with $C_k=(1/2)_k/k!$, $G_{k+3/2}(0)=\frac{1}{(k+1/2)^2}$, $\mathrm{E}_{n+3/2}(0)=\frac{1}{(n+1/2)}$, $\mathrm{I}_5(a,b,0)=\frac{1}{a\sqrt{\pi}}\tan^{-1}\frac{a}{b}$

and $I^{c}_{20}(b,a,0)$ is computed above in case I since the first argument is smaller than the second argument. From the relation

$$I_{20}(a,b,T)=\frac{\sqrt{\pi}}{2a}[\mathrm{erfc}(aT)\ln T+G(aT)]-I^{c}_{20}(a,b,T)$$

we get

$$I_{20}(a,b,0)=\frac{\sqrt{\pi}}{2a}[-\frac{\gamma}{2}-\ln(2a)]-I^{c}_{20}(a,b,0)=-\frac{\sqrt{\pi}}{2a}R(b,a,0)+\frac{b}{a}I^{c}_{20}(b,a,0) \qquad \text{a>b}$$

Special Case $a = 0$. Take Case I, $a \le b$, and let $a \to 0$. Then, integration by parts, with the relation

$$ierfc(bx) = -bx\, erfc(bx) + \frac{e^{-b^2x^2}}{\sqrt{\pi}}$$

gives

$$I_{20}^{c}(0,b,T) = \int_T^\infty \operatorname{erfc}(bx)\ln x dx = \frac{\ln T}{b}\operatorname{ierfc}(bT) - \frac{\operatorname{ierfc}(bT)}{b} + \frac{E_1(b^2T^2)}{2b\sqrt{\pi}}$$

or

$$I_{20}^{c}(0,b,T) = \frac{\operatorname{ierfc}(bT)}{b}\ln\frac{T}{e} + \frac{E_1(b^2T^2)}{2b\sqrt{\pi}}.$$

Special Case, $b = 0$. This case was evaluated explicitly in Folder 16,

$$I_{20}^{c}(a,0,T) = \int_T^\infty e^{-a^2x^2}\ln x dx = \frac{\sqrt{\pi}}{2a}[\operatorname{erfc}(aT)\ln T + G(aT)]$$

where $G(x)$ can be evaluated from double precision function DGERFC.

Folder 21

Evaluation of

$$\mathbf{U(a,b,t) = e^{a^2t+2ab}\,erfc(a\sqrt{t} + b/\sqrt{t})}$$

$$\mathbf{V(a,b,t) = \int_0^t U(a,b,\tau)d\tau, \quad I_{21}(a,b,c,t) = \int_0^t U(a,b,\tau)erf(c/\sqrt{\tau})d\tau}$$

$$\mathbf{I_{21}^c(a,b,c,t) = \int_0^t U(a,b,\tau)erfc(c/\sqrt{\tau})d\tau, \quad J_{21}(a,b,c,t) = \int_0^t \frac{U(a,b,\tau)e^{-c^2/\tau}}{\tau^{3/2}}d\tau}$$

$$\boldsymbol{a > 0, \quad b > 0, \quad c > 0, \quad t > 0}$$

$$\boldsymbol{G_n(a,b,T) = \int_T^\infty \frac{e^{-a^2w^2}\, i^n\, \mathrm{erfc}(bw)}{w^n}dw, \quad a > 0, \quad b \ge 0, \quad T > 0, \quad n \ge 0}$$

Donald E. Amos, September 2002

Summary

Note that $V(t)$ can be evaluated by the LaPlace transform

$$V(t) = V_1(t) - V_2(t) + V_3(t)$$

where

$$V_1(t) = \frac{2}{a}\sqrt{\frac{t}{\pi}}e^{-b^2/t}, \qquad V_2(t) = \left(\frac{1}{a^2} + \frac{2b}{a}\right)\mathrm{erfc}\left(\frac{b}{\sqrt{t}}\right), \quad V_3(t) = \frac{U(t)}{a^2}.$$

The computational form for a away from zero is

$$V(a,b,t) = \frac{2\sqrt{t}}{a}ierfc\left(\frac{b}{\sqrt{t}}\right) + \frac{1}{a^2}\left[U(a,b,t) - erfc\left(\frac{b}{\sqrt{t}}\right)\right].$$

Next, we note that for $t \le c^2/36$, $c/\sqrt{t} \ge 6$, $\mathrm{erf}(c/\sqrt{t}) = 1 + O(10^{-16})$ uniformly in $t \le c^2/36$. Therefore, to double precision arithmetic, we have

$$I_{21}(a,b,c,t) = V(t) + O(10^{-16}) \quad \text{for} \quad t \le c^2/36.$$

For the general cases we develop forms which are evaluated by series or by quadrature for the default cases. The results are:

$$I_{21}(a,b,c,t) = V(t)\,\mathrm{erf}(c/\sqrt{t}) + \frac{c}{\sqrt{\pi}}\left[\frac{2}{a\sqrt{\pi}}E_1(X) - 2\left(\frac{1}{a^2} + \frac{2b}{a}\right)I_5(c,b,T) + \frac{2\sqrt{t}}{a^2\sqrt{\pi}}S_1(a,b,c,t)\right]$$

$$I_{21}^c(a,b,c,t) = V(t)\,\mathrm{erfc}(c/\sqrt{t}) - \frac{c}{\sqrt{\pi}}\left[\frac{2}{a\sqrt{\pi}}E_1(X) - 2\left(\frac{1}{a^2} + \frac{2b}{a}\right)I_5(c,b,T) + \frac{2\sqrt{t}}{a^2\sqrt{\pi}}S_1(a,b,c,t)\right]$$

where $X = (b^2+c^2)/t$, I_5 is the I function of Folder 5 and $S_1(a,b,c,t)$ in computational form suitable for a quadrature is

$$S_1(a,b,c,t) = e^{-X} \int_0^\infty \frac{e^{-2Bw-w^2}}{c^2+(b+w\sqrt{t})^2}\,dw, \qquad B = a\sqrt{t} + b/\sqrt{t}\,.$$

For J_{21}, we get

$$J_{21}(a,b,c,t) = \frac{2\sqrt{t}}{\sqrt{\pi}} S_1(a,b,c,t)\,.$$

A series for large parameter L ($L \ge 2$) is

$$S_1(a,b,c,t) = \frac{e^{-X}}{2tB} \sum_{k=0}^\infty \frac{U_k(x)}{L^k} [e^{B^2} E_{(k+3)/2}(B^2)],$$

$$\mathrm{X} = (b^2+c^2)/t, \qquad B = a\sqrt{t} + b/\sqrt{t}\,, \qquad L = \frac{B}{\sqrt{a^2t + c^2/t}}, \qquad x = \frac{a\sqrt{t}}{\sqrt{a^2t+c^2/t}},$$

and $U_k(x)$, $k \ge 0$, are Chebyshev polynomials of the second kind which can be generated by forward recurrence on their three-term recurrence relation.

The expressions for $V(t)$, I_{21} and I_{21}^c above contain reciprocal powers of a, but the integrals are analytic functions of a. Therefore, to avoid losses of significance by small differences of large numbers when a is small, we develop the power series in the parameter a. The results are:

$$V(t) = 4t \sum_{n=2}^\infty i^n \operatorname{erfc}\left(\frac{b}{\sqrt{t}}\right)(-2a\sqrt{t})^{n-2}$$

$$I_{21}(a,b,c,t) = V(t)\operatorname{erf}\left(\frac{c}{\sqrt{t}}\right) + \frac{4c\sqrt{t}}{\sqrt{\pi}} e^{-X} \sum_{n=2}^\infty (-2a\sqrt{t})^{n-2} y_n\,, \qquad 0 \le a\sqrt{t} \le 1$$

$$I_{21}^c(a,b,c,t) = V(t)\operatorname{erfc}\left(\frac{c}{\sqrt{t}}\right) - \frac{4c\sqrt{t}}{\sqrt{\pi}} e^{-X} \sum_{n=2}^\infty (-2a\sqrt{t})^{n-2} y_n$$

$$J_{21}(a,b,c,t) = \frac{e^{-X}}{\sqrt{t}} \sum_{n=0}^\infty (-2a\sqrt{t})^n y_n \qquad X = (b^2+c^2)/t,$$

where

$$y_n = 2T^{n-1} e^X G_n(c,b,T), \quad , \; G_n(c,b,T) = \int_T^\infty e^{-c^2w^2} \frac{i^n \operatorname{erfc}(bw)}{w^n}\,dw \qquad T = \frac{1}{\sqrt{t}}\,.$$

An extensive stability analysis on the 3-term recurrence for y_n shows that the recurrence must be started with two (quadrature) values near the index [2X]. Again, the integrand of G_n is analytic and DQUAD8 can be used to compute the two starting values.

The APPENDIX describes how DQUAD8 is applied to compute the quadratures for this folder.

Evaluation of $V(t)$ Let

$$U(t) = e^{a^2 t + 2ab}\,\mathrm{erfc}(a\sqrt{t} + b/\sqrt{t}), \qquad V(t) = \int_0^t U(\tau)d\tau .$$

Then, the LaPlace transform gives ([4], Vol. 1, p247)

$$\overline{U}(p) = \frac{e^{-2b\sqrt{p}}}{\sqrt{p}(\sqrt{p}+a)}, \qquad \overline{V}(p) = \overline{U}(p)/p$$

and ([4], Vol. 1, p247)

$$V(t) = V_1(t) - V_2(t) + V_3(t)$$

where

$$V_1(t) = \frac{2}{a}\sqrt{\frac{t}{\pi}}e^{-b^2/t}, \qquad V_2(t) = \left(\frac{1}{a^2} + \frac{2b}{a}\right)\mathrm{erfc}\left(\frac{b}{\sqrt{t}}\right), \qquad V_3(t) = \frac{U(t)}{a^2}.$$

With $ierfc(x) = \dfrac{e^{-x^2}}{\sqrt{\pi}} - x\,erfc(x)$ we get the preferred computational for for a away from zero

$$V(a,b,t) = \frac{2\sqrt{t}}{a} ierfc\left(\frac{b}{\sqrt{t}}\right) + \frac{1}{a^2}\left[U(a,b,t) - erfc\left(\frac{b}{\sqrt{t}}\right)\right].$$

Evaluation of $I_{21}(a,b,c,t)$ First we note that for $t \le c^2/36$ and $\tau \le t$, $c/\sqrt{\tau} \ge 6$,

$$\mathrm{erf}\left(\frac{c}{\sqrt{\tau}}\right) = 1 + O(10^{-16})$$

uniformly in $\tau \le t$. Then, to double precision arithmetic,

$$I_{21}(a,b,c,t) = V(t) + O(10^{-16}) \qquad t \le c^2/36.$$

To obtain working formulas for more general cases, we integrate I_{21} by parts. Let

$$u = \mathrm{erf}(c/\sqrt{\tau}) \qquad dv = U(\tau)d\tau$$

$$du = -\frac{c}{\sqrt{\pi}}\frac{e^{-c^2/\tau}}{\tau^{3/2}}d\tau \qquad v = V(\tau)$$

where $U(\tau)$ and $V(\tau)$ are defined above. Then

$$I_{21}(a,b,c,t) = V(t)\,\mathrm{erf}(c/\sqrt{t}) + \frac{c}{\sqrt{\pi}}\int_0^t \frac{V(\tau)e^{-c^2/\tau}}{\tau^{3/2}}\,d\tau, \qquad V(t) = V_1(t) - V_2(t) + V_3(t)$$

$$= V(t)\,\mathrm{erf}(c/\sqrt{t}) + \frac{c}{\sqrt{\pi}}[W_1(t) - W_2(t) + W_3(t)]$$

where $W_1(t)$, $W_2(t)$ and $W_3(t)$ are analyzed as follows:

<u>Term $W_1(t)$</u> = $\displaystyle\int_0^t \frac{V_1(\tau)}{\tau^{3/2}}e^{-c^2/\tau}d\tau = \frac{2}{a\sqrt{\pi}}\int_0^t \frac{e^{-(b^2+c^2)/\tau}}{\tau}d\tau$

Let $\tau = t/v$, $d\tau = -t/v^2 dv$, and

$$W_1(t) = \frac{2}{a\sqrt{\pi}}\int_1^\infty \frac{e^{-(b^2+c^2)v/t}}{v}dv = \frac{2}{a\sqrt{\pi}}E_1\left(\frac{b^2+c^2}{t}\right).$$

<u>Term $W_2(t)$</u> = $\displaystyle\int_0^t \frac{V_2(\tau)}{\tau^{3/2}}e^{-c^2/\tau}d\tau = \left(\frac{1}{a^2}+\frac{2b}{a}\right)\int_0^t \frac{erfc(b/\sqrt{\tau})}{\tau^{3/2}}e^{-c^2/\tau}d\tau$

Let $\tau = 1/v^2$, $d\tau = (-2/v^3)dv$. Then

$$W_2(t) = 2\left(\frac{1}{a^2}+\frac{2b}{a}\right)\int_T^\infty e^{-c^2v^2}\,\mathrm{erfc}(bv)dv = 2\left(\frac{1}{a^2}+\frac{2b}{a}\right)I_5(c,b,T)$$

where $T = 1/\sqrt{t}$ and I_5 is the I function of Folder 5.

<u>Term $W_3(t)$</u> = $\displaystyle\int_0^t \frac{V_3(\tau)e^{-c^2/\tau}}{\tau^{3/2}}d\tau = \frac{1}{a^2}\int_0^t \frac{U(\tau)e^{-c^2/\tau}}{\tau^{3/2}}d\tau = \frac{1}{a^2}J_{21}(a,b,c,t)$

This is more complicated than $W_1(t)$ or $W_2(t)$ and results in an integral which, while represented for special values of the parameters, requires a quadrature for the general (default) case. Notice that J_{21} can be rephrased in terms of W_3.

$$J_{21}(a,b,c,t) = a^2 W_3(t).$$

The quadrature procedure using DQUAD8 is described in the APPENDIX. The analysis to produce forms which are suitable for DQUAD8 follows.

To proceed, from Folder 4, we have

$$\frac{2e^{-b^2/t}}{\sqrt{\pi t}}\int_0^\infty e^{-(2a+2b/t)y}e^{-y^2/t}dy = e^{a^2t+2ab}\,\mathrm{erfc}(a\sqrt{t}+b/\sqrt{t})$$

and

$$W_3(t) = \frac{2}{a^2\sqrt{\pi}}\int_0^t \frac{e^{-(b^2+c^2)/\tau}}{\tau^2}d\tau\int_0^\infty e^{-2(a+b/\tau)y}e^{-y^2/\tau}dy$$

$$= \frac{2}{a^2\sqrt{\pi}}\int_0^\infty e^{-2ay}dy\int_0^t \frac{e^{-[c^2+(b+y)^2]/\tau}}{\tau^2}d\tau$$

$$= \frac{2}{a^2\sqrt{\pi}}\int_0^\infty e^{-2ay}\frac{e^{-[c^2+(b+y)^2]/t}}{[c^2+(b+y)^2]}dy$$

Let $b+y = x$. Then

$$W_3(t) = \frac{2e^{2ab}}{a^2\sqrt{\pi}}\int_b^\infty e^{-2ax}\frac{e^{-(c^2+x^2)/t}}{c^2+x^2}dx$$

Now,

$$[c^2+x^2+2axt]/t = (x+at)^2/t+c^2/t-a^2t$$

and with $w = (x+at)/\sqrt{t}$, we get

$$W_3(t) = \frac{2\sqrt{t}}{a^2\sqrt{\pi}}S_1(a,b,c,t)$$

$$S_1(a,b,c,t) = e^{a^2t+2ab-c^2/t}\int_B^\infty \frac{e^{-w^2}}{c^2+t(w-a\sqrt{t})^2}dw, \qquad B = a\sqrt{t}+b/\sqrt{t}\,.$$

A better numerical form is obtained from the substitution $w = B+v$

$$S_1(a,b,c,t) = e^{-(b^2+c^2)/t}\int_0^\infty \frac{e^{-2Bv-v^2}}{c^2+(b+v\sqrt{t})^2}dv$$

This result is suitable for the application of a polynomial type quadrature routine like DGAUS8. The implementation is described in the APPENDIX using DQUAD8. The truncation error can be estimated for $R > 0$ by

$$T_R = \int_R^\infty \frac{e^{-2Bw-w^2}}{c^2+(b+w\sqrt{t})^2}dw, \qquad |T_R| < \frac{(\sqrt{\pi}/2)\,\mathrm{erfc}(R)e^{-2BR}}{c^2+(b+R\sqrt{t})^2} \qquad R > 0$$

and from Mill's ratio ([6], Chapter 7, 7.1.13), we get for R > 0

$$\left|T_R\right| < \frac{e^{-2BR}}{c^2+(b+R\sqrt{t})^2}\cdot\frac{e^{-R^2}}{(R+\sqrt{R^2+4/\pi})} < \frac{1}{c^2+(b+R\sqrt{t})^2}\cdot\frac{e^{-R^2-2BR}}{2R} = B_R\,, \quad \text{R} > 0$$

How B_R is incorporated into the quadrature computation in subroutine DQUAD8 is described in the APPENDIX.. The scale of integration SIG for DQUAD8 can be estimated from

$$\text{SIG} = \min\left[\frac{1}{2B},\frac{1}{\sqrt{2}}\right]\cdot m\,, \qquad\qquad m = 4 \text{ or } 5.$$

Term $W_3(t)$ by Series for $L = \dfrac{a\sqrt{t}+b\sqrt{t}}{\sqrt{a^2t+c^2/t}} \geq 2$

We start with the quadrature from above and manipulate the denominator of the integrand,

$$c^2+t(w-a\sqrt{t})^2 = t[w^2-2a\sqrt{t}w+a^2t+c^2/t]$$

to get

$$S_1 = \frac{e^{a^2t+2ab-c^2/t}}{t}\int_B^\infty \frac{e^{-w^2}\,dw}{w^2\left[1-\dfrac{2a\sqrt{t}}{w}+\dfrac{a^2t+c^2/t}{w^2}\right]}.$$

Let $w = v\sqrt{a^2t+c^2/t}$. Then

$$S_1(t) = K(t)\int_L^\infty \frac{e^{-(a^2t+c^2/t)v^2}}{v^2\left[1-\dfrac{2x}{v}+\dfrac{1}{v^2}\right]}dv\,, \qquad x = \frac{a\sqrt{t}}{\sqrt{a^2t+c^2/t}}$$

where

$$K(t) = \frac{1}{t}\frac{e^{a^2t+2ab-c^2/t}}{\sqrt{a^2t+c^2/t}}\,, \qquad L = \frac{B}{\sqrt{a^2t+c^2/t}} = \frac{a\sqrt{t}+b/\sqrt{t}}{\sqrt{a^2t+c^2/t}}.$$

The generating function for Chebyshev polynomials of the second kind is

$$\frac{1}{1-2xz+z^2} = \sum_{k=0}^{\infty} U_k(x)z^k\,, \qquad |z|<1, \qquad |x|\leq 1\,,$$

With $z = 1/v$ and $x = a\sqrt{t}/\sqrt{a^2t+c^2/t} < 1$, we have

$$S_1(t) = K(t)\sum_{k=0}^{\infty} U_k(x)\int_L^\infty \frac{e^{-(a^2t+c^2/t)v^2}}{v^{k+2}}dv$$

and with $v = L\sqrt{u}$, we get

$$S_1(t) = \frac{e^{-X}}{2tB}\sum_{k=0}^{\infty} \frac{U_k(x)}{L^k}[e^{B^2} E_{(k+3)/2}(B^2)], \quad B = a\sqrt{t} + b/\sqrt{t}\,, \; X = (b^2+c^2)/t, \quad x = \frac{a\sqrt{t}}{\sqrt{a^2 t + c^2/t}}$$

where we have convergence for $L > 1$. For computational purposes, we want $L \geq 2$ for rapid convergence since

$$|U_k(x)| \leq |U_k(1)| = k+1, \qquad |x| \leq 1, \qquad e^z E_v(z) \sim \frac{1}{z+v} \qquad \text{for } v \to \infty.$$

The polynomials $U_k(z)$ can be generated in a stable manner by forward recurrence on

$$U_0(z) = 1, \qquad U_1(z) = 2z$$

$$U_{k+1}(z) = 2zU_k(z) - U_{k-1}(z), \qquad k \geq 1.$$

Then, in summary,

$$W_3(t) = \frac{2\sqrt{t}}{a^2\sqrt{\pi}} S_1(a,b,c,t)$$

and

$$J_{21}(a,b,c,t) = \frac{2\sqrt{t}}{\sqrt{\pi}} S_1(a,b,c,t)$$

whether S_1 is computed by series or quadrature. Also,

$$I_{21}^c(a,b,c,t) = \int_0^t U(t)\,\text{erfc}\left(\frac{c}{\sqrt{\tau}}\right) d\tau = V(t) - I_{21}(a,b,c,t)$$

$$= V(t)\,\text{erfc}\left(\frac{c}{\sqrt{t}}\right) - \frac{c}{\sqrt{\pi}}[W_1(t) - W_2(t) + W_3(t)]$$

$$I_{21}(a,b,c,t) = \int_0^t U(t)\,\text{erf}\left(\frac{c}{\sqrt{\tau}}\right) d\tau$$

$$= V(t)\,\text{erf}\left(\frac{c}{\sqrt{t}}\right) + \frac{c}{\sqrt{\pi}}[W_1(t) - W_2(t) + W_3(t)]$$

<u>Expansions for Small</u> $a\sqrt{t} \leq 1$

In the formula for $V(t)$

$$V(t) = \frac{2}{a}\sqrt{\frac{t}{\pi}}e^{-b^2/t} - \left(\frac{1}{a^2} + \frac{2b}{a}\right)\operatorname{erfc}\left(\frac{b}{\sqrt{t}}\right) + \frac{1}{a^2}U(a,b,t)$$

we notice that there are computational difficulties for small a. That is, we get losses of significance by small differences of large numbers. Furthermore,

$$V(t) = \int_0^t U(\tau)d\tau$$

is analytic in a which means that the series expansion in a cannot contain negative powers. This means that the expression above is an indeterminate form for $a \to 0$. The following series manipulations resolves this indeterminate form and makes $V(t)$ and

$$I_{21}(a,b,c,t) = V(t)\operatorname{erf}\left(\frac{c}{\sqrt{t}}\right) + \frac{c}{\sqrt{\pi}}\int_0^t \frac{V(\tau)e^{-c^2/\tau}}{\tau^{3/2}}d\tau$$

computable for small $a\sqrt{t}$.

We start with the Taylor expansion

$$e^{z^2}\operatorname{erfc}(z) = \sum_{n=0}^{\infty} (-1)^n 2^n e^{x^2} i^n \operatorname{erfc}(x)(z-x)^n$$

about $z = x$ where ([6], Chapter 7, 7.2.9)

$$\frac{d^n}{dz^n}[e^{z^2}\operatorname{erfc}(z)] = (-1)^n 2^n n! e^{z^2} i^n \operatorname{erfc}(z)$$

Let $z = x + y$. Then

$$e^{(x+y)^2}\operatorname{erfc}(x+y) = \sum_{n=0}^{\infty} (-1)^n 2^n e^{x^2} i^n \operatorname{erfc}(x) y^n$$

and with $x = b/\sqrt{t}$, $y = a\sqrt{t}$, we get

$$U(t) = e^{a^2t+2ab}\operatorname{erfc}(a\sqrt{t} + b/\sqrt{t}) = \sum_{n=0}^{\infty} i^n \operatorname{erfc}\left(\frac{b}{\sqrt{t}}\right)(-2a\sqrt{t})^n$$

and

$$V(a,b,t) = \frac{2\sqrt{t}}{a} ierfc\left(\frac{b}{\sqrt{t}}\right) + \frac{1}{a^2}\left[U(a,b,t) - erfc\left(\frac{b}{\sqrt{t}}\right)\right].$$

Then, with the series above for U(t) with the n=0 and n=1 terms written separately, V(t) becomes

$$V(t) = 4t\sum_{n=2}^{\infty} i^n \operatorname{erfc}\left(\frac{b}{\sqrt{t}}\right)(-2a\sqrt{t})^{n-2}$$

which shows $V(t)$ to be analytic in a.

The computation of $I_{21}(a,b,c,t)$ for $a \to 0$ is reduced to

$$I_{21}(a,b,c,t) = V(t)\,\mathrm{erf}\left(\frac{c}{\sqrt{t}}\right) + \frac{c}{\sqrt{\pi}}\int_0^t \frac{V(\tau)e^{-c^2/\tau}}{\tau^{3/2}}\,d\tau$$

$$I_{21}(a,b,c,t) = V(t)\,\mathrm{erf}\left(\frac{c}{\sqrt{t}}\right) + \frac{4c}{\sqrt{\pi}}\sum_{n=2}^{\infty} (-2a)^{n-2}\int_0^t e^{-c^2/\tau} i^n\,\mathrm{erfc}\left(\frac{b}{\sqrt{\tau}}\right)\tau^{(n-3)/2}\,d\tau$$

and, finally,

$$I_{21}(a,b,c,t) = V(t)\,\mathrm{erf}\left(\frac{c}{\sqrt{t}}\right) + \frac{8c\sqrt{t}}{\sqrt{\pi}}\sum_{n=2}^{\infty} (-2a\sqrt{t})^{n-2}[T^{n-1}G_n(c,b,T)], \qquad T = \frac{1}{\sqrt{t}},$$

where, under the transformation $\tau = 1/w^2$,

$$\frac{1}{2}\int_0^t e^{-c^2/\tau} i^n\,\mathrm{erfc}\left(\frac{b}{\sqrt{\tau}}\right)\tau^{(n-3)/2}\,d\tau = G_n(c,b,T) = \int_T^{\infty} e^{-c^2w^2}\,\frac{i^n\,\mathrm{erfc}(bw)}{w^n}\,dw$$

The next order of business is to develop the recurrence for G_n and study the stability properties of the recurrence. Integrate G_n by parts:

$$u = \frac{e^{-c^2w^2}}{w^n} \qquad\qquad dv = i^n\mathrm{erfc}(bw)\,dw$$

$$du = \left[-\frac{2c^2}{w^{n-1}}e^{-c^2w^2} - \frac{ne^{-c^2w^2}}{w^{n+1}}\right]dw \qquad\qquad v = -i^{n+1}\mathrm{erfc}(bw)/b$$

Then,

$$G_n = \frac{e^{-c^2T^2}i^{n+1}\,\mathrm{erfc}(bT)}{bT^n} - \frac{2c^2}{b}H_{n+1} - \frac{nG_{n+1}}{b}, \qquad T = \frac{1}{\sqrt{t}},$$

where

$$H_{n+1} = \int_T^{\infty} e^{-c^2w^2}\,\frac{i^{n+1}\,\mathrm{erfc}(bw)}{w^{n-1}}\,dw,$$

Now, $\quad i^{n+1}\,\mathrm{erfc}(bw) + \dfrac{bw}{n+1}i^n\,\mathrm{erfc}(bw) = \dfrac{i^{n-1}\,\mathrm{erfc}(bw)}{2(n+1)}$ $\qquad$ and

$$H_{n+1} + \frac{bH_n}{n+1} = \frac{1}{2(n+1)}G_{n-1}$$

Elimination of H from this pair of difference equations gives

$$n(n+1)G_{n+1} + 2nbG_n + (b^2 + c^2)G_{n-1} = \frac{e^{-c^2T^2}}{2T^n}\mathrm{i}^{\mathrm{n-1}}\mathrm{erfc}(bw)$$

Now we introduce the scaling

$$y_n = 2T^{n-1}G_n(c,b,T)e^X, \qquad X = (b^2 + c^2)/t = (b^2 + c^2)T^2$$

to get

$$n(n+1)y_{n+1} + 2n(bT)y_n + Xy_{n-1} = e^{b^2T^2} i^{n-1}\,\mathrm{erfc}(bT)$$

and, finally,

$$I_{21}(a,b,c,t) = V(t)\,\mathrm{erf}\left(\frac{c}{\sqrt{t}}\right) + \frac{4c\sqrt{t}}{\sqrt{\pi}}e^{-X}\sum_{n=2}^{\infty}(-2a\sqrt{t})^{n-2}y_n$$

$$V(t) = 4t\sum_{n=2}^{\infty} i^n\,\mathrm{erfc}\left(\frac{b}{\sqrt{t}}\right)(-2a\sqrt{t})^{n-2} \qquad 0 \le a\sqrt{t} \le 1$$

$$I_{21}^c(a,b,c,t) = V(t)\,\mathrm{erfc}\left(\frac{c}{\sqrt{t}}\right) - \frac{4c\sqrt{t}}{\sqrt{\pi}}e^{-X}\sum_{n=2}^{\infty}(-2a\sqrt{t})^{n-2}y_n$$

The introduction of y_n keeps the computation well scaled and makes a test for under flow on $\exp(-X)$ easy.

The computation for $J_{21}(a,b,c,t)$ for small $a\sqrt{t} \le 1$ is direct:

$$J_{21}(a,b,c,t) = \int_0^t \frac{U(\tau)e^{-c^2/\tau}}{\tau^{3/2}}d\tau = \sum_{n=0}^{\infty}(-2a)^n\int_0^t e^{-c^2/\tau}i^n\,\mathrm{erfc}\left(\frac{b}{\sqrt{\tau}}\right)\tau^{(n-3)/2}d\tau$$

$$= 2\sum_{n=0}^{\infty}(-2a)^n G_n(c,b,T), \qquad T = \frac{1}{\sqrt{t}},$$

or, in terms of y_n,

$$J_{21}(a,b,c,t) = Te^{-X}\sum_{n=0}^{\infty}\left(\frac{-2a}{T}\right)^n[2T^{n-1}G_n(c,b,T)e^X]$$

$$= \frac{e^{-X}}{\sqrt{t}}\sum_{n=0}^{\infty}(-2a\sqrt{t})^n y_n, \qquad X = (b^2+c^2)/t.$$

The y_n sequence is computed in subroutine GNSEQ.

Estimates for y_n and the Convergence of the Series for $a\sqrt{t} \le 1$

It is easy to estimate y_n

$$y_n = 2T^{n-1}e^{(b^2+c^2)T^2}\int_T^\infty e^{-c^2w^2}\,\frac{i^n\,\mathrm{erfc}(bw)}{w^n}dw$$

using the monotonicity of $i^n\mathrm{erfc}(bw)$ and $E_n(x) \le e^{-x}/(x+n-1)$,

$$|y_n| \le e^{(b^2+c^2)T^2}i^n\,\mathrm{erfc}(bT)E_{(n+1)/2}(c^2T^2) \le e^{(b^2+c^2)T^2}\,\frac{i^n\,\mathrm{erfc}(0)e^{-c^2T^2}}{(n-1)/2+c^2T^2} \le \frac{2e^{b^2T^2}}{2^n\Gamma\left(1+n/2\right)(n-1+2c^2T^2)}$$

Consequently, the terms of the series in $a\sqrt{t} \le 1$ for $V(a,b,t)$, $I_{21}(a,b,c,t)$, and $I_{21}^c(a,b,c,t)$ are dominated by $K/\Gamma\left(1+n/2\right)$ which is sufficient to get relative errors $O(10^{-16})$ in less than 50 terms.

In the stability analysis to follow, we need an asymptotic estimate for y_n, $n \to \infty$. We note that $i^n erfc(w)$ can be expressed in terms of the confluent hypergeometric function U [6, p.300 (7.2.12); p.504 (13.1.3) and p.505 (13.1.29)]

$$i^n erfc(w) = \frac{e^{-w^2}}{2^n\sqrt{\pi}}U(\frac{n+1}{2},\frac{1}{2},w^2) = \frac{we^{-w^2}}{2^n\sqrt{\pi}}U(\frac{n}{2}+1,\frac{3}{2},w^2)$$

and with the result from [5, p.80 (4.6.43)] we get the results of [3],

$$i^n\,\mathrm{erfc}(w) \sim e^{-(w^2/2+w\sqrt{2n})}\Big/\left[2^n\Gamma\left(\frac{n}{2}+1\right)\right] \qquad n \to \infty, \qquad |w| \text{ bounded.}$$

The analysis below is somewhat heuristic. We argue that this result can be applied on (T,∞) even though w is required to be bounded. The argument is that the multiplier $e^{-c^2w^2}/w^n$ decreases rapidly for c and n sizeable and that the truncation at $w = R$ for errors $O(10^{-16})$ effectively bounds w on (T,R). Taking $R \to \infty$ does not contribute significantly because of the strong convergence of the integral. Then,

$$y_n \sim \frac{T^{n-1}e^X}{2^n\Gamma\left(\frac{n}{2}+1\right)}\int_T^\infty \frac{e^{-\gamma w^2-bw\sqrt{2n}}}{w^n}dw\,, \qquad \gamma = c^2+b^2/2\,, \qquad X = (b^2+c^2)T^2 \quad ,$$

Now we compute the asymptotic estimate of this integral by integrating the asymptotic estimate of

$$Y(\rho) = \int_T^\infty e^{-\gamma w^2-\rho w}dw = e^{b^2/4\gamma}\int_T^\infty e^{-\gamma(w+\rho/(2\gamma))2}dw = e^{b^2/4\gamma}\,\frac{\sqrt{\pi}}{2\sqrt{\gamma}}erfc\left[\sqrt{\gamma}(T+\frac{\rho}{2\gamma})\right]$$

n times on (ρ,∞) and setting $\rho = b\sqrt{2n}$ in the final result. Then using

$$erfc(x) \sim e^{-x^2}/(x\sqrt{\pi}),\;\; x \to \infty$$

we have

$$Y(\rho) \approx \frac{e^{-\gamma T^2 - \rho T}}{2\gamma T + \rho} \qquad \text{and} \qquad \int_\rho^\infty Y(u)du \approx e^{-\gamma T^2} \int_\rho^\infty \frac{e^{-uT}}{2\gamma T + u} du$$

The substitution $u = -2\gamma T + v$ gives

$$\int_\rho^\infty Y(u)du = \int_T^\infty \frac{e^{-\gamma w^2 - \rho w}}{w} dw \approx e^{-\gamma T^2} \int_\rho^\infty \frac{e^{-uT}}{2\gamma T + u} du = e^{\gamma T^2} \int_{\rho+2\gamma T}^\infty \frac{e^{-vT}}{v} dv = e^{\gamma T^2} E_1(\rho T + 2\gamma T^2)$$

The remaining n-1 integrations are easy with $\int_x^\infty E_n(av)dv = E_{n+1}(ax)/a$ and

$$\int_T^\infty \frac{e^{-\gamma w^2 - \rho w}}{w^n} dw \approx \frac{e^{\gamma T^2}}{T^{n-1}} E_n(\rho T + 2\gamma T^2) \approx \frac{e^{-\gamma T^2} e^{-\rho T}}{T^{n-1}(n + \rho T + 2\gamma T^2)}.$$

Then with $\rho = b\sqrt{2n}$ and $E_v(x) \approx e^{-x}/(x+v), \quad v \to \infty$ we get

$$y_n \approx \frac{e^X e^{-\gamma T^2}}{2^n \Gamma(1+n/2)} \cdot \frac{e^{-bT\sqrt{2n}}}{(n + bT\sqrt{2n} + 2\gamma T^2)} = \frac{e^{bT^2/2} e^{-bT\sqrt{2n}}}{2^n \Gamma(1+n/2)(n + bT\sqrt{2n} + 2\gamma T^2)}$$

We apply this estimate for y_n in the stability analysis for the recurrence

$$n(n+1)y_{n+1} + 2nbTy_n + Xy_{n-1} = [e^{b^2T^2} i^{n-1} \operatorname{erfc}(bT)], \qquad n \ge 1,$$

to obtain a stability condition which, when applied during a recurrence, will keep unwanted solutions of the homogeneous equation from growing.

Stability Properties of the y_n Recurrence

Accurate answers using recurrence relations can only be achieved if unwanted solutions to the difference equation are kept in check. These unwanted solutions are introduced by a rounding error at one or more stages in the computation. In a typical stability analysis, we track the propagation of one or two errors made at a specific index *N*. Then, the complete solution is expressed as the wanted solution y_n plus the homogeneous solutions whose coefficients are not zero due to the errors made at index *N*. Thus, the homogeneous equation

$$n(n+1)h_{n+1} + 2n(bT)h_n + Xh_{n-1} = 0 \qquad n \ge 1,$$

has a solution

$$h_n = \frac{C^n}{n!} \text{ where } \frac{C^{n+1}}{(n-1)!} + \frac{2\alpha C^n}{(n-1)!} + \frac{\alpha^2(1+\beta^2)C^{n-1}}{(n-1)!} = 0, \qquad \alpha = \frac{b}{\sqrt{t}} = bT, \ \beta = \frac{c}{b},$$

or $\quad C^2+2\alpha C+\alpha^2(1+\beta^2) = 0$. Then,

$$C_1 = -\alpha+i\alpha\beta, \qquad C_2 = -\alpha-i\alpha\beta$$

and, taking into account that $\mathrm{Re}(C_1)$ and $\mathrm{Re}(C_2)$ are negative,

$$h_n^{1,2} = h_n e^{\pm in\theta}, \qquad h_n = \frac{C^n}{n!}, \qquad C^2 = \alpha^2(1+\beta^2), \qquad \theta = \pi - \tan^{-1}\beta$$

Now, any of the following pairs can be used for (h_n^1, h_n^2)

$$(e^{in\theta}, e^{-in\theta}), \qquad (\cos n\theta, \sin n\theta), \qquad (T_n(x), U_{n-1}(x))$$

where $T_n(x) = \cos n\theta$ and $U_{n-1}(x) = \sin n\theta/\sin\theta$ are Chebyshev polynomials of the first and second kinds with $x = \cos\theta = -1/\sqrt{1+\beta^2}$.

Thus, the complete solution of the non-homogeneous equation is given by

$$Y_n = [C_1T_n(x)+C_2U_{n-1}(x)]h_n + y_n$$

where y_n is the desired solution. The relative error e_n is

$$e_n = \frac{Y_n - y_n}{y_n} = [C_1T_n(x) + C_2U_{n-1}(x)]\frac{h_n}{y_n}.$$

To study the effects of a rounding error in e_N and e_{N+1} for some N, we compute C_1 and C_2 from

$$\frac{y_N e_N}{h_N} = C_1T_N(x) + C_2U_{N-1}(x)$$

$$\frac{y_{N+1}e_{N+1}}{h_{N+1}} = C_1T_{N+1}(x) + C_2U_N(x)$$

Notice that a solution is always possible since the coefficient matrix has a determinant

$$D = U_N(x)T_N(x)-T_{N+1}(x)U_{N-1}(x)$$

$$= \frac{\sin(N+1)\theta}{\sin\theta}\cos N\theta - \cos(N+1)\theta\frac{\sin N\theta}{\sin\theta} = \frac{\sin\theta}{\sin\theta} = 1$$

Now, $T_n(x)$ is bounded and $U_{n-1}(x)$ can be at most n for $x \le 1$. Thus, the factors $C_1T_n(x)+C_2U_{n-1}(x)$ vary slowly with n and the dominant factor in

$$e_n = [C_1T_n(x)+C_2U_{n-1}(x)]h_n/y_n$$

comes from h_n/y_n. Furthermore, $e_N \sim \mu U_r$ and $e_{N+1} \sim \nu U_r$ where μ and ν are small constants and U_r is unit roundoff. This puts C_1 and C_2 on the order of U_r. The amplification factor for e_n is

$$\frac{e_{n+1}}{e_n} = \frac{C_1 T_{n+1}(x) + C_2 U_n(x)}{C_1 T_n(x) + C_2 U_{n-1}(x)} \frac{h_{n+1}}{y_{n+1}} \cdot \frac{y_n}{h_n}$$

Thus, with $h_n = C^n/n!$ and $y_n \sim \dfrac{Ke^{-\alpha\sqrt{2n}}}{2^n \Gamma\left(\frac{n}{2}+1\right)(n+\alpha\sqrt{2n}+2\gamma T^2)}$, $\qquad \alpha = bT = b/\sqrt{t}$,

we have

$$\left|\frac{e_{n+1}}{e_n}\right| \sim K_n(N) \frac{2^{n+2}\Gamma\left(\frac{n}{2}+\frac{3}{2}\right)e^{-\alpha\sqrt{2n}}}{2^{n+1}\Gamma\left(\frac{n}{2}+1\right)e^{-\alpha\sqrt{2n+2}}} \cdot \frac{C}{n+1} \qquad \left(\frac{\Gamma(a+b)}{\Gamma(a+c)} \sim a^{b-c} \quad \text{for } a \to \infty\right)$$

$$\sim \frac{K_n(N)}{e^{-\alpha(\sqrt{2n+2}-\sqrt{2n})}}\left[\frac{2C}{n+1}\sqrt{\frac{n}{2}}\right] \sim f_n(N)\sqrt{\frac{2C^2}{n}}, \; C^2 = \alpha^2(1+\beta^2) = X, \quad X = (b^2+c^2)/t$$

where $f_n(N)$ varies slowly with n. Thus, if $n > [2C^2]$, then forward recurrence from $[2C^2]$ is stable since the amplification factor decreases with n. On backward recurrence, the amplification factor is

$$\left|\frac{e_n}{e_{n+1}}\right| \sim \frac{1}{f_n(N)}\sqrt{\frac{n}{2C^2}}, \qquad C^2 = \alpha^2(1+\beta^2) = X$$

and this decreases as n decreases from $[2C^2]$. Thus, for the next sections, to keep from amplifying errors, we define $N = [2X]$ and recur away from N.

<u>Computation of Starting Values for the y_n Sequence, $2 \le n \le M$</u>

The starting values y_0 and y_1 for $N = 0$ and $N = 1$ ($N = [2X]$) follow. In this case, we recur forward on n to generate our M+1 member sequence The first two members of the y_n sequence are:

$$y_0 = \frac{2e^X}{T}\int_T^\infty e^{-c^2w^2}\operatorname{erfc}(bw) = \frac{2e^X}{T} I_5(c,b,T), \qquad X = (b^2+c^2)/t, \qquad T = \frac{1}{\sqrt{t}},$$

and with the relation

$$i\operatorname{erfc}(bw) = (-bw)\operatorname{erfc}(bw) + \frac{e^{-b^2w^2}}{\sqrt{\pi}}$$

we get

$$y_1 = 2e^X \int_T^\infty e^{-c^2w^2} \frac{i\,\mathrm{erfc}(bw)}{w} dw = 2e^X \left[-bI_5(c,b,T) + \frac{1}{2\sqrt{\pi}} E_1(X) \right]$$

$$y_1 = -bTy_0 + \frac{1}{\sqrt{\pi}} [e^X E_1(X)], \qquad X = (b^2+c^2)/t.$$

Here $I_5(c,b,T)$ is the I function of Folder 5.

For $N \geq 2$, we use DQUAD8 quadratures (see APPENDIX) for indices k and $k-1$ to start the $M+1$ member sequence, where $k = \min(M,N)$ and $M \sim 50$ for double precision accuracy in the series. If $k = M$, then we compute y_{M-1} and y_M and recur backward for indices $0 \leq n < M$; if $k = N < M$, we compute y_{N-1} and y_N and recur forward and backward to complete the sequence. To avoid underflow and overflow problems and keep the computation well scaled, we combine exponentials which would lead to large and small multipliers. Then, for starting quadratures,

$$y_k = 2e^X T^{k-1} \int_T^\infty e^{-c^2w^2} \frac{i^k\,\mathrm{erfc}(bw)}{w^k} dw, \qquad X = (b^2+c^2)T^2$$

and with $\quad w = Tu, \quad u = 1+v, \quad$ we have

$$y_k = 2\int_0^\infty e^{-X(2+v)v} \left[\frac{e^{b^2T^2u^2} i^k\,\mathrm{erfc}(bTu)}{u^k} \right]_{u=1+v} dv$$

where $[e^{z^2} i^n\,\mathrm{erfc}(z)]$ can be calculated from DINERFC with KODE = 2. Here

$$\mathrm{SIG} = \min\left\{ \frac{1}{\sqrt{X}},\ \frac{1}{2X} \right\} \cdot m, \qquad m = 3 \text{ or } 4$$

is a scale of integration for DQUAD8 (see APPENDIX). Bounds on the truncation error can be computed using the results of [1]. Notice that no subtractions, which can lose significant digits, are used to compute y_k.

Numerical experiments were carried out in PROGRAM GNCOMP which compared a direct quadrature evaluation of y_n with each member of a sequence generated by SUBROUTINE GNSEQ which utilizes the y_n recurrence. The results seem to confirm the stability condition with no massive growth of errors.

Special Cases

$\underline{c = 0}$ $\quad I_{21}(a,b,0,t) = 0, \qquad I_{21}^c(a,b,0,t) = V(t)$

$\underline{a = 0 \text{ for } I_{21}(0,b,c,t)}$ $\quad$ In this case we have

$$I_{21}(0,b,c,t) = \int_0^t \operatorname{erfc}\left(\frac{b}{\sqrt{\tau}}\right)\operatorname{erf}\left(\frac{c}{\sqrt{\tau}}\right)d\tau = \int_0^t \operatorname{erf}\left(\frac{c}{\sqrt{\tau}}\right)d\tau - \int_0^t \operatorname{erf}\left(\frac{b}{\sqrt{\tau}}\right)\operatorname{erf}\left(\frac{c}{\sqrt{\tau}}\right)d\tau$$

$$= \int_0^t \operatorname{erf}\left(\frac{c}{\sqrt{\tau}}\right)d\tau - 2W_3(b,c,t)$$

where $W_3(b,c,t)$ is computed in Folder 10 in terms of I_1. If we let $\tau = 1/w^2$, we get

$$\int_0^t \operatorname{erf}\left(\frac{c}{\sqrt{\tau}}\right)d\tau = 2\int_T^\infty \frac{\operatorname{erf}(cw)}{w^3}dw, \qquad T = \frac{1}{\sqrt{t}},$$

and integration by parts gives

$u = \operatorname{erf}(cw)$ $\qquad$ $dv = dw/w^3$

$du = \dfrac{2c}{\sqrt{\pi}}e^{-c^2w^2}dw$ $\qquad$ $v = -1/2w^2$

$$\int_0^t \operatorname{erf}\left(\frac{c}{\sqrt{\tau}}\right)d\tau = \frac{\operatorname{erf}(cT)}{T^2} + \frac{2c}{\sqrt{\pi}}\int_T^\infty \frac{e^{-c^2w^2}}{w^2}dw = \frac{\operatorname{erf}(cT)}{T^2} + \frac{c}{T\sqrt{\pi}}E_{3/2}(c^2T^2)$$

and finally, writing $E_{3/2}$ in terms of $i$erfc, we get

$$I_{21}(0,b,c,t) = \frac{\operatorname{erf}(cT)}{T^2} + \frac{2c}{T}i\operatorname{erfc}(cT) - 2W_3(b,c,t)$$

where W_3 is computed in Folder 10a.

The following analysis shows that some losses of significance can be avoided by analytically combining terms whose leading digits tend to cancel out for some range of arguments. Thus, from Folder 10a,

$$2W_3(b,c,t) = \frac{\operatorname{erf}(bT)\operatorname{erf}(cT)}{T^2} + \frac{2}{\sqrt{\pi}}[bI_1(b,c,T) + cI_1(c,b,T)]$$

$$I_1(c,b,T) = \frac{e^{-c^2T^2}\operatorname{erf}(bT)}{T} - c\sqrt{\pi}\operatorname{erfc}(cT) + \frac{b}{\sqrt{\pi}}E_1(X) + 2c^2I_5(c,b,T)$$

where I_1 is obtained from Folder 10a and I_5 is the I function of Folder 5 and $X = (b^2+c^2)T^2$.

Then, with $\qquad i\operatorname{erfc}(x) = -x\operatorname{erfc}(x) + \dfrac{e^{-x^2}}{\sqrt{\pi}}$

we get

$$I_{21}(0,b,c,t) = \frac{\mathrm{erf}(cT)\,\mathrm{erfc}(bT)}{T^2} - \frac{2b}{\sqrt{\pi}} I_1(b,c,T) + \frac{2ce^{-c^2T^2}}{T\sqrt{\pi}} \mathrm{erfc}(bT) -$$

$$-\frac{2bc}{\pi} E_1(X) - \frac{4c^3}{\sqrt{\pi}} I_5(c,b,T)$$

The series for $0 \le a\sqrt{t} \le 1$ gives significant digit results for a = 0,

$$I_{21}(0,b,c,t) = 4t\ i^2 erfc\left(\frac{b}{\sqrt{t}}\right) erf\left(\frac{c}{\sqrt{t}}\right) + \frac{4c\sqrt{t}}{\sqrt{\pi}} e^{-X} y_2(c,b,T)\,,$$

when y_2 is computed from subroutine GNSEQ.

<u>a = 0 for $I_{21}^c(0,b,c,t)$</u>

For $I_{21}^c(0,b,c,t)$, we have

$$I_{21}^c(0,b,c,t) = \int_0^t \ \mathrm{erfc}\left(\frac{b}{\sqrt{\tau}}\right) \mathrm{erfc}\left(\frac{c}{\sqrt{\tau}}\right) d\tau = 2W_3^c(b,c,T), \quad \mathrm{T} = 1/\sqrt{\mathrm{t}}$$

where W_3^c is computed in Folder 10b. From above, we also have

$$I_{21}^c(0,b,c,t) = 4ti^2\ \mathrm{erfc}\left(\frac{b}{\sqrt{t}}\right) \mathrm{erfc}\left(\frac{c}{\sqrt{t}}\right) - \frac{4c\sqrt{t}e^{-X}}{\sqrt{\pi}} y_2(c,b,T)\,.$$

<u>a = 0 for $J_{21}(0,b,c,t)$</u>

$$J_{21}(0,b,c,t) = \int_0^t \ \mathrm{erfc}\left(\frac{b}{\sqrt{\tau}}\right) \frac{e^{-c^2/\tau}}{\tau^{3/2}} d\tau\,, \qquad \tau = \frac{1}{w^2}$$

$$= 2\int_T^\infty \ e^{-c^2w^2}\ \mathrm{erfc}(bw)dw = 2I_5(c,b,T)\,, \qquad T = \frac{1}{\sqrt{t}}\,.$$

<u>a = 0, c = 0 for $J_{21}(0,b,0,t)$</u>

$$J_{21}(0,b,0,t) = 2I_5(0,b,T) = 2\int_T^\infty \ \mathrm{erfc}(bw)dw = \frac{2}{b} i\,\mathrm{erfc}(bT)$$

<u>a = 0, b = 0 for $I_{21}(0,0,c,t)$</u>

$$I_{21}(0,0,c,t) = \int_0^t \ \mathrm{erf}\left(\frac{c}{\sqrt{\tau}}\right) d\tau = \frac{\mathrm{erf}(cT)}{T^2} + \frac{2c}{T} i\,\mathrm{erfc}(cT) \qquad \text{(see case for } a = 0\text{)}$$

<u>a = 0 for $S_1(0,b,c,t)$</u> (see also case for a = 0 for $J_{21}(0,b,c,t)$)

$$S_1(a,b,c,t) = \frac{\sqrt{\pi}}{2\sqrt{t}} J_{21}(a,b,c,t) = \frac{\sqrt{\pi}}{2\sqrt{t}} \int_0^t \ \frac{U(a,b,\tau)}{\tau^{3/2}} e^{-c^2/\tau} d\tau$$

$$U(0,b,\tau) = \operatorname{erfc}\left(\frac{b}{\sqrt{\tau}}\right)$$

$$S_1(0,b,c,t) = \frac{\sqrt{\pi}}{2\sqrt{t}}\int_0^t \frac{e^{-c^2/\tau}\operatorname{erfc}(b/\sqrt{\tau})}{\tau^{3/2}}d\tau\,, \qquad \tau = \frac{1}{w^2}\,,$$

$$= \frac{2\sqrt{\pi}}{2\sqrt{t}}\int_T^\infty e^{-c^2w^2}\operatorname{erfc}(bw)dw = \frac{\sqrt{\pi}}{\sqrt{t}}I_5(c,b,T)\,, \quad T = \frac{1}{\sqrt{t}}\,. \qquad .$$

<u>a = 0, b = 0 for</u> $S_1(0,0,c,t)$

From above,

$$S_1(0,0,c,t) = \frac{\sqrt{\pi}}{\sqrt{t}}I_5(c,0,T) = \sqrt{\frac{\pi}{t}}\int_T^\infty e^{-c^2w^2}dw = \frac{\pi}{2c}T\operatorname{erfc}(cT)\,, \qquad T = \frac{1}{\sqrt{t}}\,.$$

<u>b=0 for</u> $I_{21}(a,0,c,t)$

The expression,

$$I_{21}(a,b,c,t) = V(t)\operatorname{erf}\left(\frac{c}{\sqrt{t}}\right) + \frac{4c}{\sqrt{\pi}}\sum_{n=2}^{\infty}(-2a)^{n-2}\int_0^t e^{-c^2/\tau}i^n\operatorname{erfc}\left(\frac{b}{\sqrt{\tau}}\right)\tau^{(n-3)/2}d\tau$$

goes over to

$$I_{21}(a,0,c,t) = V(a,0,t)\operatorname{erf}\left(\frac{c}{\sqrt{t}}\right) + \frac{4c}{\sqrt{\pi}}\sum_{n=2}^{\infty}\frac{(-2a)^{n-2}}{2^n\Gamma(\frac{n}{2}+1)}\int_0^t e^{-c^2/\tau}\tau^{(n-3)/2}d\tau$$

$$= V(a,0,t)\operatorname{erf}\left(\frac{c}{\sqrt{t}}\right) + \frac{c\sqrt{t}}{\sqrt{\pi}}\sum_{n=0}^{\infty}\frac{(-a\sqrt{t})^n}{\Gamma(\frac{n}{2}+2)}E_{\frac{n+3}{2}}\left(\frac{c^2}{t}\right)$$

Notice also that

$$V(a,0,t) = 2\,\mathrm{I}_{23}(a\sqrt{t})/a^2 \quad , a \neq 0$$

$$\mathrm{I}_{23}(x) = (e^{x^2}erfc(x)-1)/2 + x/\sqrt{\pi} = x^2\sum_{n=0}^{\infty}\frac{(-x)^n}{\Gamma(\frac{n}{2}+1)(n+2)} = \frac{x^2}{2}\sum_{n=0}^{\infty}\frac{(-x)^n}{\Gamma(\frac{n}{2}+2)}$$

where I_{23} and related functions are found in Folder 23.

<u>Numerical Evaluation of</u> V(a,b,t)

In Folder10a, a discussion of the ramifications of computing ierfc(x) by

$$ierfc(x) = \frac{e^{-x^2}}{\sqrt{\pi}} - x\,erfc(x)$$

is presented. Specifically, losses of significance occur for larger values of x (2 digits for x as small as 7). It is explained that accurate results with errors $O(10^{-13})$ can be obtained by using subroutine DIERFC, which uses the asymptotic expansion of ierfc(x) for $x \geq 7$. Therefore the default form for computation of V(a,b,t) is

$$V(a,b,t) = \frac{2\sqrt{t}}{a} ierfc(bT) + \frac{1}{a^2}\left[U(a,b,t) - erfc(bT)\right].$$

The power series above is used for $a\sqrt{t} \leq 1$.

References

1. Amos, D.E., “Bounds on Iterated Coerror Functions and their Ratios,” *Mathematics of Computation*, Vol. 27, No. 122, April 1973.

2. Amos, D.E., “Evaluation of Some Cumulative Distribution Functions By Numerical Quadrature,” *SIAM Review*, Vol. 20, No. 4, October 1974.

3. Gautschi, W., “Computational Aspects of Three-Term Recurrence Relations,” *SIAM Review*, Vol. 9, No. 1, pp. 24-82, 1967.

4. Erdelyi, A., et al., *Tables of Integral Transforms*, Vol. 1, McGraw Hill, 1954.

5. Slater, L.J., “Confluent Hypergeometric Functions”, Cambridge Univ. Press, 1960

6. Abramowitz, M. and Stegun, I.A., *Handbook of Mathematical Functions*, National Bureau of Standards, AMS 55, 1964.

Folder 21A

APPENDIX

Description of Double Precision Subroutine DQUAD8

Infinte interval **(a,∞)**

DQUAD8 is designed to ensure that major contributions to integrals of the form

$$I = \int_a^{\infty} f(x)dx\,,\ \ f(x) \text{ monotone decreasing for } x \to \infty$$

are accounted for by proper selection of the scale of integration in a quadrature [2]. These scales can be small or large, depending on how fast $f(x)$ decreases for $x \to \infty$. DQUAD8 performs best when the dominant behavior of $f(x)$ is exponentially decreasing where the scale of integration is typically a small multiple of the standard deviation. However further success is also ensured by using a robust adaptive quadrature routine like DGAUS8 on functions with analytic integrands. The term “robust” means that a correct answer at the required accuracy is returned or failure is noted by an error flag. DGAUS8 is known to handle steep gradients and narrow spikes quite well and can be relied upon to return a proper result, whether it be an accurate answer with IERR = 1 or a failure to achieve that which was requested with IERR ≠ 1.

DQUAD8 computes I by summing quadratures over lengths σ,

$$I_K = \sum_{k=1}^{K} Q_k\,, \qquad\qquad Q_k = \int_{a+(k-1)\sigma}^{a+k\sigma} f(x)dx$$

where Q_k is computed by DGAUS8. The sum is terminated on a relative error test

$$\left| Q_K / I_K \right| < REL$$

where REL is a preassigned tolerance which is also used to terminate the quadrature in DGAUS8 for each Q_k, $k = 1, \ldots K$.

The call list for DQUAD8 is

```
CALL DQUAD8 (FOFX, INIT, X1, SIG, REL, X2, QUAD, IERR)
```

where the possibility for more than one call with no change in call list is allowed.

INPUT

FOFX = External double precision function for $f(x)$, $x \geq$ a

INIT = 0 on the first call to start the computation. On return, INIT = K and QUAD is the answer on ($X1$, $X2$) when IERR = 1

$X1$ = Lower limit of integration, a

SIG = Scale of integration, usually a measure of the standard deviation σ for the dominant part of $f(x)$

REL = Relative error tolerance used to terminate DQUAD8 and DGAUS8,
1.0D−3 .LE. REL .LE. 1.0D−14

OUTPUT

INIT = K after each call (K is incremented up if DQUAD8 is called more than once)

$X2$ = $X1+K\cdot$SIG = Endpoint of the last quadrature interval

QUAD = Double precision answer for I_K on ($X1$,$X2$)

IERR = 1 Normal return
= 5 Convergence of the integral not obtained in 20 steps of length SIG
= Any other value is an error flag from DGAUS8

Usually, $|Q_K / I_K|$ is a measure of the truncation error when $f(x)$ decreases rapidly.

Consequently, the termination condition

$$|Q_K / I_K| < REL$$

while not rigorous, represents a reasonable way to terminate the quadrature. A known error-estimate can be tested after succeeding calls to DQUAD8 with no change in parameters between calls. For example, a truncation error estimate B_{X2},

$$|T_{X2}| = \left|\int_{X2}^{\infty} f(x)dx\right| \leq B_{X2}, \qquad X2 > X1,$$

can be tested

$$|B_{X2} / I_K| < REL$$

and, if satisfied, QUAD can be accepted as the answer. If the truncation error test is not satisfied, then DQUAD8 can be called successively (in a loop) until the test is satisfied. In the case where no truncation error estimate is given, then DQUAD8 can be called more than once to give an added measure of assurance that the truncation error is small enough.

Since many functions $f(x)$ are dominated by exponentials, we give candidates for σ for two common exponentials:

$$e^{-ax^2}, \qquad \sigma = m/\sqrt{a}, \qquad m = 3 \text{ or } 4$$

$$e^{-ax}, \qquad \sigma = m/a \qquad m = 4 \text{ or } 5.$$

<u>Finite Interval</u> (a,b)

While the main thrust of DQUAD8 is to compute on an infinite interval, it was easy to modify the code to compute exactly MQ steps of length SIG. This capability allows one to compute MQ quadratures on an interval (A,B), by setting

```
INIT= – MQ
  X1=A
 SIG=(B-A)/DBLE(FLOAT(MQ))
```

DQUAD8 recognizes the negative value of INIT as the finite interval case and does MQ quadratures of length SIG which produces X2=B. The looping is expected to terminate in MQ steps so that convergence of the integral is not an issue and IERR=5 is not possible. Therefore, if IERR is not 1, it means that some quadrature failed and the prologue of DGAUS8 should be consulted to define the problem.

If subsequent calls are made with no change in INIT, the quadrature continues on as if the interval were infinite. This can be useful if one wants to refine a quadrature near the point A because of a steep gradient or singularity.

Actually, SIG can be changed between calls in both the infinite and finite cases and X2 will be computed correctly, but it will not be X1+K*SIG as reported above. However, INIT always returns the total number of steps taken after the initial setting.

Folder 22

Evaluation of

$$\mathbf{U(a,b,t) = e^{a^2t+2ab}erfc(a\sqrt{t} + b/\sqrt{t})}$$

$$I_{22}(a,b,c,t) = \int_0^t U(a,b,\tau)\frac{e^{-c^2/\tau}}{\sqrt{\tau}}d\tau \qquad a \ge 0, \quad b \ge 0,$$

$$J_{22}(a,b,c,t) = \int_0^t U(a,b,\tau)\sqrt{\tau}e^{-c^2/\tau}\, d\tau \qquad c \ge 0, \quad t > 0$$

Donald E. Amos, November 2002

Summary

The pattern of development follows that of Folder 21 where we use the formula

$$U(\tau) = \frac{2}{\sqrt{\pi}}\frac{e^{-b^2/\tau}}{\sqrt{\tau}}\int_0^\infty e^{-2(a+b/\tau)y}e^{-y^2/\tau}dy$$

to develop forms which can be evaluated efficiently by numerical quadrature or by series. The results are:

$$I_{22}(a,b,c,t) = \frac{1}{a\sqrt{\pi}}E_1(X) - \frac{2\sqrt{t}}{a\sqrt{\pi}}S_2(a,b,c,t)$$

$$J_{22}(a,b,c,t) = \frac{t}{a\sqrt{\pi}}E_2(X) - \frac{1}{a^2\sqrt{\pi}}\left(b+\frac{1}{2a}\right)E_1(X) + \frac{\sqrt{t}}{a^2}e^{-X}[e^{B^2}\ \mathrm{erfc}(B)]$$

$$-\frac{2c^2\sqrt{t}}{a^2\sqrt{\pi}}S_1(a,b,c,t) + \frac{\sqrt{t}}{a^3\sqrt{\pi}}S_2(a,b,c,t)$$

where $S_1(a,b,c,t)$ was defined in Folder 21,

$$S_1(a,b,c,t) = e^{-X}\int_0^\infty \frac{e^{-2Bw-w^2}}{c^2+(b+w\sqrt{t})^2}dw, \qquad B = a\sqrt{t} + b/\sqrt{t}$$

$$X = (b^2+c^2)/t$$

and

$$S_2(a,b,c,t) = e^{-X}\int_0^\infty \frac{e^{-2Bw-w^2}(b+w\sqrt{t})}{c^2+(b+w\sqrt{t})^2}\,dw.$$

Quadratures on these integrals are the default computation. We also have for $L = B/\sqrt{a^2t+c^2/t} \geq 2$,

$$S_1(a,b,c,t) = \frac{e^{-X}}{2Bt}\sum_{k=0}^{\infty}\frac{U_k(x)}{L^k}[e^{B^2}E_{(k+3)/2}(B^2)], \qquad B = a\sqrt{t}+b/\sqrt{t}$$

$$X = (b^2+c^2)/t$$

$$S_2(a,b,c,t) = \frac{e^{-X}}{2\sqrt{t}}\sum_{k=0}^{\infty}\frac{T_k(x)}{L^k}[e^{B^2}E_{(k+2)/2}(B^2)], \qquad x = \frac{a\sqrt{t}}{\sqrt{a^2t+c^2/t}}$$

where $T_k(x)$ and $U_k(x)$ are Chebyshev polynomials of the first and second kinds. Both polynomials can be generated by forward recurrence on their three-term recurrence relations. While each series converges for $L > 1$, we apply these series for $L \geq 2$ to obtain rapid convergence in a numerical evaluation.

The forms developed above have numerical problems for small a because of small differences of large numbers (an indeterminant form for $a \to 0$). Therefore we develop the power series for small a, $(a\sqrt{t} \leq 1)$,

$$I_{22}(a,b,c,t) = 2e^{-X}\left[b\sum_{n=0}^{\infty}(-2a\sqrt{t})^n y_{n+1} + \sqrt{t}\sum_{n=0}^{\infty}(n+2)(-2a\sqrt{t})^n y_{n+2}\right]$$

$$J_{22}(a,b,c,t) = 4\sqrt{t}e^{-X}\left[b^2\sum_{n=0}^{\infty}(-2a\sqrt{t})^n y_{n+2} + b\sqrt{t}\sum_{n=0}^{\infty}(2n+5)(-2a\sqrt{t})^n y_{n+3}\right.$$

$$\left.+t\sum_{n=0}^{\infty}(n+2)(n+4)(-2a\sqrt{t})^n y_{n+4}\right]$$

where

$$y_n = 2T^{n-1}e^X G_n(c,b,t) = 2T^{n-1}e^X\int_T^\infty e^{-c^2w^2}\frac{\mathrm{i}^n\mathrm{erfc}(bw)}{w^n}\,dw, \qquad T = 1/\sqrt{t}\,,$$

is computed in Folder 21. Special formulae are developed for cases where one or more parameters are zero. The default quadrature for S_1 and S_2 are described in the APPENDIX of Folder 21. Suitable truncation error bounds for S_1 and S_2 are supplied in Folders 21 and 22 for the quadrature.

Evaluation of $I_{22}(a,b,c,t)$ by Quadrature

$$I_{22}(a,b,c,t)=\int_0^t U(a,b,\tau)\frac{e^{-c^2/\tau}}{\sqrt{\tau}}d\tau$$

From Folder 4, we have

$$U(a,b,\tau)=e^{a^2\tau+2ab}\,\mathrm{erfc}(a\sqrt{\tau}+b/\sqrt{\tau})=\frac{2e^{-b^2/\tau}}{\sqrt{\pi\tau}}\int_0^\infty e^{-2(a+b/\tau)y}e^{-y^2/\tau}dy$$

and exchanging the order of integration gives

$$I_{22}(a,b,c,t)=\frac{2}{\sqrt{\pi}}\int_0^\infty e^{-2ay}\int_0^t\frac{e^{-[c^2+(y+b)^2]/\tau}}{\tau}d\tau \qquad \tau=t/w,$$

$$=\frac{2}{\sqrt{\pi}}\int_0^\infty e^{-2ay}E_1\{[c^2+(y+b)^2]/t\}dy$$

Integration by parts gives

$$u=E_1\{[c^2+(y+b)^2]/t\} \qquad dv=e^{-2ay}dy$$

$$du=\frac{-te^{-[c^2+(y+b)^2]/t}}{c^2+(y+b)^2}\cdot\frac{2(y+b)}{t}dy \qquad v=-\frac{1}{2a}e^{-2ay}$$

$$I_{22}(a,b,c,t)=\frac{2}{\sqrt{\pi}}\left[\frac{1}{2a}E_1(X)-\frac{1}{a}\int_0^\infty\frac{e^{-2ay-[c^2+(y+b)^2]/t}}{c^2+(y+b)^2}(y+b)dy\right]$$

$$=\frac{E_1(X)}{a\sqrt{\pi}}-\frac{2e^{2ab}}{a\sqrt{\pi}}\int_b^\infty e^{-2ax}e^{-(c^2+x^2)/t}\cdot\frac{x}{c^2+x^2}dx$$

under the substitution $y+b=x$ with $X=(b^2+c^2)/t$. Now, let $w=(x+at)/\sqrt{t}$ and note that

$$2ax+(c^2+x^2)/t=\frac{c^2}{t}+\frac{(x^2+2axt+a^2t^2-a^2t^2)}{t}=\frac{c^2}{t}+\frac{(x+at)^2}{t}-\frac{a^2t^2}{t}=\frac{c^2}{t}+w^2-a^2t.$$

Then, with proper scaling, we have

$$I_{22}(a,b,c,t)=\frac{e^{-X}}{a\sqrt{\pi}}[e^XE_1(X)]-\frac{2\sqrt{t}}{a\sqrt{\pi}}S_2(a,b,c,t), \qquad X=(b^2+c^2)/t,$$

where

$$S_2(a,b,c,t) = e^{a^2 t + 2ab - c^2/t} \sqrt{t} \int_B^\infty \frac{e^{-w^2}(w - a\sqrt{t})}{c^2 + t(w - a\sqrt{t})^2} dw\,, \qquad B = a\sqrt{t} + b/\sqrt{t}\,,$$

or, with the substitution $w = B+v$, we get the computational form

$$S_2(a,b,c,t) = e^{-X} \int_0^\infty \frac{e^{-2Bv - v^2}(b + v\sqrt{t})}{c^2 + (b + v\sqrt{t})^2} dv\,, \qquad X = (b^2+c^2)/t.$$

Notice that X can be tested for underflow and the integrand contains no subtractions which can lose significant digits. We follow the format of Folder 21 and estimate a truncation error for the quadrature procedure described in the APPENDIX of Folder 21 using subroutine DQUAD8. The truncation error of this integral at $v = R$ is

$$T_R = \int_R^\infty \frac{e^{-2Bv - v^2}(b + v\sqrt{t})}{c^2 + (b + v\sqrt{t})^2} dv\,, \qquad R>0$$

and

$$|T_R| \le e^{-2BR} \int_R^\infty \frac{e^{-v^2} dv}{\sqrt{c^2 + (b + v\sqrt{t})^2}} < \frac{e^{-2BR}}{\sqrt{c^2 + (b + R\sqrt{t})^2}} \int_R^\infty e^{-v^2} dv < \frac{(\sqrt{\pi}/2) e^{-2BR} \operatorname{erfc}(R)}{\sqrt{c^2 + (b + R\sqrt{t})^2}}$$

since

$$\frac{x}{A^2 + x^2} = \frac{x}{\sqrt{A^2 + x^2}} \cdot \frac{1}{\sqrt{A^2 + x^2}} \le \frac{1}{\sqrt{A^2 + x^2}}\,, \qquad A > 0, \quad x \ge 0.$$

From Mill’s ratio, [A&S, 7.1.13],

$$\frac{\sqrt{\pi}}{2} \operatorname{erfc}(R) \le \frac{e^{-R^2}}{R + \sqrt{R^2 + 4/\pi}} < \frac{e^{-R^2}}{2R}$$

we get

$$|T_R| < \frac{e^{-2BR - R^2}}{2R\sqrt{c^2 + (b + R\sqrt{t})^2}} = B_R\,, \qquad R > 0$$

As in Folder 21, we require an estimate of the scale of integration, σ, for DQUAD8. The exponentials give

$$\text{SIG} = m \cdot \min\left[\frac{1}{2B}, \frac{1}{\sqrt{2}}\right] \text{ where } m = 3 \text{ or } 4.$$

Evaluation of $I_{22}(a,b,c,t)$ and $S_2(a,b,c,t)$ by Series

Notice that

$$c^2 + t(w - a\sqrt{t})^2 = t(w^2 - 2a\sqrt{t}w + a^2 t + c^2/t)$$

and

$$P \equiv \int_B^\infty \frac{e^{-w^2}(w - a\sqrt{t})}{c^2 + t(w - a\sqrt{t})^2}\,dw = \frac{1}{t}\int_B^\infty \frac{e^{-w^2}(w - a\sqrt{t})dw}{w^2\left[1 - \frac{2a\sqrt{t}}{w} + \frac{a^2 t + c^2/t}{w^2}\right]}$$

Let $w = v\sqrt{a^2 t + c^2/t}$, $\qquad L = B/\sqrt{a^2 t + c^2/t}$, $\qquad x = a\sqrt{t}/\sqrt{a^2 t + c^2/t}$. Then

$$P = \frac{1}{t}\left[\int_L^\infty \frac{e^{-(a^2t + c^2/t)v^2}}{v} D(v)dv - x\int_L^\infty \frac{e^{-(a^2t + c^2/t)v^2}}{v^2} D(v)dv\right]$$

where

$$D(v) = \left[1 - \frac{2x}{v} + \frac{1}{v^2}\right]^{-1}.$$

Now the generating function for Chebyshev polynomials of the second kind is

$$\frac{1}{1 - 2xz + z^2} = \sum_{k=0}^\infty U_k(x) z^k \qquad |x| < 1, \quad |z| < 1$$

and

$$D(v) = \sum_{k=0}^\infty \frac{U_k(x)}{v^k}, \qquad x = \frac{a\sqrt{t}}{\sqrt{a^2 t + c^2/t}}, \qquad L = \frac{B}{\sqrt{a^2 t + c^2/t}}.$$

Then

$$P = \frac{1}{t}\left[\sum_{k=0}^\infty U_k(x)\int_L^\infty \frac{e^{-(a^2t + c^2/t)v^2}}{v^{k+1}}dv - x\sum_{k=0}^\infty U_k(x)\int_L^\infty \frac{e^{-(a^2t + c^2/t)v^2}}{v^{k+2}}dv\right]$$

and with $v = L\sqrt{w}$ we have

$$P = \frac{1}{2t}\left[\sum_{k=0}^{\infty} \frac{U_k(x)}{L^k} E_{(k+2)/2}(B^2) - \frac{x}{L}\sum_{k=0}^{\infty} \frac{U_k(x)}{L^k} E_{(k+3)/2}(B^2)\right]$$

If we replace k by $k-1$ in the second sum and sum from $k = 1$ to infinity, we can use the identity

$$T_k(x) = U_k(x) - xU_{k-1}(x), \qquad k \geq 1,$$

To get

$$P = \frac{1}{2t}\left[E_1(B^2) + \sum_{k=1}^{\infty} \frac{T_k(x)}{L^k} E_{(k+2)/2}(B^2)\right], \qquad x = \frac{a\sqrt{t}}{\sqrt{a^2 t + c^2/t}}$$

and

$$S_2(a,b,c,t) = \frac{e^{-X}}{2\sqrt{t}} \sum_{k=0}^{\infty} \frac{T_k(x)}{L^k}[e^{B^2} E_{(k+2)/2}(B^2)], \qquad B = a\sqrt{t} + b/\sqrt{t}\,,\ L = \frac{B}{\sqrt{a^2 t + c^2/t}}$$

in computational form since $e^x E_v(x)$ is well scaled and an underflow test can be made on X.

Then,

$$I_{22}(a,b,c,t) = e^{a^2 t + 2ab - c^2/t}\sqrt{t}\ P = \frac{e^{-X}}{a\sqrt{\pi}}[e^X E_1(X)] - \frac{2\sqrt{t}}{a\sqrt{\pi}} S_2(a,b,c,t)\,, \qquad X = (b^2+c^2)/t.$$

Here, $T_k(x)$ and $U_k(x)$ are Chebyshev polynomials of the first and second kids and can be generated numerically by forward recurrence on their three-term recurrence relations,

$$T_o(x) = 1, \qquad T_1(x) = x, \qquad T_{k+1}(x) = 2xT_k(x) - T_{k-1}(x), \qquad k \geq 1$$

$$U_o(x) = 1, \qquad U_1(x) = 2x, \qquad U_{k+1}(x) = 2xU_k(x) - U_{k-1}(x), \qquad k \geq 1$$

Notice that $E_{(k+2)/2}(B^2)$ contains both integer and half odd integer orders which can be generated, with the scaling from subroutines DEXINT and DHEXINT. For rapid convergence we take $L \geq 2$.

Evaluation of J_{22} (a,b,c,t) by Quadrature

Notice that the integrand of $J_{22}(a,b,c,t)$ differs from that of $I_{22}(a,b,c,t)$ by a factor of τ. Thus, we have from our basic formula

$$J_{22}(a,b,c,t) = \frac{2}{\sqrt{\pi}}\int_0^\infty e^{-2ay}\int_0^t e^{-[c^2+(y+b)^2]/\tau}d\tau, \qquad \tau = \frac{t}{w},$$

$$= \frac{2t}{\sqrt{\pi}}\int_0^\infty e^{-2ay}E_2\{[c^2+(y+b)^2]/t\}dy$$

Integration by parts gives

$$u = E_2\{[c^2+(y+b)^2]/t\} \qquad dv = e^{-2ay}dy$$

$$du = -\frac{2(y+b)}{t}E_1\{[c^2+(y+b)^2]/t\}dy \qquad v = -\frac{1}{2a}e^{-2ay}$$

Then, with $X = (b^2+c^2)/t$,

$$J_{22}(a,b,c,t) = \frac{2t}{\sqrt{\pi}}\left[\frac{E_2(X)}{2a} - \frac{1}{at}\int_0^\infty e^{-2ay}(y+b)E_1\{[c^2+(y+b)^2]/t\}dy\right]$$

Let $y+b = x$. Then

$$J_{22}(a,b,c,t) = \frac{2t}{\sqrt{\pi}}\left[\frac{E_2(X)}{2a} - \frac{e^{2ab}}{at}\int_b^\infty e^{-2ax}xE_1[(c^2+x^2)/t]dx\right].$$

Integration by parts again gives

$$u = E_1[(c^2+x^2)/t] \qquad dv = e^{-2ax}\cdot xdx$$

$$du = -\frac{te^{-(c^2+x^2)/t}}{c^2+x^2}\cdot\frac{2x}{t}dx \qquad v = -\frac{xe^{-2ax}}{2a} - \frac{e^{-2ax}}{4a^2}$$

$$F \equiv \int_b^\infty e^{-2ax}xE_1[(c^2+x^2)/t]dx = \frac{e^{-2ab}}{2a}\left(b+\frac{1}{2a}\right)E_1(X) - \int_b^\infty e^{-2ax-(c^2+x^2)/t}\frac{x}{c^2+x^2}\left(\frac{x}{a}+\frac{1}{2a^2}\right)dx$$

where $X = (b^2+c^2)/t$. Then,

$$F = \frac{e^{-2ab}}{2a}\left(b+\frac{1}{2a}\right)E_1(X) - \frac{1}{a}\int_b^\infty e^{-2ax-(c^2+x^2)/t}\left(1-\frac{c^2}{x^2+c^2}\right)dx$$

$$-\frac{1}{2a^2}\int_b^\infty e^{-2ax-(c^2+x^2)/t}\frac{x}{x^2+c^2}dx\,.$$

As before,

$$2ax+(c^2+x^2)/t = \frac{c^2}{t}+w^2-a^2t\,, \qquad\qquad w=(x+at)/\sqrt{t}\,,$$

and

$$F = \frac{e^{-2ab}}{2a}\left(b+\frac{1}{2a}\right)E_1(X) - \frac{\sqrt{t}e^{a^2t-c^2/t}}{a}\int_B^\infty e^{-w^2}dw + \frac{c^2\sqrt{t}e^{a^2t-c^2/t}}{a}\int_B^\infty \frac{e^{-w^2}dw}{c^2+t(w-a\sqrt{t})^2}$$

$$-\frac{te^{a^2t-c^2/t}}{2a^2}\int_B^\infty \frac{e^{-w^2}(w-a\sqrt{t})}{c^2+t(w-a\sqrt{t})^2}dw.$$

Then,

$$J_{22}(a,b,c,t) = \frac{te^{-X}}{a\sqrt{\pi}}[e^X E_2(X)] + \frac{\sqrt{t}}{a^2}e^{-X}[e^{B^2}\ \mathrm{erfc}(B)] - \frac{1}{a^2\sqrt{\pi}}\left(b+\frac{1}{2a}\right)e^{-X}[e^X E_1(X)]$$

$$-\frac{2c^2\sqrt{t}}{a^2\sqrt{\pi}}S_1(a,b,c,t) + \frac{\sqrt{t}}{a^3\sqrt{\pi}}S_2(a,b,c,t)$$

$$X=(b^2+c^2)/t, \qquad\qquad B=a\sqrt{t}+b/\sqrt{t}$$

where S_1 (Folder 21) and S_2 were defined previously for a quadrature or series evaluation. The quantities in brackets are well scaled and can be computed directly from DRERF and DEXINT codes. Notice that all terms can be tested for underflow on X. $S_1(a,b,c,t)$ was derived in Folder 21 and has the computational form (by quadrature)

$$S_1(a,b,c,t) = e^{-X}\int_0^\infty \frac{e^{-2Bv-v^2}}{c^2+(b+v\sqrt{t})^2}dv, \qquad B=a\sqrt{t}+b/\sqrt{t}, \qquad X=(b^2+c^2)/t.$$

Folder 21 also shows the power series in reciprocal powers of $L \geq 2$.

Evaluation of $I_{22}(a,b,c,t)$ and $J_{22}(a,b,c,t)$ for $a\sqrt{t}\ \le 1$

From Folder 21, we have the series for

$$U(a,b,t) = e^{a^2t+2ab}\ \mathrm{erfc}(a\sqrt{t} + b/\sqrt{t}) = \sum_{n=0}^{\infty}\ i^n\ \mathrm{erfc}\left(\frac{b}{\sqrt{t}}\right)(-2a\sqrt{t})^n$$

Then,

$$I_{22}(a,b,c,t) = \sum_{n=0}^{\infty}\ (-2a)^n \int_0^t\ \mathrm{i}^{\mathrm{n}}\mathrm{erfc}\left(\frac{b}{\sqrt{\tau}}\right)(\sqrt{\tau})^{n-1} e^{-c^2/\tau} d\tau$$

and with $\tau = 1/w^2$, we get

$$I_{22}(a,b,c,t) = 2\sum_{n=0}^{\infty}\ (-2a)^n \int_T^{\infty}\ e^{-c^2w^2}\ \frac{\mathrm{i}^{\mathrm{n}}\mathrm{erfc}(bw)}{w^{n+2}} dw, \qquad\qquad T = \frac{1}{\sqrt{t}},$$

the recurrence

$$i^n\mathrm{erfc}(bw) = 2bwi^{n+1}\mathrm{erfc}(bw)+2(n+2)i^{n+2}\mathrm{erfc}(bw)$$

gives

$$\int_T^{\infty}\ e^{-c^2w^2}\ \frac{\mathrm{i}^{\mathrm{n}}\mathrm{erfc}(bw)}{w^{n+2}} dw = 2bG_{n+1}(c,b,T) + 2(n+2)G_{n+2}(c,b,T)$$

where $G_n(c,b,T)$ was defined in Folder 21 as

$$G_n(c,b,T) = \int_T^{\infty}\ e^{-c^2w^2}\ \frac{\mathrm{i}^{\mathrm{n}}\mathrm{erfc}(bw)}{w^n} dw$$

and the computational form is a scaling of G_n,

$$y_n = 2T^{n-1}e^X G_n(c,b,T), \qquad\qquad X = (b^2+c^2)/t,$$

which has a recurrence. (See Folders 21 and subroutine GNSEQ for details). Then,

$$I_{22}(a,b,c,t) = \frac{2e^{-X}}{T}\sum_{n=0}^{\infty}\ \frac{(-2a)^n}{T^n}[bTy_{n+1}(c,b,T) + (n+2)y_{n+2}(c,b,T)]$$

$$= 2e^{-X}\left[b\sum_{n=0}^{\infty}\ (-2a\sqrt{t})^n\, y_{n+1} + \sqrt{t}\sum_{n=0}^{\infty}\ (n+2)(-2a\sqrt{t})^n\, y_{n+2}\right]$$

The y_n sequence is generated in subroutine GNSEQ.

Also, for $J_{22}(a,b,c,t)$ we have

$$J_{22}(a,b,c,t) = \sum_{n=0}^{\infty} (-2a)^n \int_0^t \mathrm{i}^n \mathrm{erfc}\left(\frac{b}{\sqrt{\tau}}\right)(\sqrt{\tau})^{n+1} e^{-c^2/\tau} d\tau, \qquad \tau = \frac{1}{w^2},$$

$$= 2\sum_{n=0}^{\infty} (-2a)^n \int_T^{\infty} e^{-c^2 w^2} \frac{\mathrm{i}^n \mathrm{erfc}(bw)}{w^{n+4}} dw$$

We use the recurrence again:

$$i^n \mathrm{erfc}(bw) = 2bwi^{n+1}\mathrm{erfc}(bw) + 2(n+2)i^{n+2}\mathrm{erfc}(bw)$$

Now we repeat this relation for indices $n+1$ and $n+2$ and make the substitutions to get

$$\begin{aligned} i^n \mathrm{erfc}(bw) &= 2bw[2bwi^{n+2}\mathrm{erfc}(bw) + 2(n+3)i^{n+3}\mathrm{erfc}(bw)] \\ &\quad + 2(n+2)[2bwi^{n+3}\mathrm{erfc}(bw) + 2(n+4)i^{n+4}\mathrm{erfc}(bw)] \\ &= 4b^2w^2i^{n+2}\mathrm{erfc}(bw) + 4bw(2n+5)i^{n+3}\mathrm{erfc}(bw) \\ &\quad + 4(n+2)(n+4)i^{n+4}\mathrm{erfc}(bw) \end{aligned}$$

and

$$\int_T^{\infty} e^{-c^2w^2} \frac{i^n \mathrm{erfc}(bw)}{w^{n+4}} dw = 4b^2 G_{n+2}(c,b,T) + 4b(2n+5)G_{n+3}(c,b,T) + 4(n+2)(n+4)G_{n+4}(c,b,T)$$

Scaling the G's to get the y's gives the computational form (computed in subroutine GNSEQ)

$$\int_T^{\infty} e^{-c^2w^2} \frac{\mathrm{i}^n \mathrm{erfc}(bw)}{w^{n+4}} dw = \frac{2}{T^{n+3}}[b^2T^2 y_{n+2} + bT(2n+5)y_{n+3} + (n+2)(n+4)y_{n+4}]e^{-X}$$

and

$$J_{22}(a,b,c,t) = 4\sqrt{t}e^{-X}\left[b^2 \sum_{n=0}^{\infty} (-2a\sqrt{t})^n y_{n+2} + b\sqrt{t}\sum_{n=0}^{\infty} (2n+5)(-2a\sqrt{t})^n y_{n+3} + t\sum_{n=0}^{\infty} (n+2)(n+4)(-2a\sqrt{t})^n y_{n+4}\right]$$

The convergence of series of these forms is described in Folder 21.

Special Cases

$\underline{a = 0 \text{ for } I_{22}(0,b,c,t)}$ $\quad = \int_0^t \operatorname{erfc}\left(\frac{b}{\sqrt{\tau}}\right)\frac{e^{-c^2/\tau}}{\sqrt{\tau}}\,d\tau\,, \qquad \tau = \frac{1}{w^2}\,,$

$$= 2\int_T^\infty \frac{e^{-c^2w^2}}{w^2}\,dw - 2\int_T^\infty \frac{\operatorname{erf}(bw)}{w^2}e^{-c^2w^2}\,dw$$

$$= \frac{1}{T}E_{3/2}(c^2T^2) - 2I_1(c,b,T)$$

where $T = 1/\sqrt{t}$ and I_1 is computed in Folder 10. The Formulas

$$E_{3/2}(x^2) = 2\sqrt{\pi}\ \operatorname{ierfc}(x) = 2\sqrt{\pi}\left[-x\operatorname{erfc}(x) + \frac{e^{-x^2}}{\sqrt{\pi}}\right]$$

$$2I_1(c,b,T) = 2e^{-c^2/t}\sqrt{t}\operatorname{erf}(b/\sqrt{t}) + \frac{2b}{\sqrt{\pi}}E_1(X) - 2c\sqrt{\pi}\operatorname{erfc}(c/\sqrt{t}) + 4c^2 I_5(c,b,T)\,,$$

$$X = (b^2+c^2)/t, \qquad T = \frac{1}{\sqrt{t}}$$

give a form more suitable for numerical evaluation

$$I_{22}(0,b,c,t) = 2\sqrt{t}e^{-c^2/t}\operatorname{erfc}(b/\sqrt{t}) - \frac{2b}{\sqrt{\pi}}E_1(X) - 4c^2 I_5(c,b,T)$$

where I_5 is the I function of Folder 5. Notice also that the series for small a at $a = 0$ gives

$$I_{22}(0,b,c,t) = 2e^{-X}[by_1(c,b,T) + 2\sqrt{t}\,y_2(c,b,T)]$$

$\underline{a = 0 \text{ for } J_{22}(0,b,c,t)}$ $\quad = \int_0^t \operatorname{erfc}\left(\frac{b}{\sqrt{\tau}}\right)\sqrt{\tau}e^{-c^2/\tau}\,d\tau\,, \qquad \tau = \frac{1}{w^2}$

$$= 2\int_T^\infty \frac{\operatorname{erfc}(bw)}{w^4}e^{-c^2w^2}\,dw, \qquad T = \frac{1}{\sqrt{t}}\,.$$

Integration by parts gives

$$u = e^{-c^2w^2}\ \mathrm{erfc}(bw) \qquad\qquad dv = dw/w^4$$

$$du = \left[-\frac{2b}{\sqrt{\pi}} e^{-(b^2+c^2)w^2} - 2c^2 w e^{-c^2w^2}\ \mathrm{erfc}(bw)\right] dw \qquad\qquad v = -\frac{1}{3w^3}$$

Then

$$J_{22}(0,b,c,t) = 2\left[\frac{e^{-c^2T^2}\ \mathrm{erfc}(bT)}{3T^3} - \frac{2b}{3\sqrt{\pi}}\int_T^\infty \frac{e^{-(b^2+c^2)w^2}}{w^3} dw - \frac{2c^2}{3}\int_T^\infty \frac{e^{-c^2w^2}\ \mathrm{erfc}(bw)}{w^2} dw\right]$$

$$= 2\left[\frac{e^{-c^2T^2}\ \mathrm{erfc}(bT)}{3T^3} - \frac{b}{3T^2\sqrt{\pi}} E_2(X) - \frac{2c^2}{3} I_1^c(c,b,T)\right]$$

where I_1^c is defined and computed in Folder 10b,

$$I_1^c(c,b,T) = \frac{\sqrt{\pi}}{T} i\,\mathrm{erfc}(cT) - I_1(c,b,T)$$

in terms of I_1 of Folder 10a,

$$I_1(c,b,T) = \frac{e^{-c^2T^2}\ \mathrm{erf}(bT)}{T} + \frac{b}{\sqrt{\pi}} E_1(X) - c\sqrt{\pi}\ \mathrm{erfc}(cT) + 2c^2 I_5(c,b,T)$$

where I_5 is the I function of Folder 5. Then, with

$$\mathrm{ierfc}(cT) = -cT\ \mathrm{erfc}(cT) + \frac{e^{-c^2T^2}}{\sqrt{\pi}}$$

we get

$$J_{22}(0,b,c,t) = 2\left[\frac{e^{-c^2T^2}\ \mathrm{erfc}(bT)}{3T^3} - \frac{b}{3T^2\sqrt{\pi}} E_2(X) - \frac{2c^2}{3T} e^{-c^2T^2}\ \mathrm{erfc}(bT)\right.$$

$$\left. + \frac{2bc^2}{3\sqrt{\pi}} E_1(X) + \frac{4c^4}{3} I_5(c,b,T)\right], \quad X = (b^2+c^2)T^2, \qquad \mathrm{T}=1/\sqrt{\mathrm{t}}\ .$$

$$J_{22}(0,b,c,t) = \frac{2}{3}\left[\sqrt{t}(t-2c^2)e^{-c^2/t}\ \mathrm{erfc}(bT) - \frac{b}{T^2\sqrt{\pi}}E_2(X) + \frac{2bc^2}{\sqrt{\pi}}E_1(X)\right.$$

$$\left. + 4c^4 I_5(c,b,T)\ \right] \qquad X = (b^2+c^2)/t, \qquad T = \frac{1}{\sqrt{t}}$$

We also have from the small a expression for $a = 0$,

$$J_{22}(0,b,c,t) = 4\sqrt{t}e^{-X}[b^2 y_2(c,b,T) + 5b\sqrt{t}y_3(c,b,T) + 8ty_4(c,b,T)]$$

<u>$a = 0,\ b = 0$ for $I_{22}(0,0,c,t)$</u> $\quad = \int_0^t \frac{e^{-c^2/\tau}d\tau}{\sqrt{\tau}}, \qquad \tau = \frac{t}{w}$

$$= \sqrt{t}E_{3/2}(c^2T^2) \qquad T = \frac{1}{\sqrt{t}}$$

$$= 2\sqrt{\pi t}\ \mathrm{ierfc}(cT)$$

<u>$a = 0,\ b = 0$ for $J_{22}(0,0,c,t)$</u> $\quad = \int_0^t \sqrt{\tau}e^{-c^2/\tau}d\tau, \qquad \tau = \frac{t}{w}$

$$= t^{3/2}E_{5/2}(c^2T^2), \qquad T = \frac{1}{\sqrt{t}}.$$

<u>$a = 0,\ c = 0$ for $I_{22}(0,b,0,t)$</u> $\quad = \int_0^t \frac{\mathrm{erfc}(b/\sqrt{\tau})}{\sqrt{\tau}}\,d\tau \qquad \tau = \frac{1}{w^2}$

$$= 2\int_T^\infty \frac{\mathrm{erfc}(bw)}{w^2}dw, \qquad T = \frac{1}{\sqrt{t}}$$

$$= 2I_1^c(0,b,T) = 2\left[\frac{1}{T} - I_1(0,b,T)\right]$$

$$= \frac{2}{T} - \frac{2}{T}\mathrm{erf}(bT) - \frac{2b}{\sqrt{\pi}}E_1(b^2T^2)$$

$$= \frac{2\,\mathrm{erfc}(bT)}{T} - \frac{2b}{\sqrt{\pi}}E_1(b^2T^2)$$

$\underline{a = 0,\ c = 0 \text{ for } J_{22}(0,b,0,t)}$ $\quad = \int_0^t \operatorname{erfc}(b/\sqrt{\tau})\sqrt{\tau}\, d\tau\,, \qquad \tau = \dfrac{1}{w^2}$

$$= 2\int_T^\infty \frac{\operatorname{erfc}(bw)}{w^4}\,dw$$

Integrate by parts

$$u = \operatorname{erfc}(bw) \qquad\qquad dv = dw/w^4$$

$$du = -\frac{2b}{\sqrt{\pi}}e^{-b^2w^2}dw \qquad\qquad v = -\frac{1}{3w^3}$$

$$\int_T^\infty \frac{\operatorname{erfc}(bw)}{w^4}dw = \frac{\operatorname{erfc}(bT)}{3T^3} - \frac{2b}{3\sqrt{\pi}}\int_T^\infty \frac{e^{-b^2w^2}}{w^3}dw \qquad\qquad w = T\sqrt{v}\,,$$

$$= \frac{\operatorname{erfc}(bT)}{3T^3} - \frac{b}{3T^2\sqrt{\pi}}E_2(b^2T^2)$$

$$J_{22}(0,b,0,t) = \frac{2\operatorname{erfc}(bT)}{3T^3} - \frac{2b}{3T^2\sqrt{\pi}}E_2(b^2T^2)\,, \qquad\qquad T = \frac{1}{\sqrt{t}}\,.$$

$\underline{a = 0 \text{ for } S_1(0,b,c,t)}$ $\quad = \sqrt{\dfrac{\pi}{t}}I_5(c,b,T)\,, \qquad\qquad T = \dfrac{1}{\sqrt{t}}$

from Folder 21 where I_5 is the I function of Folder 5.

$\underline{a = 0 \text{ for } S_2(0,b,c,t)}$ The relation

$$I_{22}(a,b,c,t) = \frac{1}{a\sqrt{\pi}}E_1(X) - \frac{2\sqrt{t}}{a\sqrt{\pi}}S_2(a,b,c,t)$$

was derived. In the form

$$\lim_{a\to 0} aI_{22}(a,b,c,t) = \frac{1}{\sqrt{\pi}}E_1(X) - \frac{2\sqrt{t}}{\sqrt{\pi}}S_2(0,b,c,t) = 0$$

we see that

$$S_2(0,b,c,t) = \frac{1}{2\sqrt{t}}E_1(X)\,, \qquad\qquad X = (b^2+c^2)/t.$$

<u>$b = 0$, $c = 0$ for $I_{22}(a,0,0,t)$</u> $= \int_0^t e^{a^2\tau}\,\mathrm{erfc}(a\sqrt{\tau})\frac{d\tau}{\sqrt{\tau}}$, $\qquad \tau = w^2$,

$$= 2\int_0^{\sqrt{t}} e^{a^2w^2}\,\mathrm{erfc}(aw)dw$$

$$= \frac{2}{a}\int_0^{a\sqrt{t}} e^{v^2}\,\mathrm{erfc}(v)dv$$

The computational form of this integral is developed in Folder 23 as function $H_{23}(x)$ with a corresponding double precision subroutine DHERFC(X). Then,

$$I_{22}(a,0,0,t) = \frac{2}{a}H_{23}(a\sqrt{t})$$

<u>$b = 0$, $c = 0$ for $J_{22}(a,0,0,t)$</u> $= \int_0^t e^{a^2\tau}\,\mathrm{erfc}(a\sqrt{\tau})\sqrt{\tau}d\tau$, $\qquad \tau = w^2$,

$$= 2\int_0^{\sqrt{t}} e^{a^2w^2}\,\mathrm{erfc}(aw)w^2dw$$

$$= \frac{2}{a^3}\int_0^{a\sqrt{t}} e^{v^2}\,\mathrm{erfc}(v)v^2dv$$

As with the previous case, this integral is developed in Folder 23 as $J_{23}(x)$ and

$$J_{22}(a,0,0,t) = \frac{2}{a^3}J_{23}(a\sqrt{t}).$$

<u>$b = 0$ *for* $I_{22}(a,0,c,t)$</u>

The expression

$$I_{22}(a,b,c,t) = \sum_{n=0}^{\infty} (-2a)^n \int_0^t \mathrm{i}^n\mathrm{erfc}\left(\frac{b}{\sqrt{\tau}}\right)(\sqrt{\tau})^{n-1}e^{-c^2/\tau}d\tau$$

goes over to

$$I_{22}(a,0,c,t) = \sum_{n=0}^{\infty} \frac{(-2a)^n}{2^n\,\Gamma(\frac{n}{2}+1)}\int_0^t (\sqrt{\tau})^{n-1}e^{-c^2/\tau}d\tau$$

$$= \sqrt{\mathrm{t}}\sum_{n=0}^{\infty} \frac{(-a\sqrt{t})^n}{\Gamma(\frac{n}{2}+1)}E_{\frac{n+3}{2}}(\frac{c^2}{t})$$

Asymptotic Expansion For $\mathbf{I_{22}(a,0,c,t)}$ For Large a

We start with

$$I_{22}(a,0,c,t)=\int_0^t \frac{e^{a^2\tau}erfc(a\sqrt{\tau})}{\sqrt{\tau}}e^{-c^2/\tau}d\tau\,.$$

It is tempting to replace the erfc function with its asymptotic expansion for large argument, but this can be questioned since τ can be as small as desired and the argument is not large for all τ. We change the variable $\tau=t/v^2$ to obtain

$$I_{22}(a,0,c,t)=2\sqrt{t}\int_1^\infty \frac{e^{a^2t/v^2}erfc(a\sqrt{t}/v)}{v^2}e^{-c^2v^2/t}dv\,.$$

This has shifted the problem from the origin to infinity, but we have a more familiar form for the exponential term. We apply the asymptotic form for the error function with a remainder and show that the new remainder after integration goes to zero as the product ac goes to infinity. Thus

$$e^{x^2}erfc(x)=\frac{1}{x\sqrt{\pi}}\sum_{k=0}^{N}\frac{(-1)^k(1/2)_k}{x^{2k}}+R_N(x)\quad where\quad |R_N|\le\frac{1}{x\sqrt{\pi}}\frac{(1/2)_{N+1}}{x^{2N+2}}\,.$$

Then,

$$I_{22}(a,0,c,t)=\frac{2}{a\sqrt{\pi}}\sum_{k=0}^{N}\frac{(-1)^k(1/2)_k}{(a\sqrt{t})^{2k}}\int_1^\infty e^{-c^2v^2/t}v^{2k-1}dv+G_N\,,\quad where\ G_N=2\sqrt{t}\int_1^\infty\frac{e^{-c^2v^2/t}}{v^2}R_N(a\sqrt{t}/v)dv$$

and with the change of variables, $v=\frac{\sqrt{t}}{c}\sqrt{w}$, we get

$$I_{22}(a,0,c,t)=\frac{1}{a\sqrt{\pi}}\sum_{k=0}^{N}\frac{(-1)^k(1/2)_k}{(ac)^{2k}}\int_{c^2/t}^\infty e^{-w}w^{k-1}dw+G_N$$

or

$$I_{22}(a,0,c,t)=\frac{1}{a\sqrt{\pi}}\sum_{k=0}^{N}\frac{(-1)^k(1/2)_k}{(a^2c^2)^k}\Gamma(k,\frac{c^2}{t})+G_N$$

where Γ is the incomplete gamma function. Now we analyze G_N and show that, like the remainder R_N, G_N is bounded by the magnitude of the next term of the sum.

$$\left|G_N\right|=\left|2\sqrt{t}\int_1^\infty\frac{e^{-c^2v^2/t}}{v^2}R_N(a\sqrt{t}/v)dv\right|\le 2\sqrt{t}\int_1^\infty\left|\frac{e^{-c^2v^2/t}}{v^2}R_N(a\sqrt{t}/v)\right|dv\le\frac{(1/2)_{N+1}2\sqrt{t}}{\sqrt{\pi}(a\sqrt{t})^{2N+3}}\int_1^\infty e^{-c^2v^2/t}v^{2N+1}dv$$

and with $v=\frac{\sqrt{t}}{c}\sqrt{w}$ we get

$$\left|G_N\right|\le\frac{(1/2)_{N+1}}{a\sqrt{\pi}(a^2c^2)^{N+1}}\int_{c^2/t}^\infty e^{-w}w^N dw=\frac{(1/2)_{N+1}}{a\sqrt{\pi}(a^2c^2)^{N+1}}\Gamma(N+1,\frac{c^2}{t})\,.$$

Thus, the asymptotic expansion for $I_{22}(a,0,c,t)$ has an error term bounded by the magnitude of the next term of the expansion and the remainder goes to zero as ac goes to infinity.

For k=0 and k=1 we have

$$\Gamma(0,x) = \int_x^\infty \frac{e^{-w}}{w} dw = \int_1^\infty \frac{e^{-xv}}{v} dv = E_1(x) \ , \qquad w = xv$$

$$\Gamma(1,x) = \int_x^\infty e^{-w} dw = e^{-x} \ .$$

Now, the forward recurrence relation

$$\Gamma(a+1,x) = a\Gamma(a,x) + x^a e^{-x}$$

is numerically stable and can be applied starting with k=1. It is convenient to write the expansion in a scaled form for numerical evaluation:

$$I_{22}(a,0,c,t) \sim \frac{e^{-c^2/t}}{a\sqrt{\pi}} \sum_{k=0}^{N} \frac{(-1)^k (1/2)_k}{(ac)^{2k}} [e^{c^2/t}\Gamma(k,\frac{c^2}{t})] + G_N$$

so that the leading exponential can be tested for underflow. The recurrence becomes

$$[e^x\Gamma(k+1,x)] = k[e^x\Gamma(k,x)] + x^k, \qquad k = 1,2... \qquad x = c^2/t \ .$$

The exponential integral has a natural exponential scaling as well

$$[e^x\Gamma(0,x)] = [e^x E_1(x)]$$

since

$$E_1(x) \sim \frac{e^{-x}}{x}, \qquad x \to \infty \ .$$

It looks like only the product ac must be large, but for large c,

$$[e^{c^2/t}\Gamma(k,c^2/t)] \sim \left(\frac{c^2}{t}\right)^{k-1} , \qquad c^2/t \quad \text{large}$$

and the expansion is in reciprocal powers of $a^2 t$. On the otherhand, when t is large,

$$[e^{c^2/t}\Gamma(k,c^2/t)] \sim \Gamma(k) \ , \qquad c^2/t \quad \text{small}$$

and we have reciprocal powers of ac in the expansion. In either case, we have the expansion for large a.

Folder 23

Evaluation of

$$\mathbf{H_{23}(x) = \int_0^x e^{w^2} erfc(w)\,dw = \frac{1}{2}\int_0^{x^2} e^v erfc(\sqrt{v})\frac{dv}{\sqrt{v}}}$$

$$\mathbf{I_{23}(x) = \int_0^x e^{w^2} erfc(w)\,w\,dw = \frac{1}{2}\int_0^{x^2} e^v erfc(\sqrt{v})\,dv}$$

$$\mathbf{J_{23}(x) = \int_0^x e^{w^2} erfc(w)\,w^2 dw = \frac{1}{2}\int_0^{x^2} e^v erfc(\sqrt{v})\sqrt{v}\,dv}$$

$$\boldsymbol{x \geq 0}$$

Donald E. Amos, November 2002

Summary

In Folders 21 and 22 (see also Beck, et al., [3], p. 424 # 9 & 10) the integrals for H_{23} and J_{23} arise as special cases of the integrals considered there. In this folder we compute H_{23} using power series for $x \leq 4$ and Chebyshev series for x>4. I_{23} has a closed form, but J_{23} can be related to H_{23}. A reduction formula

$$\int_0^x e^{w^2} \mathrm{erfc}(w) w^\alpha dw = \frac{x^{\alpha-1}}{2} e^{x^2} \mathrm{erfc}(x) + \frac{x^\alpha}{\alpha\sqrt{\pi}} - \frac{(\alpha-1)}{2}\int_0^x e^{w^2} \mathrm{erfc}(w) w^{\alpha-2} dw$$

$$\alpha = 2n \text{ or } 2n+1, \quad n = 1, 2, \ldots$$

when applied repeatedly ends up at $H_{23}(x)$ or $I_{23}(x)$. In all of these cases, a direct application of the formula for $x \to 0$ leads to heavy cancellation of digits and the power series is necessary for small x in a numerical evaluation,

$$\int_0^x e^{w^2} \mathrm{erfc}(w) w^\alpha dw = x^{\alpha+1}\sum_{n=0}^{\infty} \frac{(-1)^n x^n}{\Gamma\left(\frac{n}{2}+1\right)(n+\alpha+1)}, \qquad \alpha \geq 0, \quad x \leq 1.$$

H_{23} is evaluated from the relations

$$H_{23}(x) = \begin{cases} S(x,0) & 0 \leq x \leq 1 \\ S_2 + S(x,2) & 1 < x \leq 3 \\ S_4 + S(x,4) & 3 < x \leq 4 \\ S_4 + V(x,4) & 4 < x < \infty \end{cases}$$

where $S(x,x_0)$ is the Taylor expansion about x_0,

$$S(x,x_0)=\sum_{n=0}^{\infty}\ (-2)^n[e^{x_0^2}i^n\ \mathrm{erfc}(x_0)]\frac{(x-x_0)^{n+1}}{n+1}$$

$$S_2=\int_0^2 e^{v^2}\ \mathrm{erfc}(v)dv, \qquad\qquad S_4=\int_0^4 e^{v^2}\ \mathrm{erfc}(v)dv$$

and $V(x,4)$ is the integral

$$V(x,4)=\int_4^x y(v)dv \qquad\qquad y(v)=e^{v^2}\mathrm{erfc}(v)=\frac{1}{v\sqrt{\pi}}\sum_{n=0}^{\infty}{}' a_{2r}T_{2r}\left(\frac{4}{v}\right), \qquad\qquad v\geq 4,$$

where $y(v)$ is a Chebyshev sum with coefficients a_{2r}. The result for $V(x,4)$ is

$$V(x,4)=\frac{1}{\sqrt{\pi}}\left[\sum_{r=0}^{\infty}{}' a_{2r}A_{2r}(x)+\ln\left(\frac{x}{4}\right)\right]$$

where $A_{2r}(x)$ is computed from the recurrence

$$A_0(x)=0, \qquad\qquad A_2(x)=1-\left(\frac{4}{x}\right)^2$$

$$A_{2r+2}(x)+A_{2r}(x)=-\frac{1}{2r(r+1)}-\left[\frac{T_{2r+2}(4/x)}{2r+2}-\frac{T_{2r}(4/x)}{2r}\right], \qquad\qquad r\geq 1.$$

Truncation of $V(x,4)$ at 18 terms suffices for errors $O(10^{-16})$. Here, the a_{2r}'s are Chebyshev coefficients of TABLE 9 of reference [2] and the prime on the sum means to halve the first term.

$I_{23}(x)$ can be evaluated explicitly,

$$I_{23}(x)=\frac{e^{x^2}\ \mathrm{erfc}(x)-1}{2}+\frac{x}{\sqrt{\pi}}$$

and $J_{23}(x)$ can be expressed in terms of $H_{23}(x)$,

$$J_{23}(x)=\frac{1}{2}\left[xe^{x^2}\ \mathrm{erfc}(x)+\frac{x^2}{\sqrt{\pi}}-H_{23}(x)\right]$$

($\alpha = 2$ in the reduction formula). $H_{23}(x)$ is programmed as DOUBLE PRECISION FUNCTION DHERFC(X).

Formulae for $H_{23}(x)$

The basic idea is to integrate the power series for $e^{x^2}\,\mathrm{erfc}(x)$ for x up to 4 and integrate the Chebyshev series developed in [2] for $x > 4$.

The power series about x_0 is arranged so that the differences $|x - x_0|$ do not exceed 1 and we get rapid convergence. Thus, this agenda for the power series is obtained on $0 \le x \le 4$ by

$$H_{23}(x) = \begin{cases} S(x,0) & 0 \le x \le 1 \\ S_2 + S(x,2) & 1 < x \le 3 \\ S_4 + S(x,4) & 3 < x \le 4 \end{cases}$$

where $S(x,x_0)$ is the integral of $e^{x^2}\,\mathrm{erfc}(x)$ on $[x,x_0]$ and $e^{x^2}\,\mathrm{erfc}(x)$ is represented by its power series about x_0. S_2 and S_4 are the integrals of $e^{x^2}\,\mathrm{erfc}(x)$ on [0,2] and [0,4] respectively. The details for generating the power series about x_0 are contained in the relations [1, 7.2.9]

$$\frac{d^n}{dz^n}[e^{z^2}\,\mathrm{erfc}(z)] = (-2)^n\, n!\, e^{z} i^n \mathrm{erfc}(z), \qquad n \ge 0,$$

$$e^{x^2}\,\mathrm{erfc}(x) = \sum_{n=0}^{\infty} (-2)^n [e^{x_0^2} i^n \mathrm{erfc}(x_0)](x - x_0)^n$$

$$S(x, x_0) = \int_{x_0}^{x} e^{v^2}\,\mathrm{erfc}(v)dv = \sum_{n=0}^{\infty} (-2)^n [e^{x_0^2} i^n \mathrm{erfc}(x_0)] \frac{(x - x_0)^{n+1}}{n+1}$$

The convergence of $S(x,x_0)$ can be gauged from the monotonicity of $e^{x^2}\,\mathrm{erfc}(x)$ for $x > 0$ and its value at $x = 0$. The derivative

$$\frac{d}{dx}[e^{x^2} i^n \,\mathrm{erfc}(x)] = -2(n+1) i^{n+1}\,\mathrm{erfc}(x)\cdot e^{x^2}$$

shows that $e^{x^2} i^n \mathrm{erfc}(x)$ is monotone decreasing on $0 \le x < \infty$ and this gives [1, 7.2.7]

$$f(x_0) \equiv e^{x_0^2} i^n \,\mathrm{erfc}(x_0) \le f(0) = \frac{1}{2^n \Gamma\left(\frac{n}{2}+1\right)}, \qquad x_0 \ge 0\,.$$

Forty terms of $S(x,x_0)$ suffice for errors $O(10^{-16})$. The integrals for S_2 and S_4 were generated by integrating $e^{x^2}\,\mathrm{erfc}(x)$ numerically using DGAUS8 with a relative error tolerance of 0.5×10^{-15}.

The results are

$S_2 = 0.9753620874841564$
$S_4 = 1.344468257503159.$

The iterated coerror functions $e^{x_0^2} i^n \operatorname{erfc}(x_0)$ for $S(x,x_0)$ can be generated by subroutine DINERFC with the exponential scaling included (KODE=2 in the call list).

The next part of computing $H_{23}(x)$ for $x > 4$ uses

$$H_{23}(x) = S_4 + V(x,4), \qquad x \geq 4,$$

where V is an expansion for the integral

$$V(x,4) = \int_4^x e^{v^2} \operatorname{erfc}(v)dv, \qquad x \geq 4.$$

The Chebyshev expansion for $e^{v^2} \operatorname{erfc}(v)$ on $v \geq 4$ is [2, p. 28]

$$e^{v^2} \operatorname{erfc}(v) = \frac{1}{v\sqrt{\pi}} \sum_{r=0}^{\infty}{}' a_{2r} T_{2r}\left(\frac{4}{v}\right), \qquad v \geq 4,$$

where 18 values of the a_{2r}'s are tabulated in TABLE 9 of [2] and the prime means to halve the first term. The main problem is to evaluate the integrals in

$$V(x,4) = \int_4^x e^{v^2} \operatorname{erfc}(v)dv = \frac{1}{\sqrt{\pi}} \sum_{r=0}^{\infty}{}' a_{2r} \int_4^x \frac{T_{2r}(4/v)}{v} dv, \qquad x \geq 4.$$

We note that the even order Chebyshev polynomials $T_{2r}(v)$ have a constant term

$$b_{2r,0} = T_{2r}(0) = (-1)^r$$

and a logarithmic term will evolve. To eliminate this, we work with

$$A_{2r}(x) = \int_4^x \frac{T_{2r}(4/v) - b_{2r,0}}{v} dv = \int_{4/x}^1 \frac{T_{2r}(w) - b_{2r,0}}{w} dw = \text{polynomial of degree } 2r \text{ in } (4/x)$$

Since we subtracted off all of the integrals involving constant terms of T_{2r}'s, we have to add them back on. Thus

$$V(x,4) = \frac{1}{\sqrt{\pi}}\left[\sum_{r=0}^{\infty}{}' a_{2r} A_{2r}(x) + \left(\sum_{r=0}^{\infty}{}' a_{2r} b_{2r,0}\right) \ln \frac{x}{4}\right]$$

Now, since $b_{2r,0} = (-1)^r$, reference [2] at the bottom of TABLE 9 gives

$$\sum_{r=0}^{\infty}{}' a_{2r}(-1)^r = 1$$

and

$$V(x,4) = \frac{1}{\sqrt{\pi}}\left[\sum_{r=0}^{\infty}{}' a_{2r}A_{2r}(x) + \ln\frac{x}{4}\right].$$

The next step is to derive a recurrence for $A_{2r}(x)$ from the recurrence of the Chebyshev polynomials,

$$T_{2r+2}(v) = 2vT_{2r+1}(v) - T_{2r}(v), \qquad r \geq 0,$$

where $T_0(v) = 1$ and $T_1(v) = v$. Then

$$\int_{4/x}^{1} \frac{T_{2r+2}(v) - b_{2r+2,0}}{v}dv = 2\int_{4/x}^{1} T_{2r+1}(v)dv - \int_{4/x}^{1} \frac{T_{2r}(v) + b_{2r+2,0}}{v}dv$$

But $b_{2r+2,0} = (-1)^{r+1} = -b_{2r,0} = -(-1)^r$ and

$$A_{2r+2}(x) = 2\int_{4/x}^{1} T_{2r+1}(v)dv - A_{2r}(x), \qquad r \geq 0.$$

Since $T_0(v) = 1$ and $b_{0,0} = 1$, we get

$$A_0(x) = 0, \qquad A_{2r+2}(x) + A_{2r}(x) = 2\int_{4/x}^{1} T_{2r+1}(v)dv, \quad r \geq 0.$$

For the general case where $r \geq 1$, we use the general formula [2]

$$\int T_r(x)dx = \begin{cases} T_1(x) & r = 0 \\ \frac{1}{4}T_2(x) & r = 1 \\ \frac{1}{2}\left[\frac{T_{r+1}(x)}{r+1} - \frac{T_{r-1}(x)}{r-1}\right], & r > 1 \end{cases}$$

to integrate $T_{2r+1}(v)$. Then, for $r \geq 1$

$$2\int_{4/x}^{1} T_{2r+1}(v)dv = \left[\frac{T_{2r+2}(1)}{2r+2} - \frac{T_{2r}(1)}{2r}\right] - \left[\frac{T_{2r+2}(4/x)}{2r+2} - \frac{T_{2r}(4/x)}{2r}\right]$$

or, since $T_n(1) = 1$, $n \geq 0$

$$R_r(x) = 2\int_{4/x}^{1} T_{2r+1}(v)dv = \frac{1}{2}\left\{\frac{-1}{r(r+1)} - \left[\frac{T_{2r+2}(4/x)}{r+1} - \frac{T_{2r}(4/x)}{r}\right]\right\}, \qquad r \geq 1,$$

and

$$A_0(x) = 0, \qquad A_2(x) = 2\int_{4/x}^{1} T_1(v)dv = 1 - \left(\frac{4}{x}\right)^2$$

$$A_{2r+2}(x) + A_{2r}(x) = R_r(x), \qquad r \geq 1.$$

Then

$$V(x,4) = \frac{1}{\sqrt{\pi}}\left[\sum_{r=0}^{\infty}{}' a_{2r} A_{2r}(x) + \ln\left(\frac{x}{4}\right)\right]$$

where the Chebyshev coefficients a_{2r} are given in [2] in TABLE 9 on page 28. Notice that $A_{2r}(x)$ is bounded by

$$|A_{2r}(x)| = \left|\int_{4/x}^{1} \frac{T_{2r}(v) - (-1)^r}{v} dv\right| \leq 2\ln\left(\frac{x}{4}\right), \qquad r \geq 1.$$

Thus, the error in truncating $V(x,4)$ is

$$e_n = \sum_{n+1}^{\infty} a_{2r} A_{2r}(x)$$

and is bounded by

$$|e_n| \leq 2\ln\left(\frac{x}{4}\right)\sum_{n+1}^{\infty} |a_{2r}|$$

which is basically a relative error on the truncation of $V(x,4)$. Thus we take n so that

$$\sum_{n+1}^{\infty} |a_{2r}| = O(10^{-16}).$$

TABLE 9 shows that the terms for $r = 16$ through $r = 18$ sum to $O(10^{-16})$ with $a_{36} = O(10^{-19})$. Thus, $n = 16$ suffices but we take all 18 terms to be conservative.

In summary,

$$H_{23}(x) = \begin{cases} S(x,0) & 0 \le x \le 1 \\ S_2 + S(x,2) & 1 < x \le 3 \\ S_4 + S(x,4) & 3 < x \le 4 \\ S_4 + V_n(x,4) & 4 < x < \infty, \qquad n = 18 \end{cases}$$

where $S(x,x_0)$ is the Taylor expansion about x_0

$$S(x,x_0) = \sum_{n=0}^{\infty} (-2)^n [e^{x_0^2} i^n \operatorname{erfc}(x_0)] \frac{(x-x_0)^{n+1}}{n+1}$$

$$S_2 = \int_0^2 e^{v^2} \operatorname{erfc}(v)dv = 0.9753620874841564$$

$$S_4 = \int_0^4 e^{v^2} \operatorname{erfc}(v)dv = 1.344468257503159$$

and $V_n(x,4)$ is the truncation of

$$V(x,4) = \frac{1}{\sqrt{\pi}} \left[\sum_{r=0}^{\infty}{}' a_{2r} A_{2r}(x) + \ln \frac{x}{4} \right]$$

where $A_{2r}(x)$ is computed from the recurrence

$$A_0(x) = 0 \qquad A_2(x) = 1 - \left(\frac{4}{x}\right)^2$$

$$A_{2r+2}(x) + A_{2r}(x) = -\frac{1}{2r(r+1)} - \left[\frac{T_{2r+2}(4/x)}{2r+2} - \frac{T_{2r}(4/x)}{2r}\right], \qquad r \ge 1.$$

The values for a_{2r}, $r \ge 0$ are copied from [2], TABLE 9. $H_{23}(x)$ is programmed as double precision function DHERFC(X).

CHEBYSHEV COEFFICIENTS FOR EXP(X*X)*ERFC(X)*(X*SQRT(PI))
ON THE INTERVAL X.GE.4

REF. C.W. CLENSHAW, CHEBYSHEV SERIES FOR MATHEMATICAL FUNCTIONS, NATIONAL PHYSICAL LABORATORY MATHEMATICAL TABLES, VOLUME 5, HER MAJESTY'S STATIONERY OFFICE, LONDON, 1963, TABLE 9.

```
      DATA (AR(I),   I=1,18)/
&          1.97070527225754492387D0,      -0.01433974027177497552D0,
&          0.00029736169220261895D0,      -0.00000980351604336237D0,
&          0.00000043313342034728D0,      -0.00000002362150026241D0,
&          0.00000000151549676581D0,      -0.00000000011084939856D0,
&          0.00000000000904259014D0,      -0.00000000000080947054D0,
&          0.00000000000007853856D0,      -0.00000000000000817918D0,
&          0.00000000000000090715D0,      -0.00000000000000010646D0,
&          0.00000000000000001315D0,      -0.00000000000000000170D0,
&          0.00000000000000000023D0,      -0.00000000000000000003D0/
```

Formula for $I_{23}(x)$

In this case, integration by parts on

$$I_{23}(x) = \int_0^x e^{w^2} \operatorname{erfc}(w) w\,dw$$

$$u = \operatorname{erfc}(w) \qquad dv = e^{w^2} w\,dw$$

$$du = -\frac{2}{\sqrt{\pi}} e^{-w^2} dw \qquad v = \frac{e^{w^2}}{2}$$

gives

$$I_{23}(x) = \frac{1}{2}[e^{x^2} \operatorname{erfc}(x) - 1] + \frac{1}{\sqrt{\pi}} \int_0^x dw$$

$$= \frac{1}{2}[e^{x^2} \operatorname{erfc}(x) - 1] + \frac{x}{\sqrt{\pi}}.$$

Formula for $J_{23}(x)$

We integrate $J_{23}(x)$ by parts

$$J_{23}(x) = \int_0^x e^{w^2} \operatorname{erfc}(w) w^2\,dw$$

$$u = w\text{erfc}(w) \qquad\qquad dv = e^{w^2}\, w dw$$

$$du = \left[-\frac{2w}{\sqrt{\pi}} e^{-w^2} + \text{erfc}(w) \right] dw \qquad v = \frac{e^{w^2}}{2}$$

Then,

$$J_{23}(x) = \frac{1}{2} x e^{x^2} \text{ erfc}(x) + \frac{1}{\sqrt{\pi}} \int_0^x w dw - \frac{1}{2} \int_0^x e^{w^2} \text{ erfc}(w) dw$$

$$= \frac{1}{2} x e^{x^2} \text{ erfc}(x) + \frac{x^2}{2\sqrt{\pi}} - \frac{1}{2} H_{23}(x)$$

Reduction Formula for More General Cases

We consider

$$I_\alpha(x) = \int_0^x e^{w^2} \text{ erfc}(w) w^\alpha dw, \qquad \alpha = 2n \text{ or } 2n{+}1, \quad \text{n} = 1,2, \ldots$$

and integrate by parts to get a reduction formula for larger α's. Then

$$u = w^{\alpha-1}\text{erfc}(w) \qquad\qquad dv = w e^{w^2}$$

$$du = \left[-\frac{2w^{\alpha-1}}{\sqrt{\pi}} e^{-w^2} + (\alpha - 1) w^{\alpha-2} \text{ erfc}(w) \right] dw \qquad v = \frac{1}{2} e^{w^2}$$

and

$$I_\alpha(x) = \frac{1}{2} x^{\alpha-1} e^{x^2} \text{ erfc}(x) + \frac{1}{\sqrt{\pi}} \int_0^x w^{\alpha-1} dw - \frac{(\alpha - 1)}{2} \int_0^x e^{w^2} \text{ erfc}(w) w^{\alpha-2} dw$$

$$= \frac{1}{2} x^{\alpha-1} e^{x^2} \text{ erfc}(\mathbf{x}) + \frac{1}{\alpha\sqrt{\pi}} x^\alpha - \frac{(\alpha - 1)}{2} I_{\alpha-2}(x).$$

Notice that for $\alpha = 2$ we get the formula for $J_{23}(x)(=I_2(x))$ since $I_0(x) = H_{23}(x)$. Since $\alpha = 2n$ or $2n+1$, a repeated application of this formula ends up with $I_0(x)$ or $I_1(x)$ which translate to $H_{23}(x)$ or $I_{23}(x)$ respectively.

Numerically, all of the formulas derived so far suffer from high cancellation of significant digits when x is small which results in a loss of relative error.

To combat this flaw, we generate the power series for $x \le 1$.

Power Series for $I_\alpha(x)$, $H_{23}(x)$, $I_{23}(x)$, $J_{23}(x)$, for $x \le 1$

In order to retain significant digits when x is small, we generate the power series for $I_\alpha(x)$. We start with

$$e^{w^2}\operatorname{erfc}(w) = \sum_{n=0}^{\infty} \frac{(-w)^n}{\Gamma\left(\frac{n}{2}+1\right)} ,$$

which is the power series expansion about $w = 0$, [1, 7.2.9 and 7.2.7] since

$$\frac{d^n}{dw^n}[e^{w^2}\operatorname{erfc}(w)] = (-2)^n n! e^{w^2} i^n \operatorname{erfc}(w)$$

$$i^n \operatorname{erfc}(0) = \frac{1}{2^n \Gamma\left(\frac{n}{2}+1\right)}, \qquad n = -1, 0, 1, \ldots$$

Then

$$I_\alpha(x) = \int_0^x e^{w^2}\operatorname{erfc}(w) w^\alpha \, dw = x^{\alpha+1} \sum_{n=0}^{\infty} \frac{(-x)^n}{\Gamma\left(\frac{n}{2}+1\right)(n+\alpha+1)}$$

The cancellation of leading terms is quite evident for $I_{23}(x) = I_1(x)$ since the result is $O(x^2)$ for small x, but the main terms of $I_{23}(x)$ from the closed form are $O(x)$ for $x \to 0$. The power series for $H_{23}(x)$, $I_{23}(x)$ and $J_{23}(x)$ are:

$$(\alpha = 0) \qquad H_{23}(x) = x \sum_{n=0}^{\infty} \frac{(-x)^n}{\Gamma\left(\frac{n}{2}+1\right)(n+1)} \qquad 0 \le x \le 1$$

$$(\alpha = 1) \qquad I_{23}(x) = x^2 \sum_{n=0}^{\infty} \frac{(-x)^n}{\Gamma\left(\frac{n}{2}+1\right)(n+2)} \qquad 0 \le x \le 1$$

$(\alpha = 2)$ $\quad J_{23}(x) = x^3 \sum_{n=0}^{\infty} \dfrac{(-x)^n}{\Gamma\left(\dfrac{n}{2}+1\right)(n+3)}$ $\quad 0 \le x \le 1$

Forty terms of each series suffice for relative errors $O(10^{-16})$.

<u>Asymptotic Expansion for</u> $\mathbf{H_{23}(x)}$ <u>for</u> $\mathbf{x \to \infty}$

We start with the integral representation for (EMOT, Tables of Integral Transforms, Bateman Project,Vol.1, Page 146, (21))

$$\sqrt{\pi\tau}e^{a^2\tau} erfc(a\sqrt{\tau}) = \int_0^{\infty} e^{-ay-y^2/(4\tau)} dy .$$

Then, division by τ and integration yield

$$\sqrt{\pi}\int_0^t \frac{e^{a^2\tau} erfc(a\sqrt{\tau})}{\sqrt{\tau}} d\tau = \int_0^{\infty} e^{-ay} \int_0^t \frac{e^{-y^2/(4\tau)}}{\tau} d\tau dy$$

and with the substitutions $\tau = t/w$ and $y = (2\sqrt{t})v$ we get

$$\int_0^t \frac{e^{a^2\tau} erfc(a\sqrt{\tau})}{\sqrt{\tau}} d\tau = \frac{1}{\sqrt{\pi}}\int_0^{\infty} e^{-ay} E_1(\frac{y^2}{4t}) dy = \frac{2\sqrt{t}}{\sqrt{\pi}}\int_0^{\infty} e^{-2a\sqrt{t}\,v} E_1(v^2) dv .$$

Now this is a Laplace transform and the asymptotic expansion for large $2a\sqrt{t}$ is obtained by expanding $E_1(v^2)$ *about* $v = 0$,

$$E_1(v^2) = -\gamma - \ln v^2 - \sum_{n=1}^{\infty} \frac{(-1)^n v^{2n}}{nn!} .$$

The Laplace transform integrals,

$$\int_0^{\infty} e^{-pv} \ln v^2 dv = -\frac{2}{p}(\gamma + \ln p)$$

$$\int_0^{\infty} e^{-pv} v^{2n} dv = \frac{\Gamma(2n+1)}{p^{2n+1}} = \frac{(2n)!}{p^{2n+1}}$$

with $p = 2a\sqrt{t}$ and a=1 give

$$\int_0^t \frac{e^{\tau} erfc(\sqrt{\tau})}{\sqrt{\tau}} d\tau \sim \frac{1}{\sqrt{\pi}}\left[\gamma + 2\ln(2\sqrt{t}) - \sum_{n=1}^{\infty} \frac{C_n}{(2\sqrt{t})^{2n}}\right], \qquad C_n = \frac{(-1)^n}{n}\frac{\Gamma(2n+1)}{\Gamma(n+1)} = \frac{(-1)^n 2^{2n}(1/2)_n}{n} .$$

Here we have used

$$\Gamma(2z) = \frac{2^{2z-1}}{\sqrt{\pi}}\Gamma(z)\Gamma(z+\frac{1}{2}), \quad \Gamma(1/2) = \sqrt{\pi}, \quad \textit{and} \quad (1/2)_n = (1/2)(3/2)...(n-1/2)$$

to simplify C_n.

Now,

$$H_{23}(x) = \int_0^x e^{w^2}\, erfc(w)\,dw = \frac{1}{2}\int_0^{x^2} \frac{e^{\tau}\, erfc(\sqrt{\tau})}{\sqrt{\tau}}\, d\tau \sim \frac{1}{2\sqrt{\pi}}\left[\gamma + 2\ln(2x) - \sum_{n=1}^{\infty} \frac{A_n}{x^{2n}}\right]$$

where

$$A_n = \frac{(-1)^n (1/2)_n}{n}.$$

Numerical experiments comparing this expansion with a DGAUS8 quadrature on the first integral show that relative errors $O(10^{-14})$ are obtained for $x \geq 7$ in less than 20 terms.

References

[1] Abramowitz, M. and Stegun, I.A., *Handbook of Mathematical Functions*, U.S. Department of Commerce, National Bureau of Standards, AMS55, 1964.

[2] Clenshaw, C.W., *Chebyshev Series for Mathematical Functions*, National Physical Laboratory Mathematical Tables, Volume 5, Her Majesty's Stationery Office, London, 1963, TABLE 9.

[3] Beck, J.V., et al., *Heat Conduction Using Green's Functions*, Hemisphere Publishing Corp., Washington, D.C., 1992.

Folder 24

Evaluation of

$$\boldsymbol{I_{24}(a,b,c,t)=\int_0^t \tau U(\tau)\,\mathrm{erf}\left(\frac{c}{\sqrt{\tau}}\right)d\tau,\qquad I_{24}^{c}(a,b,c,t)=\int_0^t \tau U(\tau)\,\mathrm{erfc}\left(\frac{c}{\sqrt{\tau}}\right)d\tau}$$

$$\boldsymbol{J_{24}(a,b,t)=\int_0^t \tau U(\tau)d\tau,\qquad V_{24}(a,b,t)=\int_0^t V(\tau)d\tau}$$

where

$$\mathbf{U(t)=e^{a^2t+2ab}\,erfc(a\sqrt{t}+b/\sqrt{t})}$$

$$\mathbf{V(t)=\int_0^t U(\tau)\,d\tau=\frac{2}{a}\sqrt{\frac{t}{\pi}}\,e^{-b^2/t}-\left(\frac{1}{a^2}+\frac{2b}{a}\right)erfc\left(\frac{b}{\sqrt{t}}\right)+\frac{U(t)}{a^2}}$$

$$\boldsymbol{a>0,\quad b>0,\quad c>0,\quad t>0}$$

Donald E. Amos, December 2002

Summary

In Folders 21 and 22, we derived formulas closely related to those above. This folder extends those results:

$$I_{24}(a,b,c,t)=tV(t)\,\mathrm{erf}\left(\frac{c}{\sqrt{t}}\right)+\frac{c}{\sqrt{\pi}}T_0-[T_1-T_2+T_3]$$

where $V(t)$ is computed in Folder 21 and

$$T_0(a,b,c,t)=\frac{2t}{a\sqrt{\pi}}E_2(X)-2\left(\frac{1}{a}+\frac{2b}{a}\right)I_1^c(c,b,T)+\frac{1}{a^2}I_{22}(a,b,c,t)$$

$$T_1(a,b,c,t)=\frac{4}{3a\sqrt{\pi}}\left[\frac{e^{-b^2T^2}\,\mathrm{erf}(cT)}{T^3}+\frac{c}{T^2\sqrt{\pi}}E_2(X)-2b^2I_1(b,c,T)\right]$$

$$T_2(a,b,c,t)=2\left(\frac{1}{a^2}+\frac{2b}{a}\right)[J_3(c,T)-W_3(b,c,T)]=2\left(\frac{1}{a^2}+\frac{2b}{a}\right)[J_3^c(b,T)-W_3^c(b,c,T)]$$

$$T_3(a,b,c,t)=\frac{1}{a^2}I_{21}(a,b,c,t),\qquad T=\frac{1}{\sqrt{t}},\qquad X=(b^2+c^2)T^2.$$

Similarly for

$$I_{24}^{c}(a,b,c,t) = tV(t)\,\mathrm{erfc}\left(\frac{c}{\sqrt{t}}\right) - \frac{c}{\sqrt{\pi}}T_0^c - [T_1^c - T_2^c + T_3^c]$$

where

$$T_0^c = T_0$$

$$T_1^c = \frac{4}{3a\sqrt{\pi}}\left[\frac{e^{-b^2T^2}\,\mathrm{erfc}(cT)}{T^3} - \frac{c}{T^2\sqrt{\pi}}E_2(X) - 2b^2 I_1^c(b,c,T)\right]$$

$$T_2^c = 2\left(\frac{1}{a^2} + \frac{2b}{a}\right)W_3^c(b,c,T)$$

$$T_3^c = \frac{1}{a^2}I_{21}^c(a,b,c,t), \qquad T = \frac{1}{\sqrt{t}}, \qquad X = (b^2+c^2)T^2.$$

The integrals J_3, J_3^c, I_1, I_1^c, W_3, W_3^c, are given in Folder 10, and I_{21} and I_{22} are the integrals of Folders 21 and 22. We also have

$$\int_0^t \frac{V(\tau)e^{-c^2/\tau}}{\sqrt{\tau}}d\tau = T_0, \quad \int_0^t V(\tau)\mathrm{erf}\left(\frac{c}{\sqrt{\tau}}\right)d\tau = T_1 - T_2 + T_3, \quad \int_0^t V(\tau)\mathrm{erfc}\left(\frac{c}{\sqrt{\tau}}\right)d\tau = T_1^c - T_2^c + T_3^c$$

and

$$J_{24}(a,b,t) = \lim_{c\to\infty} I_{24}(a,b,c,t) = tV(t) - \int_0^t V(\tau)d\tau$$

The results are

$$J_{24}(a,b,t) = \left(t - \frac{1}{a^2}\right)V(t) - \frac{2}{aT^3\sqrt{\pi}}E_{5/2}(b^2T^2) + \frac{4}{T^2}\left(\frac{1}{a^2} + \frac{2b}{a}\right)i^2\,\mathrm{erfc}(bT)$$

It follows that

$$V_{24}(a,b,t) = \frac{1}{a^2}V(t) + \frac{2}{aT^3\sqrt{\pi}}E_{5/2}(b^2T^2) - \frac{4}{T^2}\left(\frac{1}{a^2} + \frac{2b}{a}\right)i^2\,\mathrm{erfc}(bT)$$

and

$$V_{24}(a,b,t) = \frac{1}{a^2}V(t) + \frac{8}{aT^3}i^3 erfc(bT) - \frac{4}{a^2T^2}i^2 erfc(bT)$$

It is clear that there are losses of significance when a is small and the corresponding formulae for $a\sqrt{t} \le 1$ are:

$$I_{24}(a,b,c,t) = J_{24}(a,b,t)\,\mathrm{erf}\left(\frac{c}{\sqrt{t}}\right) + \frac{8cte^{-X}}{\sqrt{\pi}}\left[b\sum_{n=0}^{\infty} (-2a\sqrt{t})^n y_{n+3} + \sqrt{t}\sum_{n=0}^{\infty} (n+2)(-2a\sqrt{t})^n y_{n+4}\right]$$

$$I_{24}^{c}(a,b,c,t) = J_{24}(a,b,t)\,\mathrm{erfc}\left(\frac{c}{\sqrt{t}}\right) - \frac{8cte^{-X}}{\sqrt{\pi}}\left[b\sum_{n=0}^{\infty} (-2a\sqrt{t})^n y_{n+3} + \sqrt{t}\sum_{n=0}^{\infty} (n+2)(-2a\sqrt{t})^n y_{n+4}\right]$$

$$J_{24}(a,b,t) = tV(t) - V_{24}(a,b,t)$$

$$V_{24}(a,b,t) = 16t^2 \sum_{n=0}^{\infty} (-2a\sqrt{t})^n\, i^{n+4}\,\mathrm{erfc}(bT)$$

where $V(t)$ and the $y_n(c,b,T)$ sequence are computed in Folder 21,

$$V(t) = 4t\sum_{n=0}^{\infty} (-2a\sqrt{t})^n\, i^{n+2}\,\mathrm{erfc}(bT) \qquad T = \frac{1}{\sqrt{t}}, \qquad X = (b^2+c^2)T^2$$

$$y_n(c,b,T) = 2T^{n-1}e^{X}\int_T^{\infty} e^{-c^2w^2}\,\frac{i^n\,\mathrm{erfc}(bw)}{w^n}\,dw\,.$$

The iterated coerror functions are generated in subroutine DINERFC.

Computation of $I_{24}(a,b,c,t)$ and $I_{24}^{c}(a,b,c,t)$

For I_{24} we integrate by parts

$$I_{24}(a,b,c,t) = \int_0^t \tau U(\tau)\,\mathrm{erf}\left(\frac{c}{\sqrt{\tau}}\right)d\tau$$

$$u = \tau\,\mathrm{erf}\left(\frac{c}{\sqrt{\tau}}\right) \qquad\qquad dv = U(\tau)d\tau$$

$$du = \left[-\frac{c}{\sqrt{\pi}}\frac{c^{-c^2/\tau}}{\sqrt{\tau}} + \mathrm{erf}\left(\frac{c}{\sqrt{\tau}}\right)\right]d\tau \qquad\qquad v = V(\tau)$$

where $V(t)$ is computed in Folder 21,

$$V(t)=\frac{2}{a}\sqrt{\frac{t}{\pi}}e^{-b^2/t}-\left(\frac{1}{a^2}+\frac{2b}{a}\right)\operatorname{erfc}\left(\frac{b}{\sqrt{t}}\right)+\frac{U(t)}{a^2}.$$

Then,

$$I_{24}(a,b,c,t)=tV(t)\operatorname{erf}\left(\frac{c}{\sqrt{t}}\right)+\frac{c}{\sqrt{\pi}}\int_0^t\frac{V(\tau)}{\sqrt{\tau}}e^{-c^2/\tau}d\tau-\int_0^t V(\tau)\operatorname{erf}\left(\frac{c}{\sqrt{\tau}}\right)d\tau.$$

Now, we evaluate each of the integrals coming from $V(\tau)$ and write I_{24} as

$$I_{24}(a,b,c,t)=tV(t)\operatorname{erf}\left(\frac{c}{\sqrt{t}}\right)+\frac{c}{\sqrt{\pi}}T_0-[T_1-T_2+T_3].$$

$$\underline{T_0(a,b,c,t)=}\int_0^t\frac{V(\tau)}{\sqrt{\tau}}e^{-c^2/\tau}d\tau$$

$$=\frac{2}{a\sqrt{\pi}}\int_0^t e^{-(b^2+c^2)/\tau}d\tau-\left(\frac{1}{a^2}+\frac{2b}{a}\right)\int_0^t\frac{e^{-c^2/\tau}\operatorname{erfc}(b/\sqrt{\tau})}{\sqrt{\tau}}d\tau+\frac{1}{a^2}\int_0^t\frac{U(\tau)}{\sqrt{\tau}}e^{-c^2/\tau}d\tau$$

$$=\frac{2t}{a\sqrt{\pi}}E_2(X)-2\left(\frac{1}{a^2}+\frac{2b}{a}\right)\int_T^\infty\frac{e^{-c^2w^2}\operatorname{erfc}(bw)}{w^2}dw+\frac{1}{a^2}I_{22}(a,b,c,t)$$

$$=\frac{2t}{a\sqrt{\pi}}E_2(X)-2\left(\frac{1}{a^2}+\frac{2b}{a}\right)I_1^c(c,b,T)+\frac{1}{a^2}I_{22}(a,b,c,t)$$

where I_1^c is evaluated in Folder 10 and I_{22} is evaluated in Folder 22.

$$\underline{T_1(a,b,c,t)}=\frac{2}{a\sqrt{\pi}}\int_0^t\sqrt{\tau}e^{-b^2/\tau}\operatorname{erf}\left(\frac{c}{\sqrt{\tau}}\right)d\tau=\frac{4}{a\sqrt{\pi}}\int_T^\infty\frac{e^{-b^2w^2}\operatorname{erf}(cw)}{w^4}dw$$

Integration by parts gives

$$u=e^{-b^2w^2}\operatorname{erf}(cw)\qquad\qquad dw=dw/w^4$$

$$du=\left[\frac{2c}{\sqrt{\pi}}e^{-(b^2+c^2)w^2}-2b^2we^{-b^2w^2}\operatorname{erf}(cw)\right]dw\qquad\qquad v=-\frac{1}{3w^3}$$

$$T_1=\frac{4}{a\sqrt{\pi}}\left[\frac{e^{-b^2T^2}\operatorname{erf}(cT)}{3T^3}+\frac{2c}{3\sqrt{\pi}}\int_T^\infty\frac{e^{-(b^2+c^2)w^2}}{w^3}dw-\frac{2b^2}{3}\int_T^\infty\frac{e^{-b^2w^2}\operatorname{erf}(cw)}{w^2}dw\right]$$

$$=\frac{4}{3a\sqrt{\pi}}\left[\frac{e^{-b^2T^2}\operatorname{erf}(cT)}{T^3}+\frac{c}{T^2\sqrt{\pi}}E_2(X)-2b^2I_1(b,c,T)\right],\qquad X=(b^2+c^2)/t,$$

where I_1 is computed in Folder 10.

$$\underline{T_2(a,b,c,t)} = 2\left(\frac{1}{a^2}+\frac{2b}{a}\right)\int_0^t \operatorname{erfc}\left(\frac{b}{\sqrt{\tau}}\right)\operatorname{erf}\left(\frac{c}{\sqrt{\tau}}\right)d\tau$$

$$= 2\left(\frac{1}{a^2}+\frac{2b}{a}\right)\int_T^\infty \frac{\operatorname{erfc}(bw)\operatorname{erf}(cw)}{w^3}dw$$

$$T_2(a,b,c,t) = 2\left(\frac{1}{a^2}+\frac{2b}{a}\right)\left[\int_T^\infty \frac{\operatorname{erfc}(bw)}{w^3}dw - \int_T^\infty \frac{\operatorname{erfc}(bw)\operatorname{erfc}(cw)}{w^3}dw\right]$$

$$= 2\left(\frac{1}{a^2}+\frac{2b}{a}\right)[J_3^c(b,T) - W_3^c(b,c,T)]$$

We also have

$$T_2(a,b,c,t) = 2\left(\frac{1}{\mathrm{a}^2}+\frac{2\mathrm{b}}{\mathrm{a}}\right)\left[\int_\mathrm{T}^\infty \frac{\operatorname{erf}(\mathrm{cw})}{\mathrm{w}^3}\mathrm{dw} - \int_\mathrm{T}^\infty \frac{\operatorname{erf}(\mathrm{bw})\operatorname{erf}(\mathrm{cw})}{\mathrm{w}^3}\mathrm{dw}\right]$$

$$= 2\left(\frac{1}{a^2}+\frac{2b}{a}\right)[J_3(c,T) - W_3(b,c,T)]$$

Obviously, any convex linear combination of these two results is also a representation of $T_2(a,b,c,t)$. Here, $J_3(c,T)$, $J_3^c(bT)$, $W_3(b,c,T)$ and $W_3^c(b,c,T)$ are computed in Folder 10.

$$\underline{T_3(a,b,c,t)} = \frac{1}{a^2}\int_0^t U(\tau)\operatorname{erf}\left(\frac{c}{\sqrt{\tau}}\right)d\tau = \frac{1}{a^2}I_{21}(a,b,c,t)$$

where I_{21} is computed in Folder 21.

The final result for $I_{24}(a,b,c,t)$ is

$$I_{24}(a,b,c,t) = tV(t)\operatorname{erf}\left(\frac{c}{\sqrt{t}}\right) + \frac{c}{\sqrt{\pi}}T_0 - [T_1 - T_2 + T_3]$$

where

$$T_0 = \frac{2t}{a\sqrt{\pi}}E_2(X) - 2\left(\frac{1}{a^2}+\frac{2b}{a}\right)I_1^c(c,b,T) + \frac{1}{a^2}I_{22}(a,b,c,t)$$

$$T_1 = \frac{4}{3a\sqrt{\pi}}\left[\frac{e^{-b^2T^2}\operatorname{erf}(cT)}{T^3} + \frac{c}{T^2\sqrt{\pi}}E_2(X) - 2b^2 I_1(b,c,T)\right]$$

$$T_2 = 2\left(\frac{1}{a^2}+\frac{2b}{a}\right)[J_3(c,T)-W_3(b,c,T)] = 2\left(\frac{1}{a^2}+\frac{2b}{a}\right)[J_3^c(b,T)-W_3^c(b,c,T)]$$

$$T_3 = \frac{1}{a^2}I_{21}(a,b,c,t) \qquad T = \frac{1}{\sqrt{t}}, \qquad X = (b^2+c^2)T^2.$$

While we can use erfc(x) = 1-erf(x) to get $I_{24}^c(a,b,c,t)$ in terms of $I_{24}(a,b,c,t)$,

$$I_{24}^c(a,b,c,t) = J_{24}(a,b,t) - I_{24}(a,b,c,t)$$

it is probably better to re-compute T_0^c, T_1^c, T_2^c and T_3^c to get to proper scaling on each term. Thus

$$I_{24}^c(a,b,c,t) = tV(t)\,\mathrm{erfc}\left(\frac{c}{\sqrt{t}}\right) - \frac{c}{\sqrt{\pi}}T_0^c - [T_1^c - T_2^c + T_3^c]$$

where

$$T_0^c = T_0$$

$$T_1^c = \frac{4}{3a\sqrt{\pi}}\left[\frac{e^{-b^2T^2}\,\mathrm{erfc}(cT)}{T^3} - \frac{c}{T^2\sqrt{\pi}}E_2(X) - 2b^2 I_1^c(b,c,T)\right]$$

$$T_2^c = 2\left(\frac{1}{a^2}+\frac{2b}{a}\right)W_3^c(b,c,T)$$

$$T_3^c = \frac{1}{a^2}I_{21}^c(a,b,c,t), \qquad T = \frac{1}{\sqrt{t}}, \qquad X = (b^2+c^2)T^2.$$

The integrals I_1, I_1^c, W_3, W_3^c, J_3 and J_3^c are computed in Folder 10 and I_{21} and I_{21}^c are computed in Folder 21.

Computation of $J_{24}(a,b,t)$

Notice that

$$J_{24}(a,b,t) = \lim_{c\to\infty} I_{24}(a,b,c,t).$$

Taking the limit of each term gives:

$$\lim_{c\to\infty} tV(t)erf(c/\sqrt{t}) = tV(t)$$
$$\lim_{c\to\infty} cT_0(a,b,c,t)/\sqrt{\pi} = 0$$

since

$$I_1^c = \int_T^\infty \frac{e^{-c^2w^2}\,\mathrm{erfc}(bw)}{w^2}\,dw \quad \text{and} \quad I_{22} = \int_0^t \frac{U(\tau)e^{-c^2/\tau}}{\sqrt{\tau}}\,d\tau$$

$$I_1^c \le e^{-c^2T^2}\int_T^\infty \frac{\mathrm{erfc}(bw)}{w^2}\,dw \qquad\qquad I_{22} \le e^{-c^2/t}\int_0^t \frac{U(\tau)}{\sqrt{\tau}}\,d\tau$$

$$\lim_{c\to\infty} T_1(a,b,c,t) = \frac{4}{3a\sqrt{\pi}}\left[\frac{e^{-b^2T^2}}{T^3} - 2b^2\lim_{c\to\infty} I_1(b,c,T)\right]$$

$$= \frac{4}{3a\sqrt{\pi}}\left[\frac{e^{-b^2T^2}}{T^3} - 2b^2\lim_{c\to\infty}\int_T^\infty \frac{e^{-b^2w^2}\,\mathrm{erf}(cw)}{w^2}\,dw\right]$$

$$= \frac{4}{3a\sqrt{\pi}}\left[\frac{e^{-b^2T^2}}{T^3} - \frac{b^2}{T}E_{3/2}(b^2T^2)\right] = \frac{2}{aT^3\sqrt{\pi}}E_{5/2}(b^2T^2)$$

$$\lim_{c\to\infty} T_2(a,b,c,t) = 2\left(\frac{1}{a^2}+\frac{2b}{a}\right)\int_T^\infty \frac{\mathrm{erfc}(bw)}{w^3}\,dw$$

$$= 2\left(\frac{1}{a^2}+\frac{2b}{a}\right)J_3^c(b,T) = \frac{4}{T^2}\left(\frac{1}{a^2}+\frac{2b}{a}\right)i^2\,\mathrm{erfc}(bT)$$

where $J_3^c(b,T)$ is defined in Folder 10. Finally,

$$\lim_{c\to\infty} T_3(a,b,c,t) = \lim_{c\to\infty}\frac{1}{a^2}I_{21}(a,b,c,t) = \frac{V(t)}{a^2}$$

Then,

$$J_{24}(a,b,t) = \left(t-\frac{1}{a^2}\right)V(t) - \frac{2}{aT^3\sqrt{\pi}}E_{5/2}(b^2T^2) + \frac{4}{T^2}\left(\frac{1}{a^2}+\frac{2b}{a}\right)i^2\,\mathrm{erfc}(bT)\,.$$

Computation of $V_{24}(a,b,t)$

Notice that

$$J_{24}(a,b,t) = \int_0^t \tau U(\tau)\,d\tau = tV(t) - \int_0^t V(\tau)\,d\tau$$

by integration by parts. Therefore

$$V_{24}(a,b,t) = \int_0^t V(\tau)d\tau = tV(t) - J_{24}(a,b,t)$$

$$= \frac{V(t)}{a^2} + \frac{2}{aT^3\sqrt{\pi}} E_{5/2}(b^2T^2) - \frac{4}{T^2}\left(\frac{1}{a^2} + \frac{2b}{a}\right) i^2 \operatorname{erfc}(bT)$$

and since

$$E_{5/2}(b^2T^2) = \frac{2}{3}\left[\, e^{-b^2T^2} - b^2T^2 E_{3/2}(b^2T^2) \,\right],$$

$$E_{3/2}(b^2T^2) = 2\sqrt{\pi}\ \operatorname{ierfc}(bT) \qquad \frac{e^{-b^2T^2}}{\sqrt{\pi}} = i\operatorname{erfc}(bT) + (bT)\operatorname{erfc}(bT)$$

we can express V_{24} in terms of iterated coerror functions

$$V_{24}(a,b,t) = \frac{V(t)}{a^2} + \frac{4}{3aT^3}[(bT)\operatorname{erfc}(bT) + (1-2b^2T^2)\operatorname{ierfc}(bT)] - \frac{4}{T^2}\left(\frac{1}{a^2} + \frac{2b}{a}\right) i^2 \operatorname{erfc}(bT)$$

As in previous folders, the results break down computationally for $a \to 0$ due to small differences of large numbers. These indeterminate forms are resolved in the next section.

Expansions for small $a\sqrt{t} \le 1$

The closed form expression for $V(t)$

$$V(t) = \frac{2}{a}\sqrt{\frac{t}{\pi}} e^{-b^2/t} - \left(\frac{1}{a^2} + \frac{2b}{a}\right) \operatorname{erfc}\left(\frac{b}{\sqrt{t}}\right) + \frac{1}{a^2} U(t)$$

breaks down for $a \to 0$ and results in an indeterminate form which is resolved in Folder 21,

$$V(t) = 4t \sum_{n=0}^{\infty} (-2a\sqrt{t})^n\, i^{n+2} \operatorname{erfc}(bT), \qquad\qquad T = \frac{1}{\sqrt{t}}.$$

Since $I_{24}(a,b,c,t)$ is analytic in a, we know that the coefficients of the powers a^{-1}, a^{-2} must be zero. Consequently we need only pick out the non-negative powers from the expansions. Now T_1 and T_2 have reciprocal powers a^{-1}, a^{-2}, but only $I_{22}(a,b,c,t)$ from T_0 contributes to non-negative powers of a. This is shown by the expansion from Folder 22,

$$I_{22}(a,b,c,t) = 2e^{-X}\left[b\sum_{n=0}^{\infty} (-2a\sqrt{t})^n\, y_{n+1} + \sqrt{t}\sum_{n=0}^{\infty} (n+2)(-2a\sqrt{t})^n\, y_{n+2} \right]$$

and similarly for T_3 and T_3^c, only I_{21} and I_{21}^c contribute to the non-negative powers,

$$I_{21}(a,b,c,t) = V(t)\mathrm{erf}\left(\frac{c}{\sqrt{t}}\right) + \frac{4c\sqrt{t}e^{-X}}{\sqrt{\pi}}\sum_{n=0}^{\infty}(-2a\sqrt{t})^n y_{n+2}$$

$$I_{21}^c(a,b,c,t) = V(t)\mathrm{erfc}\left(\frac{c}{\sqrt{t}}\right) - \frac{4c\sqrt{t}e^{-X}}{\sqrt{\pi}}\sum_{n=0}^{\infty}(-2a\sqrt{t})^n y_{n+2}$$

$$V(t) = 4t\sum_{n=0}^{\infty}(-2a\sqrt{t})i^{n+2}\,\mathrm{erfc}(bT), \qquad X = (b^2+c^2)T^2.$$

Here the y_n sequence, $y_n(c,b,T)$, is given in Folder 21 and computed in FORTRAN subroutine GNSEQ. Then, dropping all terms in a^{-1} and a^{-2}, we get

$$I_{24}(a,b,c,t) = tV(t)\,\mathrm{erf}\left(\frac{c}{\sqrt{t}}\right) + \frac{2ce^{-X}}{a^2\sqrt{\pi}}\left[b\sum_{n=2}^{\infty}(-2a\sqrt{t})^n y_{n+1} + \sqrt{t}\sum_{n=2}^{\infty}(n+2)(-2a\sqrt{t})^n y_{n+2}\right]$$

$$-\frac{1}{a^2}\mathrm{erf}(cT)\left[4t\sum_{n=2}^{\infty}(-2a\sqrt{t})^n i^{n+2}\,\mathrm{erfc}(bT)\right] - \frac{4c\sqrt{t}}{a^2\sqrt{\pi}}e^{-X}\sum_{n=2}^{\infty}(-2a\sqrt{t})^n y_{n+2}$$

The final result for $I_{24}(a,b,c,t)$ is (by replacing n by n+2 and summing from n=0)

$$I_{24}(a,b,c,t) = tV(t)\mathrm{erf}\left(\frac{c}{\sqrt{t}}\right) - 16t^2\mathrm{erf}\left(\frac{c}{\sqrt{t}}\right)\sum_{n=0}^{\infty}(-2a\sqrt{t})^n i^{n+4}\mathrm{erfc}(bT)$$

$$+\frac{8cte^{-X}}{\sqrt{\pi}}\left[b\sum_{n=0}^{\infty}(-2a\sqrt{t})^n y_{n+3} + \sqrt{t}\sum_{n=0}^{\infty}(n+2)(-2a\sqrt{t})^n y_{n+4}\right]$$

and

$$I_{24}^c(a,b,c,t) = tV(t)\mathrm{erfc}\left(\frac{c}{\sqrt{t}}\right) - 16t^2\mathrm{erfc}\left(\frac{c}{\sqrt{t}}\right)\sum_{n=0}^{\infty}(-2a\sqrt{t})^n i^{n+4}\mathrm{erfc}(bT)$$

$$-\frac{8cte^{-X}}{\sqrt{\pi}}\left[b\sum_{n=0}^{\infty}(-2a\sqrt{t})^n y_{n+3} + \sqrt{t}\sum_{n=0}^{\infty}(n+2)(-2a\sqrt{t})^n y_{n+4}\right]$$

Now, taking the limit for $c \to \infty$ in $I_{24}(a,b,c,t)$ or $c \to 0$ in $I_{24}^c(a,b,c,t)$ gives ($X = (b^2+c^2)/t$)

$$J_{24}(a,b,t) = tV(t) - 16t^2\sum_{n=0}^{\infty}(-2a\sqrt{t})^n i^{n+4}\,\mathrm{erfc}(bT)$$

and since

$$J_{24}(a,b,t) = tV(t) - \int_0^t V(\tau)d\tau$$

we get

$$V_{24}(a,b,t) = \int_0^t V(\tau)d\tau = tV(t) - J_{24}(a,b,t) = 16t^2 \sum_{n=0}^{\infty} (-2a\sqrt{t})^n i^{n+4} \operatorname{erfc}(bT)$$

Again, the $y_n(c,bT)$ sequence is derived in Folder 21 and calculated in double precision subroutine GNSEQ,

$$y_n(c,b,T) = 2T^{n-1} e^X \int_T^{\infty} e^{-c^2w^2} \frac{i^n \operatorname{erfc}(bw)}{w^n} dw.$$

Then, combining forms, we have

$$I_{24}(a,b,c,t) = J_{24}(a,b,t) \operatorname{erf}\left(\frac{c}{\sqrt{t}}\right) + \frac{8cte^{-X}}{\sqrt{\pi}} \left[b\sum_{n=0}^{\infty} (-2a\sqrt{t})^n y_{n+3} + \sqrt{t}\sum_{n=0}^{\infty} (n+2)(-2a\sqrt{t})^n y_{n+4} \right].$$

$$I_{24}^c(a,b,c,t) = J_{24}(a,b,t) \operatorname{erfc}\left(\frac{c}{\sqrt{t}}\right) - \frac{8cte^{-X}}{\sqrt{\pi}} \left[b\sum_{n=0}^{\infty} (-2a\sqrt{t})^n y_{n+3} + \sqrt{t}\sum_{n=0}^{\infty} (n+2)(-2a\sqrt{t})^n y_{n+4} \right]$$

Direct Evaluation of V_{24}

From Folder 21 we have

$$V(a,b,t) = \frac{2}{a}\sqrt{\frac{t}{\pi}} e^{-b^2/t} - \left(\frac{1}{a^2} + \frac{2b}{a}\right) \operatorname{erfc}\left(\frac{b}{\sqrt{t}}\right) + \frac{1}{a^2} U(a,b,t)$$

Then

$$V_{24}(a,b,t) = \int_0^t V(a,b,\tau)d\tau = \frac{2}{a\sqrt{\pi}} \int_0^t \sqrt{\tau} e^{-b^2/\tau} d\tau - \left(\frac{1}{a^2} + \frac{2b}{a}\right) \int_0^t \operatorname{erfc}\left(\frac{b}{\sqrt{\tau}}\right) d\tau + \frac{V(a,b,t)}{a^2}$$

$$= \frac{2t^{3/2}}{a\sqrt{\pi}} E_{5/2}\left(\frac{b^2}{t}\right) - 2\left(\frac{1}{a^2} + \frac{2b}{a}\right) \int_T^{\infty} \frac{\operatorname{erfc}(bw)}{w^3} dw + \frac{1}{a^2} V(a,b,t)$$

where we used the change of variables $\tau = t/v$ in the first integral and $\tau = 1/w^2$ in the second integral. To evaluate the second integral, we integrate by parts

$$u = \operatorname{erfc}(bw) \qquad dv = dw/w^3$$

$$du = -\frac{2b}{\sqrt{\pi}} e^{-b^2w^2} \qquad v = -\frac{1}{2w^2}$$

and from Folder 10,

$$\int_T^\infty \frac{\text{erfc}(bw)}{w^3}dw = J_3^c(b,T) = \frac{2}{T^2} i^2\text{erfc}(bT)\,, \qquad T = \frac{1}{\sqrt{t}}\,,$$

Now, applying the recurrence for $E_\nu(x)$ for $\nu = 3/2$, we get

$$E_{5/2}\left(\frac{b^2}{t}\right) = \frac{2}{3}\left[e^{-b^2/t} - \frac{b^2}{t}E_{3/2}\left(\frac{b^2}{t}\right)\right], \qquad E_{3/2}\left(\frac{b^2}{t}\right) = 2\sqrt{\pi}\ \text{ierfc}\left(\frac{b}{\sqrt{t}}\right), \qquad T = \frac{1}{\sqrt{t}}\,,$$

and combining terms, we have

$$V_{24}(a,b,t) = \frac{V(a,b,t)}{a^2} + \frac{4}{3aT^3\sqrt{\pi}}\left[e^{-b^2T^2} - 2b^2T^2\sqrt{\pi}\ \text{ierfc}(bT)\right] - \frac{4}{T^2}\left(\frac{1}{a^2} + \frac{2b}{a}\right)i^2\text{erfc}(bT)$$

Since i^nerfc(bT) and $E_{n+1/2}(b^2T^2)$ can be computed with significant digits in subroutines DINERFC and DHEXINT, the preferred computational form is

$$V_{24}(a,b,t) = \frac{1}{a^2}V(a,b,t) + \frac{2}{aT^3\sqrt{\pi}}E_{5/2}(b^2T^2) - 4t\left(\frac{1}{a^2} + \frac{2b}{a}\right)i^2\ \text{erfc}(bT)\,.$$

If we use the series for V(a,b,t) in terms of $a\sqrt{t}$, we get the expression in the previous section. Since V_{24} is analytic in a, the reciprocal powers in a^{-2} and a^{-1} must vanish. That is, the coefficients must be zero. However, the coefficient of a^{-1} yields a bonus,

$$\frac{2}{T^3\sqrt{\pi}}E_{5/2}(b^2T^2) - \frac{8b}{T^2}i^2 erfc(bT) - \frac{8}{T^3}i^3 erfc(bT) = 0$$

and we get another computational form

$$V_{24}(a,b,t) = \frac{1}{a^2}V(a,b,t) + \frac{8}{aT^3}i^3 erfc(bT) - \frac{4}{a^2T^2}i^2 erfc(bT)$$

along with the relation

$$\frac{1}{\sqrt{\pi}}E_{5/2}(x^2) = 4x\,i^2 erfc(x) + 4\,i^3 erfc(x)\,.$$

Similar forms in terms of iterated co-error functions were used above,

$$\frac{1}{\sqrt{\pi}}E_{3/2}(x^2) = 2\ ierfc(x)\,, \qquad \frac{1}{\sqrt{\pi}}E_{1/2}(x^2) = \frac{erfc(x)}{x}\ .$$

Another form for $J_{24}(a,b,t)$ is obtained by using $E_{5/2}$ above or $J_{24}(a,b,t) = tV(t) - V_{24}(a,b,t)$

$$J_{24}(a,b,t) = \left(t - \frac{1}{a^2}\right)V(t) - \frac{8}{aT^3}i^3 erfc(bT) + \frac{4}{a^2T^2}i^2 erfc(bT)\,.$$

Folder 25

Evaluation of

$$\mathbf{I_{25}(a,b,c,d,t) = \int_0^t \; U(a,b,\tau)\,U(c,d,\tau)\,d\tau}$$

$$\mathbf{U(a,b,t) = e^{a^2t+2ab}\,erfc(a\sqrt{t} + b/\sqrt{t})}$$

$$\boldsymbol{a \ge 0, \quad b \ge 0, \quad c \ge 0, \quad d \ge 0, \quad t > 0}$$

Donald E. Amos, April 2003

Summary

This integral is manipulated to produce a formula which expresses I_{25} in terms of functions from Folders 5, 10, 21 and 22. The non-symmetric formula is

$$I_{25}(a,b,c,d,t) = \frac{1}{(a^2+c^2)}\Bigg\{a^2U(c,d,t)V(a,b,t) + \frac{c}{\sqrt{\pi}}R_1(a,b,d,t) - \frac{d}{\sqrt{\pi}}R_2(a,b,d,t)$$
$$-\frac{2ac^2}{\sqrt{\pi}}J_{22}(c,d,b,t) + c^2(1+2ab)I_{21}^c(c,d,b,t) + \frac{c}{\sqrt{\pi}}I_{22}(a,b,d,t) - \frac{d}{\sqrt{\pi}}J_{21}(a,b,d,t)\Bigg\}$$

where

$$V(a,b,t) = \frac{2}{a}\sqrt{\frac{t}{\pi}}e^{-b^2/t} - \frac{(1+2ab)}{a^2}\operatorname{erfc}\left(\frac{b}{\sqrt{t}}\right) + \frac{1}{a^2}U(a,b,t)$$

$$R_1(a,b,d,t) = \frac{2at}{\sqrt{\pi}}E_2(X) - 2(1+2ab)I_1^c(d,b,T)$$

$$R_2(a,b,d,t) = \frac{2a}{\sqrt{\pi}}E_1(X) - 2(1+2ab)I_5(d,b,T)$$

$$T = \frac{1}{\sqrt{t}}, \qquad\qquad X = (b^2+d^2)/t.$$

Here, I_5 and I_1^c are computed in Folders 5 and 10. V, J_{21} I_{21}^c, I_{22} and J_{22} are the principal results of Folders 21 and 22. Notice also that I_{25} is symmetric in the pairs (a,b) and (c,d) and exchanging these pairs on the right yields an alternate form. In fact, any convex linear combination of these forms gives I_{25}; in particular, adding and dividing by 2 gives the symmetric form. If c=0 we get the formula for I_{21}^c in Folder 21,

$$I_{25}(a,b,0,d,t) = I_{21}^c(a,b,d,t)\,.$$

Derivation

$$I_{25}(a,b,c,d,t) = \int_0^t U(a,b,\tau)U(c,d,\tau)d\tau$$

$$U(a,b,t) = e^{a^2t+2ab}\ \mathrm{erfc}(a\sqrt{t} + b/\sqrt{t})$$

Integrate by parts:

$u = U(c,d,\tau)$ $\qquad\qquad$ $dv = U(a,b,\tau)d\tau$

$du = U'(c,d,\tau)d\tau$ $\qquad\qquad$ $v = V(a,b,\tau)$

where

$$U'(c,d,\tau) = -\frac{2}{\sqrt{\pi}}\left(\frac{c}{2\sqrt{\tau}} - \frac{d}{2\tau^{3/2}}\right)e^{-d^2/\tau} + c^2 U(c,d,\tau)$$

and

$$V(a,b,\tau) = \int_0^\tau U(a,b,w)dw = \frac{2}{a}\sqrt{\frac{\tau}{\pi}}e^{-b^2/\tau} - \frac{(1+2ab)}{a^2}\mathrm{erfc}\left(\frac{b}{\sqrt{\tau}}\right) + \frac{U(a,b,\tau)}{a^2}$$

is computed in Folder 21. It only remains to evaluate the appropriate integrals in

$$I_{25}(a,b,c,d,t) = U(c,d,t)V(a,b,t) + \frac{c}{\sqrt{\pi}}\int_0^t \frac{V(a,b,\tau)}{\sqrt{\tau}}e^{-d^2/\tau}d\tau$$

$$-\frac{d}{\sqrt{\pi}}\int_0^t \frac{V(a,b,\tau)}{\tau^{3/2}}e^{-d^2/\tau}d\tau - c^2\int_0^t V(a,b,\tau)U(c,d,\tau)d\tau$$

Now,

$$\underline{\underline{\int_0^t \frac{V(a,b,\tau)}{\sqrt{\tau}}e^{-d^2/\tau}d\tau}} = \frac{2}{a\sqrt{\pi}}\int_0^t e^{-(b^2+d^2)/\tau}d\tau - \frac{(1+2ab)}{a^2}\int_0^t \frac{e^{-d^2/\tau}}{\sqrt{\tau}}\mathrm{erfc}\left(\frac{b}{\sqrt{\tau}}\right)d\tau$$

$$+\frac{1}{a^2}\int_0^t \frac{U(a,b,\tau)}{\sqrt{\tau}}e^{-d^2/\tau}d\tau$$

$$= \frac{2t}{a\sqrt{\pi}}E_2(X) - 2\frac{(1+2ab)}{a^2}\int_T^\infty \frac{e^{-d^2w^2}\ \mathrm{erfc}(bw)}{w^2}dw + \frac{1}{a^2}I_{22}(a,b,d,t)$$

and finally,

$$\int_0^t \frac{V(a,b,\tau)}{\sqrt{\tau}} e^{-d^2/\tau} d\tau = \frac{2t}{a\sqrt{\pi}} E_2(X) - \frac{2(1+2ab)}{a^2} I_1^c(d,b,T) + \frac{1}{a^2} I_{22}(a,b,d,t)$$

where $T = 1/\sqrt{t}$, $X = (b^2+d^2)/t$ and I_1^c is computed in Folder 10. I_{22} is computed in Folder 22.

The next integral to be computed is

$$\underline{\underline{\int_0^t \frac{V(a,b,\tau)}{\tau^{3/2}} e^{-d^2/\tau} d\tau}} = \frac{2}{a\sqrt{\pi}} \int_0^t \frac{e^{-(b^2+d^2)/\tau}}{\tau} d\tau - \frac{(1+2ab)}{a^2} \int_0^t \frac{e^{-d^2/\tau}}{\tau^{3/2}} \operatorname{erfc}\left(\frac{b}{\sqrt{\tau}}\right) d\tau$$

$$+ \frac{1}{a^2} \int_0^t \frac{U(a,b,\tau)}{\tau^{3/2}} e^{-d^2/\tau} d\tau$$

$$= \frac{2}{a\sqrt{\pi}} E_1(X) - \frac{2(1+2ab)}{a^2} \int_T^\infty e^{-d^2w^2} \operatorname{erfc}(bw) dw + \frac{1}{a^2} J_{21}(a,b,d,t)$$

$$= \frac{2}{a\sqrt{\pi}} E_1(X) - \frac{2(1+2ab)}{a^2} I_5(d,b,T) + \frac{1}{a^2} J_{21}(a,b,d,t)$$

where I_5 is the I integral of Folder 5 and J_{21} is computed in Folder 21.

The last integral to be computed is

$$\underline{\underline{\int_0^t V(a,b,\tau)U(c,d,\tau)d\tau}} = \frac{2}{a\sqrt{\pi}} \int_0^t U(c,d,\tau) e^{-b^2/\tau} \sqrt{\tau} d\tau - \frac{(1+2ab)}{a^2} \int_0^t U(c,d,\tau) \operatorname{erfc}\left(\frac{b}{\sqrt{\tau}}\right) d\tau$$

$$+ \frac{1}{a^2} \int_0^t U(a,b,\tau)U(c,d,\tau)d\tau$$

$$= \frac{2}{a\sqrt{\pi}} J_{22}(c,d,b,t) - \frac{(1+2ab)}{a^2} I_{21}^c(c,d,b,t) + \frac{1}{a^2} I_{25}(a,b,c,d,t)$$

Notice that I_{25} appears on both the left and right sides of our basic relation. Combining terms gives

$$\left(1+\frac{c^2}{a^2}\right)I_{25}(a,b,c,d,t)=U(c,d,t)V(a,b,t)+$$

$$+\frac{c}{\sqrt{\pi}}\widetilde{R}_1(a,b,d,t)-\frac{d}{\sqrt{\pi}}\widetilde{R}_2(a,b,d,t)$$

$$-\frac{2c^2}{a\sqrt{\pi}}J_{22}(c,d,b,t)+c^2\,\frac{(1+2ab)}{a^2}I_{21}^c(c,d,b,t)$$

$$+\frac{c}{a^2\sqrt{\pi}}I_{22}(a,b,d,t)-\frac{d}{a^2\sqrt{\pi}}J_{21}(a,b,d,t)$$

The final result is

$$I_{25}(a,b,c,d,t)=\frac{1}{(a^2+c^2)}\{a^2U(c,d,t)V(a,b,t)+\frac{c}{\sqrt{\pi}}R_1(a,b,d,t)-\frac{d}{\sqrt{\pi}}R_2(a,b,d,t)$$

$$-\frac{2ac^2}{\sqrt{\pi}}J_{22}(c,d,b,t)+c^2(1+2ab)I_{21}^c(c,d,b,t)$$

$$+\frac{c}{\sqrt{\pi}}I_{22}(a,b,d,t)-\frac{d}{\sqrt{\pi}}J_{21}(a,b,d,t)\}$$

where we define

$$R_1(a,b,d,t)=a^2\widetilde{R}_1(a,b,d,t)=a^2\left[\frac{2t}{a\sqrt{\pi}}E_2(X)-2\frac{(1+2ab)}{a^2}I_1^c(d,b,T)\right]$$

$$=\frac{2at}{\sqrt{\pi}}E_2(X)-2(1+2ab)I_1^c(d,b,T)$$

$$\mathrm{R}_2(\mathrm{a,b,d,t})=\mathrm{a}^2\widetilde{\mathrm{R}}_2(\mathrm{a,b,d,t})=\mathrm{a}^2\left[\frac{2}{\mathrm{a}\sqrt{\pi}}\mathrm{E}_1(\mathrm{X})-2\frac{(1+2\mathrm{ab})}{\mathrm{a}^2}\mathrm{I}_5(\mathrm{d,b,T})\right]$$

$$=\frac{2a}{\sqrt{\pi}}E_1(X)-2(1+2ab)I_5(d,b,T)$$

$$T=\frac{1}{\sqrt{t}},\qquad X=(b^2+d^2)/t$$

As we noted before, I_5 and I_1^c are computed in Folders 5 and 10. V, J_{21}, I_{21}^c, I_{22} and J_{22} are computed in Folders 21 and 22.

Notice that I_{25} is symmetric in the pairs (a,b) and (c,d). Thus, if we exchange these pairs in the formula above, add and divide by 2, we get the symmetric form for I_{25}. In fact, any convex linear combination of these two forms

$$I_{25} = \alpha I_{25}(a,b,c,d,t)+(1-\alpha)I_{25}(c,d,a,b,t) \qquad 0 \le \alpha \le 1$$

is I_{25}. $\alpha = ½$ gives the symmetric form. If c=0 we get

$$I_{25}(a,b,0,d,t) = I_{21}^c(a,b,d,t)\,.$$

Folder 26

Evaluation of

$$\boldsymbol{I_{26}(a,b,c,d,t)=\int_0^t \tau U(a,b,\tau)U(c,d,\tau)d\tau}$$

$$\boldsymbol{U(a,b,t)=e^{a^2t+2ab}\,\mathrm{erfc}(a\sqrt{t}+b/\sqrt{t})}$$

$$\boldsymbol{a\geq 0,\quad b\geq 0,\quad c\geq 0,\quad d\geq 0,\quad t>0}$$

Donald E. Amos, April, 2003

Summary

The integral for I_{26} is developed in two ways. The first way leads to the (recommended) symmetric form (in the pairs (a,b) and (c,d))

$$\begin{aligned}I_{26}(a,b,c,d,t)=\frac{1}{(a^2+c^2)}&\{tU(a,b,t)U(c,d,t)-I_{25}(a,b,c,d,t)\\&+\frac{a}{\sqrt{\pi}}J_{22}(c,d,b,t)-\frac{b}{\sqrt{\pi}}I_{22}(c,d,b,t)\\&+\frac{c}{\sqrt{\pi}}J_{22}(a,b,d,t)-\frac{d}{\sqrt{\pi}}I_{22}(a,b,d,t)\}\end{aligned}$$

where I_{22}, J_{22} and I_{25} are defined in Folders 22 and 25. A more efficient computational form is also presented in terms of more fundamental integrals which eliminates redundant computation.

An alternate, non-symmetric, more complicated form is also derived.

Symmetric Form In this case, we integrate $I_{25}(a,b,c,d,t)$ by parts in a way which retains symmetry:

$$I_{25}(a,b,c,d,t)=\int_0^t U(a,b,\tau)U(c,d,\tau)d\tau$$

$$u=U(a,b,\tau)U(c,d,\tau) \qquad dv=d\tau$$

$$du=[U(a,b,\tau)U'(c,d,\tau)+U(c,d,\tau)U'(a,b,\tau)]d\tau \qquad v=\tau$$

where

$$U'(a,b,\tau)=-\frac{1}{\sqrt{\pi}}\left(\frac{a}{\sqrt{t}}-\frac{b}{\tau^{3/2}}\right)e^{-b^2/\tau}+a^2U(a,b,\tau)$$

and

$$du = \left\{ -\frac{U(a,b,\tau)}{\sqrt{\pi}} \left(\frac{c}{\sqrt{\tau}} - \frac{d}{\tau^{3/2}} \right) e^{-d^2/\tau} + c^2 U(a,b,\tau) U(c,d,\tau) \right.$$

$$\left. - \frac{U(c,d,\tau)}{\sqrt{\pi}} \left(\frac{a}{\sqrt{\tau}} - \frac{b}{\tau^{3/2}} \right) e^{-b^2/\tau} + a^2 U(a,b,\tau) U(c,d,\tau) \right\} d\tau$$

Then

$$I_{25}(a,b,c,d,t) \equiv tU(a,b,t)U(c,d,t)$$

$$+ \frac{c}{\sqrt{\pi}} \int_0^t \sqrt{\tau} U(a,b,\tau) e^{-d^2/\tau} - \frac{d}{\sqrt{\pi}} \int_0^t U(a,b,\tau) \frac{e^{-d^2/\tau}}{\sqrt{\tau}} d\tau$$

$$+ \frac{\mathrm{a}}{\sqrt{\pi}} \int_0^{\mathrm{t}} \sqrt{\tau} \mathrm{U}(\mathrm{a},\mathrm{c},\tau) \mathrm{e}^{-\mathrm{b}^2/\tau} - \frac{\mathrm{b}}{\sqrt{\pi}} \int_0^{\mathrm{t}} \mathrm{U}(\mathrm{c},\mathrm{d},\tau) \frac{\mathrm{e}^{-\mathrm{b}^2/\tau}}{\sqrt{\tau}} d\tau$$

$$-(a^2+c^2) I_{26}(a,b,c,d,t)$$

Identification of these integrals from Folder 22 gives

$$I_{26}(a,b,c,d,t) = \frac{1}{(a^2+c^2)} \{ tU(a,b,t)U(c,d,t) - I_{25}(a,b,c,d,t)$$

$$+ \frac{a}{\sqrt{\pi}} J_{22}(c,d,b,t) - \frac{b}{\sqrt{\pi}} I_{22}(c,d,b,t)$$

$$+ \frac{c}{\sqrt{\pi}} J_{22}(a,b,d,t) - \frac{d}{\sqrt{\pi}} I_{22}(a,b,d,t) \}$$

Notice that I_{22} and J_{22} from Folder 22 can be expressed in terms of more fundamental integrals S_1 and S_2, which, timewise, dominate the computation in subroutines INTEGS1 and INTEGS2:

$$I_{22}(a,b,c,t) = \frac{1}{a\sqrt{\pi}} E_1(X) - \frac{2\sqrt{t}}{a\sqrt{\pi}} S_2(a,b,c,t), \qquad X = (b^2+c^2)/t$$

$$\mathrm{J}_{22}(\mathrm{a},\mathrm{b},\mathrm{c},\mathrm{t}) = \frac{\mathrm{t}}{\mathrm{a}\sqrt{\pi}} \mathrm{E}_2(\mathrm{X}) - \frac{1}{\mathrm{a}^2\sqrt{\pi}} \left(\mathrm{b} + \frac{1}{2\mathrm{a}} \right) \mathrm{E}_1(\mathrm{X}) + \frac{\sqrt{\mathrm{t}}}{\mathrm{a}^2} \mathrm{e}^{-\mathrm{X}} [\mathrm{e}^{\mathrm{B}^2} \mathrm{erfc}(\mathrm{B})] -$$

$$-\frac{2c^2\sqrt{t}}{a^2\sqrt{\pi}}S_1(a,b,c,t)+\frac{\sqrt{t}}{a^3\sqrt{\pi}}S_2(a,b,c,t)\,, \qquad B=a\sqrt{t}+b/\sqrt{t}$$

Notice that E_1 and S_2 are shared by I_{22} and J_{22} for common values of a,b,c,t. It follows that by using these formulae directly (as opposed to separate subroutines for I_{22} and J_{22} like INTEGI22 and INTEGJ22) redundant computation of E_1 and S_2 functions can be eliminated. Subroutine DEXINT computes the E sequence in one call. With these modifications one can expect to save at least 1/3 in computational time.

Alternate, Non-Symmetric Form for I_{26}

If we proceed to integrate I_{26} in the manner

$$I_{26}(a,b,c,d,t)=\int_0^t \tau U(a,b,\tau)U(c,d,\tau)d\tau$$

$u=\tau U(c,d,\tau)$ $\qquad$ $dv=Ua,b,\tau)d\tau$

$du=[\tau U'(c,d,\tau)+U(c,d,\tau)]d\tau$ $\qquad$ $v=V(a,b,\tau)$

with

$$U'(c,d,\tau)=-\frac{1}{\sqrt{\pi}}\left(\frac{c}{\sqrt{\tau}}-\frac{d}{\tau^{3/2}}\right)e^{-d^2/\tau}+c^2U(c,d,\tau)$$

and (from Folder 21)

$$V(a,b,\tau)=\int_0^\tau U(a,b,w)dw=\frac{2\sqrt{\tau}}{a\sqrt{\pi}}e^{-b^2/\tau}-\frac{(1+2ab)}{a^2}\operatorname{erfc}\left(\frac{b}{\sqrt{\tau}}\right)+\frac{U(a,b,\tau)}{a^2}\,,$$

then we can write I_{26} as

$$I_{26}(a,b,c,d,t)=tU(c,d,t)V(a,b,t)+\frac{c}{\sqrt{\pi}}R_1(a,b,d,t)-\frac{d}{\sqrt{\pi}}R_2(a,b,d,t)$$

$$-R_3(a,b,c,d,t)-c^2R_4(a,b,c,d,t)$$

where we have to evaluate the integrals

$$R_1(a,b,d,t)=\int_0^t V(a,b,\tau)e^{-d^2/\tau}\sqrt{\tau}d\tau\,, \qquad R_2(a,b,d,t)=\int_0^t V(a,b,\tau)\frac{e^{-d^2/\tau}}{\sqrt{\tau}}d\tau$$

$$R_3(a,b,d,t)=\int_0^t V(a,b,\tau)U(c,d,\tau)d\tau\,, \qquad R_4(a,b,d,t)=\int_0^t \tau V(a,b,\tau)U(c,d,\tau)d\tau\,.$$

In each of these integrals, $V(a,b,\tau)$ appears in the integrand and, when expanded out according to the formula above, yields 3 terms. The primary integrals are designated with subscripts 1,2,3 and 4. Each succeeding integral resulting form one of the three terms of V are given an additional subscript to link the term with the primary integral. Thus,

$$\underline{R_1(a,b,d,t)} = \int_0^t V(a,b,\tau)e^{-d^2/\tau}\sqrt{\tau}d\tau$$

$$= \frac{2a}{a^2\sqrt{\pi}}R_{11}(b,d,t) - \frac{(1+2ab)}{a^2}R_{12}(d,b,t) + \frac{1}{a^2}R_{13}(a,b,d,t)$$

where

$$R_{11}(b,d,t) = \int_0^t e^{-(b^2+d^2)/\tau}\tau d\tau = t^2E_3(X), \qquad X = (b^2+d^2)/t$$

$$R_{12}(d,b,t) = \int_0^t e^{-d^2/\tau}\sqrt{\tau}\,\mathrm{erfc}\left(\frac{b}{\sqrt{\tau}}\right)d\tau = 2\int_T^\infty \frac{e^{-d^2w^2}\,\mathrm{erfc}(bw)}{w^4}dw, \quad T = \frac{1}{\sqrt{t}},$$

$$R_{13}(a,b,d,t) = \int_0^t U(a,b,\tau)e^{-d^2/\tau}\sqrt{\tau}d\tau = J_{22}(a,b,d,t) \text{ from Folder 22.}$$

Since R_{11} and R_{13} are identified, it only remains to identify R_{12}. We integrate R_{12} by parts,

$$u = e^{-d^2w^2}\,\mathrm{erfc}(bw) \qquad dv = dw/w^4$$

$$du = \left[-\frac{2b}{\sqrt{\pi}}e^{-(b^2+d^2)w^2} - 2d^2we^{-d^2w^2}\,\mathrm{erfc}(bw)\right]dw \qquad v = -\frac{1}{3w^3}$$

and

$$\mathrm{R}_{12}(\mathrm{d,b,t}) = 2\left[\frac{\mathrm{e}^{-\mathrm{d}^2\mathrm{T}^2}\mathrm{erfc(bT)}}{3\mathrm{T}^3} - \frac{2\mathrm{b}}{3\sqrt{\pi}}\int_\mathrm{T}^\infty \frac{\mathrm{e}^{-(\mathrm{b}^2+\mathrm{d}^2)\mathrm{w}^2}}{\mathrm{w}^3}\mathrm{dw} - \frac{2\mathrm{d}^2}{3}\int_\mathrm{T}^\infty \frac{\mathrm{e}^{-\mathrm{d}^2\mathrm{w}^2}\mathrm{erfc(bw)}}{\mathrm{w}^2}\mathrm{dw}\right]$$

$$= 2\left[\frac{e^{-d^2T^2}\,\mathrm{erfc}(bT)}{3T^3} - \frac{b}{3T^2\sqrt{\pi}}E_2(X) - \frac{2d^2}{3}I_1^c(d,b,T)\right]$$

where I_1^c is computed in Folder 10.

To continue, $R_2(a,b,d,t)$ was evaluated in Folder 25 and R_{21}, R_{22}, R_{23} are readily identified in the formula

$$\underline{R_2(a,b,d,t)} = \frac{2at}{a^2\sqrt{\pi}} E_2(X) - 2\frac{(1+2ab)}{a^2} I_1^c(d,b,T) + \frac{1}{a^2} I_{22}(a,b,d,t)\,, \quad X = (b^2+d^2)/t$$

For $\quad R_3(a,b,c,d,t) = \int_0^t V(a,b,\tau)U(c,d,\tau)d\tau$ we have

$$\underline{R_3(a,b,c,d,t)} = \frac{2a}{a^2\sqrt{\pi}} R_{31}(c,d,b,t) - \frac{(1+2ab)}{a^2} R_{32}(c,d,b,t) + \frac{1}{a^2} R_{33}(a,b,c,d,t)$$

where

$$R_{31}(c,d,b,t) = \int_0^t U(c,d,\tau)e^{-b^2/\tau}\sqrt{\tau}d\tau = J_{22}(c,d,b,t)$$

$$R_{32}(c,d,b,t) = \int_0^t U(c,d,\tau)\,\mathrm{erfc}\left(\frac{b}{\sqrt{\tau}}\right)d\tau = I_{21}^c(c,d,b,t)$$

$$R_{33}(a,b,c,d,t) = \int_0^t U(a,b,\tau)U(c,d,\tau)d\tau = I_{25}(a,b,c,d,t)$$

and the subscript indicates the Folder where the function is defined and computed.

For R_4 we have

$$\underline{R_4(a,b,c,d,t)} = \int_0^t \tau V(a,b,\tau)U(c,d,\tau)d\tau$$

$$= \frac{2a}{a^2\sqrt{\pi}} R_{41}(c,d,b,t) - \frac{(1+2ab)}{a^2} R_{42}(c,d,b,t) + \frac{1}{a^2} R_{43}(a,b,c,d,t)$$

where

$$R_{41}(c,d,b,t) = \int_0^t U(c,d,\tau)\tau^{3/2}e^{-b^2/\tau}d\tau$$

$$R_{42}(c,d,b,t) = \int_0^t \tau U(c,d,\tau)\,\mathrm{erfc}\left(\frac{b}{\sqrt{\tau}}\right)d\tau$$

$$R_{43}(a,b,c,d,t) = \int_0^t \tau U(a,b,\tau)U(c,d,\tau)d\tau = I_{26}(a,b,c,d,t)$$

Notice that we have been evaluating I_{26} in terms of the R functions on the right and I_{26} has shown up on the right. In the final step, we will have to solve for I_{26} to complete the analysis. To continue, we evaluate R_{41} and R_{42}.

$$R_{41}(c,d,b,t)=\int_0^t U(c,d,\tau)\tau^{3/2}e^{-b^2/\tau}d\tau$$

and we integrate by parts to obtain

$$u=\tau^{3/2}e^{-b^2/\tau} \qquad\qquad dv = U(c,d,\tau)$$

$$du=\left[\frac{b^2e^{-b^2/\tau}}{\sqrt{\tau}}+\frac{3}{2}\sqrt{\tau}e^{-b^2/\tau}\right]d\tau \qquad\qquad v = V(c,d,\tau)$$

$$R_{41}(c,d,b,t)=t^{3/2}V(c,d,t)e^{-b^2/t}-b^2\int_0^t \frac{V(c,d,\tau)}{\sqrt{\tau}}e^{-b^2/\tau}d\tau-\frac{3}{2}\int_0^t V(c,d,\tau)e^{-b^2/\tau}\sqrt{\tau}d\tau$$

$$=t^{3/2}e^{-b^2/t}V(c,d,t)-b^2R_2(c,d,b,t)-\frac{3}{2}R_{413}(c,d,b,t)$$

where R_{412} has been identified as R_2 above (with different parameters). It remains to expand out R_{413} using the 3 terms of V.

$$R_{413}(c,d,b,t)=\int_0^t V(c,d,\tau)e^{-b^2/\tau}\sqrt{\tau}d\tau$$

$$=\frac{2c}{c^2\sqrt{\pi}}\int_0^t e^{-(b^2+d^2)/\tau}\tau d\tau-\frac{(1+2cd)}{c^2}\int_0^t e^{-b^2/\tau}\sqrt{\tau}\,\mathrm{erfc}\left(\frac{d}{\sqrt{\tau}}\right)d\tau+\frac{1}{c^2}\int_0^t U(c,d,\tau)e^{-b^2/\tau}\sqrt{\tau}d\tau$$

$$=\frac{2\mathrm{ct}^2}{\mathrm{c}^2\sqrt{\pi}}\mathrm{E}_3(\mathrm{X})-\frac{(1+2\mathrm{cd})}{\mathrm{c}^2}\mathrm{R}_{12}(\mathrm{b,d,t})+\frac{1}{\mathrm{c}^2}\mathrm{J}_{22}(\mathrm{c,d,b,t}), \qquad\qquad X=(b^2+c^2)/t$$

While R_{12} was defined previously, these arguments in d and b are the reverse of those in the definition.

Finally, $\quad R_{42}(c,d,b,t)=\int_0^t \tau U(c,d,\tau)\,\mathrm{erfc}\left(\frac{b}{\sqrt{\tau}}\right)d\tau$ and integration by parts yields

$$u=\tau\,\mathrm{erfc}\left(\frac{b}{\sqrt{\tau}}\right) \qquad\qquad dv=U(c,d,\tau)d\tau$$

$$du=\left[\frac{b}{\sqrt{\pi}\sqrt{\tau}}e^{-b^2/\tau}+\mathrm{erfc}\left(\frac{b}{\sqrt{\tau}}\right)\right]d\tau \qquad\qquad v=V(c,d,\tau)$$

$$R_{42}(c,d,b,t) = tV(c,d,t)\,\mathrm{erfc}\left(\frac{b}{\sqrt{t}}\right) - \frac{b}{\sqrt{\pi}}\int_0^t \frac{V(c,d,\tau)}{\sqrt{\tau}} e^{-b^2/\tau} d\tau - \int_0^t V(c,d,\tau)\,\mathrm{erfc}\left(\frac{b}{\sqrt{\tau}}\right) d\tau$$

$$= tV(c,d,t)\,\mathrm{erfc}\left(\frac{b}{\sqrt{t}}\right) - \frac{b}{\sqrt{\pi}} R_2(c,d,b,t) - R_{423}(c,d,b,t)$$

where $R_{423}(c,d,b,t) = \int_0^t V(c,d,\tau)\,\mathrm{erfc}\left(\frac{b}{\sqrt{\tau}}\right) d\tau$

Expanding $V(c,d,\tau)$ out gives

$$\mathrm{R}_{423}(\mathrm{c,d,b,t}) = \frac{2\mathrm{c}}{\mathrm{c}^2\sqrt{\pi}}\int_0^t \mathrm{e}^{-d^2/\tau}\sqrt{\tau}\,\mathrm{erfc}\left(\frac{\mathrm{b}}{\sqrt{\tau}}\right) d\tau - \frac{(1+2\mathrm{cd})}{\mathrm{c}^2}\int_0^t \mathrm{erfc}\left(\frac{\mathrm{d}}{\sqrt{\tau}}\right)\mathrm{erfc}\left(\frac{\mathrm{b}}{\sqrt{\tau}}\right) d\tau$$

$$+ \frac{1}{c^2}\int_0^t U(c,d,\tau)\,\mathrm{erfc}\left(\frac{b}{\sqrt{\tau}}\right) d\tau$$

Then

$$R_{423}(c,d,b,t) = \frac{2c}{c^2\sqrt{\pi}} R_{12}(d,b,T) - 2\frac{(1+2cd)}{c^2}\int_T^\infty \frac{\mathrm{erfc}(bw)\,\mathrm{erfc}(dw)}{w^3} dw$$

$$+ \frac{1}{c^2} I_{21}^c(c,d,b,t)$$

Now the integral

$$W_3^c(b,d,T) = \int_T^\infty \frac{\mathrm{erfc}(bw)\,\mathrm{erfc}(dw)}{w^3} dw$$

is computed in Folder 10. Thus,

$$R_{423}(c,d,b,t) = \frac{2c}{c^2\sqrt{\pi}} R_{12}(d,b,T) - 2\frac{(1+2cd)}{c^2} W_3^c(b,d,T) + \frac{1}{c^2} I_{21}^c(c,d,b,t)$$

Now, all terms of I_{26} have been identified in terms of functions which are computable from previous Folders. Solving for I_{26} (I_{26} occurs on the right in R_{43}) we get

$$I_{26}(a,b,c,d,t) = \frac{a^2}{a^2+c^2}\Big\{tU(c,d,t)V(a,b,t) + \frac{c}{\sqrt{\pi}} R_1(a,b,d,t) - \frac{d}{\sqrt{\pi}} R_2(a,b,d,t)$$

$$- R_3(a,b,c,d,t) - c^2\widetilde{R}_4(a,b,c,d,t)\Big\}$$

where $\widetilde{R}_4$ is R_4 without the $I_{26}(a,b,c,d,t)/a^2$ term,

$$\widetilde{R}_4(a,b,c,d,t) = \frac{2a}{a^2\sqrt{\pi}} R_{41}(c,d,b,t) - \frac{(1+2ab)}{a^2} R_{42}(c,d,b,t)$$

Computationally, we distribute a^2 and c^2 among the appropriate terms to cancel out a^2 and c^2 in denominators arising from $V(a,b,\tau)$ and $V(c,d,\tau)$.

Then, we write

$$I_{26}(a,b,c,d,t) = \frac{1}{(a^2+c^2)}\Big\{tU(c,d,t)[a^2V(a,b,t)] + \frac{c}{\sqrt{\pi}}[a^2R_1(a,b,d,t)]$$

$$-\frac{d}{\sqrt{\pi}}[a^2R_2(a,b,d,t)] - [a^2R_3(a,b,c,d,t)]$$

$$-[a^2c^2\widetilde{R}_4(a,b,c,d,t)]\Big\}$$

and

$$[a^2c^2\widetilde{R}_4(a,b,c,d,t)] = \frac{2a}{\sqrt{\pi}}[c^2R_{41}(c,d,b,t)] - (1+2ab)[c^2R_{42}(c,d,b,t)]$$

$$[c^2R_{41}(c,d,b,t)] = t^{3/2}e^{-b^2/t}[c^2V(c,d,t)] - b^2[c^2R_2(c,d,b,t)] - \frac{3}{2}[c^2R_{413}(c,d,b,t)]$$

$$[c^2R_{42}(c,d,b,t)] = t\,\mathrm{erfc}\left(\frac{b}{\sqrt{t}}\right)[c^2V(c,d,t)] - \frac{b}{\sqrt{\pi}}[c^2R_2(c,d,b,t)] - [c^2R_{423}(c,d,b,t)]$$

Since I_{26} is symmetric in the pairs (a,b), (c,d), we can compute $I_{26} = I_{26}(c,d,a,b,t)$ also with this formula. In fact, any convex linear combination of these pairs $I_{26}(a,b,c,d,t)$, $I_{26}(c,d,a,b,t)$ is I_{26},

$$I_{26} = \alpha I_{26}(a,b,c,d,t) + (1-\alpha)I_{26}(c,d,a,b,t), \qquad 0 \le \alpha \le 1.$$

In particular, $\alpha = 1/2$ gives another symmetric form.

Folder 27

Notes on the Evaluation of

$$\mathbf{I_{27}(a,b,t)=\int_0^t \frac{U(a,b,\tau)}{\sqrt{\tau}}d\tau \quad and \quad J_{27}(a,b,t)=\int_0^t U(a,b,\tau)\sqrt{\tau}d\tau}$$

$$\mathbf{U(a,b,t)=e^{a^2t+2ab}erfc(a\sqrt{t}+b/\sqrt{t})}$$

$$\mathbf{a>0,\ \ b>0,\ t>0}$$

Donald E. Amos, February, 2005, April, 2006

Summary

Observe that $\mathrm{I}_{27}(a,b,t)$ and $\mathrm{J}_{27}(a,b,t)$ functions are the $\mathrm{I}_{22}(a,b,c,t)$ and $\mathrm{J}_{22}(a,b,c,t)$. functions of Folder 22 with c=0:

$$\mathrm{I}_{27}(\mathrm{a,b,t})=\int_0^t \frac{\mathrm{U}(\mathrm{a,b},\tau)}{\sqrt{\tau}}\mathrm{d}\tau = I_{22}(a,b,0,t)=\frac{1}{a\sqrt{\pi}}E_1(X)-\frac{2\sqrt{t}}{a\sqrt{\pi}}S_2(a,b,0,t)$$

$$\mathrm{J}_{27}(\mathrm{a,b,t})=\int_0^t \mathrm{U}(\mathrm{a,b},\tau)\sqrt{\tau}\mathrm{d}\tau = J_{22}(a,b,0,t)=\frac{t}{a\sqrt{\pi}}E_2(X)-\frac{1}{a^2\sqrt{\pi}}\left(b+\frac{1}{2a}\right)E_1(X)$$
$$+\frac{\sqrt{t}}{a^2}e^{-X}[e^{B^2}\ \mathrm{erfc}(B)]+\frac{\sqrt{t}}{a^3\sqrt{\pi}}S_2(a,b,0,t)$$

where $S_2(a,b,0,t)$ was defined in Folder 22

$$S_2(a,b,0,t)=e^{-X}\int_0^\infty \frac{e^{-2Bw-w^2}}{(b+w\sqrt{t})}dw, \qquad B=a\sqrt{t}+b/\sqrt{t}\,, \qquad X=b^2/t.$$

Integration of $\mathrm{I}_{27}(\mathrm{a,b,t})$ by parts in two different ways also gives

$$I_{27}(a,b,t)=\frac{V(a,b,t)}{\sqrt{t}}+\frac{1}{a\sqrt{\pi}}E_1(X)-\frac{1}{b}\left(\frac{1}{a^2}+\frac{2b}{a}\right)ierfc(\sqrt{X})+\frac{1}{2a^2}J_{21}(a,b,0,t)$$

and (alternatively $I_{27}(a,b,t)+2a^2J_{27}(a,b,t)$ above)

$$I_{27}(a,b,t)=2U(a,b,t)\sqrt{t}\ +\ \frac{2at}{\sqrt{\pi}}E_2(X)-\frac{2b}{\sqrt{\pi}}E_1(X)-2a^2J_{27}(a,b,t).$$

where, from Folder 21,we have

$$J_{21}(a,b,0,t)=\frac{2\sqrt{t}}{\sqrt{\pi}}S_1(a,b,0,t), \qquad S_1(a,b,0,t)=e^{-X}\int_0^\infty \frac{e^{-2Bw-w^2}}{(b+w\sqrt{t})^2}dw\,.$$

The main thrust in this folder is to change the variable from τ to $u=a\sqrt{\tau}+b/\sqrt{\tau}$ and integrate on u. $u(\tau)$ has a minimum at $\tau=t_m=b/a$ with a value $u(t_m)=u_m=2\sqrt{ab}$. This makes the inverse $\tau(u)$ double valued and we have to integrate on each of the branches for $t\le t_m$ *and* $t>t_m$. The computation for $t\le t_m$ is done by direct substitution, but the computation for $t>t_m$ relies on $I_{27}(a,b,t)=I_{27}(a,b,t_m)+$ the integral on (t_m,t) where $I_{27}(a,b,t_m)$ is to be computed from the case for $t\le t_m$ with t=t_m. The results are somewhat disappointing in that we can compute accurately with errors

$O(10^{-13})$ in about 50 terms of a series for $t \le t_m/10$. This range can be extended with more terms, but the number of terms becomes very large as t moves past $t_m/2$. Thus, $I_{27}(a,b,t_m)$ is out of reach by this method. Nevertheless, the corresponding analysis for the interval (t_m,t) was done. Satisfactory accuracies in this case were obtained only for very small a and b. The conclusion is that useful results for $I_{27}(a,b,t)$ and $J_{27}(a,b,t)$ by this method were obtained only for small t.

Finally, a recurrence,

$$Y_n = R_n - \frac{a^2}{(n+1/2)} Y_{n+1}$$

with $$\mathrm{R_n}(a,b,t) = \frac{t^n}{(n+1/2)}\left(\sqrt{t}\,U(a,b,t) + \frac{at}{\sqrt{\pi}} E_{n+2}\left(\frac{b^2}{t}\right) - \frac{b}{\sqrt{\pi}} E_{n+1}\left(\frac{b^2}{t}\right)\right).$$

is developed by generalizing $I_{27}(a,b,t)$ and $J_{27}(a,b,t)$ to $\mathrm{Y_n} = \int_0^t \tau^{n-1/2} U(a,b,\tau)d\tau$, n=0,1,... In this notation, $\mathrm{I}_{27}(a,b,t) = Y_0(a,b,t)$ and $\mathrm{J}_{27}(a,b,t) = Y_1(a,b,t)$. Power series for both $I_{27}(a,b,t)$ and $J_{27}(a,b,t)$ in $a\sqrt{t}$ is developed by backward recurrence:

$$I_{27}(a,b,t) = Y_0 = \sum_{n=0}^{\infty} \frac{(-1)^n R_n a^{2n}}{(1/2)_n}, \qquad J_{27}(a,b,t) = Y_1 = \sum_{n=1}^{\infty} \frac{(-1)^{n-1} R_n a^{2(n-1)}}{(3/2)_{n-1}}.$$

Numerical experiments gave relative errors $O(10^{-13})$ over ranges where the answers are most significant and decimal place or absolute errors on lower magnitude results for a and b up to 6 and t as high as $1.2\,t_m$.

<u>Relationship between</u> $\mathbf{I_{27}(a,b,t)}$ <u>and</u> $\mathbf{J_{27}(a,b,t)}$

Integration of $\mathrm{I}_{27}(a,b,t)$ by parts gives the relation between these integrals:

$$u = U(a,b,\tau) \qquad\qquad dv = d\tau/\sqrt{\tau}$$

$$du = \left[-\frac{2}{\sqrt{\pi}} e^{-b^2/\tau}\left(\frac{a}{2\sqrt{\tau}} - \frac{b}{2\tau^{3/2}}\right) + a^2 U(a,b,\tau)\right] d\tau \qquad\qquad v = 2\sqrt{\tau}$$

$$I_{27}(a,b,t) = 2U(a,b,t)\sqrt{t} + \frac{2a}{\sqrt{\pi}}\int_0^t e^{-b^2/\tau} d\tau - \frac{2b}{\sqrt{\pi}}\int_0^t \frac{e^{-b^2/\tau}}{\tau} d\tau - 2a^2 J_{27}(a,b,t)$$

$$= 2U(a,b,t)\sqrt{t} + \frac{2at}{\sqrt{\pi}}\int_1^\infty \frac{e^{-b^2 v/t}}{v^2} dv - \frac{2b}{\sqrt{\pi}}\int_1^\infty \frac{e^{-b^2 v/t}}{v} dv - 2a^2 J_{27}(a,b,t)$$

$$= 2U(a,b,t)\sqrt{t} + \frac{2at}{\sqrt{\pi}} E_2\left(\frac{b^2}{t}\right) - \frac{2b}{\sqrt{\pi}} E_1\left(\frac{b^2}{t}\right) - 2a^2 J_{27}(a,b,t).$$

<u>Relationship between</u> $\mathbf{I_{27}(a,b,t)}$ <u>and</u> $\mathbf{J_{21}(a,b,0,t)}$ <u>of Folder 21</u>

We integrate $\mathrm{I_{27}}$(a,b,t) by parts:

$$u = \frac{1}{\sqrt{\tau}} \qquad\qquad dv = U(a,b,\tau)\, d\tau$$

$$du = -\frac{1}{2\tau^{3/2}} d\tau \qquad\qquad v = V(a,b,\tau)$$

to obtain

$$I_{27}(a,b,t) = \frac{\mathrm{V(a,b,t)}}{\sqrt{\mathrm{t}}} + \frac{1}{2}\int_0^t \frac{V(a,b,\tau)}{\tau^{3/2}}d\tau$$

where

$$V(t) = \frac{2}{a}\sqrt{\frac{t}{\pi}}e^{-b^2/t} - \left(\frac{1}{a^2}+\frac{2b}{a}\right)\mathrm{erfc}\left(\frac{b}{\sqrt{t}}\right) + \frac{U(t)}{a^2}.$$

Each of these terms can be integrated to produce

$$I_{27}(a,b,t) = \frac{\mathrm{V(a,b,t)}}{\sqrt{\mathrm{t}}} + \frac{1}{a\sqrt{\pi}}E_1\left(\frac{b^2}{t}\right) - \frac{1}{b}\left(\frac{1}{a^2}+\frac{2b}{a}\right)ierfc\left(\frac{b}{\sqrt{t}}\right) + \frac{1}{2a^2}J_{21}(a,b,0,t)$$

where $J_{21}(a,b,0,t)$ is computed in Folder 21 in terms of S_1

$$J_{21}(a,b,0,t) = \frac{2\sqrt{t}}{\sqrt{\pi}}S_1(a,b,0,t), \qquad S_1(a,b,0,t) = e^{-X}\int_0^\infty \frac{e^{-2Bw-w^2}}{(b+w\sqrt{t})^2}dw.$$

<u>Computation of</u> $\mathbf{I_{27}(a,b,t)}$

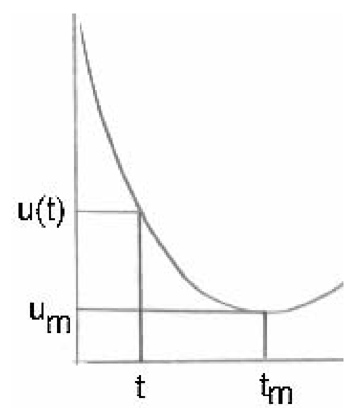
We start by changing variables with $u = a\sqrt{\tau} + b/\sqrt{\tau}$ in

$$\mathrm{I} = \mathrm{e}^{2ab}\int_0^{\mathrm{t}} \frac{e^{a^2\tau}erfc(a\sqrt{\tau}+b/\sqrt{\tau})}{\sqrt{\tau}}d\tau$$

Let $u = a\sqrt{\tau} + b/\sqrt{\tau}$ then $a(\sqrt{\tau})^2 - u\sqrt{\tau} + b = 0$ and

$$\sqrt{\tau} = \frac{u \pm \sqrt{u^2-u_m^2}}{2a}, \qquad \mathrm{u}_{\mathrm{m}}^2 = 4ab.$$

Then, $$\frac{du}{d\tau} = \frac{a}{2\sqrt{\tau}} - \frac{b}{2(\tau)^{3/2}} = \frac{1}{2\sqrt{\tau}}\left(a - \frac{b}{\tau}\right)$$



and $\dfrac{du}{d\tau} = 0$ defines the minimum point on $u = a\sqrt{\tau} + b/\sqrt{\tau}$,

$$t_m = b/a, \quad with\ \ u_m = 2\sqrt{ab} = u(t_m)$$

where $u \to \infty$ *for* $\tau \to 0$ *and* $\tau \to \infty$.

Now we associate the roots of the quadratic equation with the branches $\tau \le t_m$ *and* $\tau > t_m$:

$$\sqrt{\tau} = \frac{u - \sqrt{u^2-u_m^2}}{2a} = \frac{u_m^2}{2a\left(u+\sqrt{u^2-u_m^2}\right)}, \qquad \mathrm{u} \to \infty,\ \tau \to 0 \text{ and this applies for } \tau \le t_m.$$

Similarly,

$$\sqrt{\tau} = \frac{u + \sqrt{u^2-u_m^2}}{2a}, \qquad \mathrm{u} \to \infty,\quad \tau \to \infty \text{ and this applies for } \tau > t_m.$$

Case 1, $\boldsymbol{t \le t_m}$.

$$a^2\tau = \frac{a^2}{4a^2}\left(u-\sqrt{u^2-u_m^2}\right)^2 = \frac{1}{4}\frac{u_m^4}{\left(u+\sqrt{u^2-u_m^2}\right)^2} = \frac{u_m^4/4}{\left(u+\sqrt{u^2-u_m^2}\right)^2}$$

$$I = e^{u_m^2/2}\int_{\infty}^{u(t)}\frac{e^{(u_m^4/4)/\left[u+\sqrt{u^2-u_m^2}\right)^2}\, erfc(u)\; 2\sqrt{\tau}}{\left[a-b/\tau\right)\sqrt{\tau}}du$$

$$I = 2e^{u_m^2/2}\int_{u(t)}^{\infty}\frac{e^{(u_m^4/4)/\left[u+\sqrt{u^2-u_m^2}\right]^2}\, erfc(u)\;\tau}{b-a\tau}du$$

$$b-a\tau = b-\frac{1}{4a}\left(u-\sqrt{u^2-u_m^2}\right)^2 = b-\frac{1}{4a}\left(2u^2-u_m^2-2u\sqrt{u^2-u_m^2}\right)$$

$$b\text{ - }a\tau = \frac{2u_m^2}{4a}-\frac{2u}{4a}\left(u-\sqrt{u^2-u_m^2}\right) \;=\; \frac{u_m^2}{2a}-\frac{u}{2a}\,\frac{u_m^2}{u+\sqrt{u^2-u_m^2}} \;=\; \frac{u_m^2}{2a}\left(1-\frac{u}{u+\sqrt{u^2-u_m^2}}\right)$$

$$=\frac{u_m^2}{2a}\,\frac{\sqrt{u^2-u_m^2}}{u+\sqrt{u^2-u_m^2}}, \qquad u>u_m.$$

Then,
$$I = \frac{u_m^2 e^{u_m^2/2}}{a}\int_{u(t)}^{\infty}\frac{e^{(u_m^4/4)/\left(u+\sqrt{u^2-u_m^2}\right)^2}\, erfc(u)}{\sqrt{u^2-u_m^2}\left(u+\sqrt{u^2-u_m^2}\right)}du$$

Let u=u_m v. Then

$$I = \frac{u_m e^{u_m^2/2}}{a}\int_{\frac{u(t)}{u_m}}^{\infty}\frac{e^{(u_m^2/4)/\left(v+\sqrt{v^2-1}\right)^2}\, erfc(u_m v)}{\sqrt{v^2-1}\left(v+\sqrt{v^2-1}\right)}dv$$

$$= \frac{u_m e^{u_m^2/2}}{a}\int_{\frac{u(t)}{u_m}}^{\infty}\frac{e^{(u_m^2/4)/\left[v^2\left(1+\sqrt{1-1/v^2}\right)^2\right]} erfc(u_m v)}{v^2\sqrt{1-1/v^2}\left(1+\sqrt{1-1/v^2}\right)}dv$$

$$I=\frac{u_m e^{u_m^2/2}}{a}\sum_{n=0}^{\infty}\left(\frac{u_m^2}{4}\right)^n\frac{1}{n!}\int_{\frac{u(t)}{u_m}}^{\infty}\frac{erfc(u_m v)dv}{v^{2n+2}\sqrt{1-1/v^2}\left(1+\sqrt{1-1/v^2}\right)^{2n+1}}$$

Now, HTF, Vol. 1, p101 or A&S Chapter 15 , Eq. 15.1.14 gives

$$\frac{\left(1+\sqrt{1-z}\right)^{1-2a}}{2^{1-2a}} = \sqrt{1-z}F\left(a,a+1/2,2a,z\right)$$

and .

$$1-2a=-2n-1, \qquad a=n+1, \qquad z=1/v^2$$

Then, we get for $u(t) > u_m = 2\sqrt{ab}$

$$I = \frac{u_m e^{u_m^2/2}}{2a} \sum_{n=0}^{\infty} \left(\frac{u_m^2}{16} \right)^n \frac{1}{n!} \int_{\frac{u(t)}{u_m}}^{\infty} F(n+1, n+3/2, 2n+2, \frac{1}{v^2}) \frac{erfc(u_m v) dv}{v^{2n+2}}$$

and with the hypergeometric series,

$$F(a,b,c,z) = \sum_{k=0}^{\infty} \frac{(a)_k (b)_k}{k!(c)_k} z^k \ , \qquad |z| < 1$$

$$I = e^{2ab} \sqrt{\frac{b}{a}} \sum_{n=0}^{\infty} A_n \sum_{k=0}^{\infty} B_k(n) \int_{u(t)/u_m}^{\infty} \frac{erfc(u_m v) dv}{v^{2n+2k+2}}, \qquad u_m = 2\sqrt{ab}, \quad u(t) = a\sqrt{t} + b/\sqrt{t}$$

where

$$A_n = \left(\frac{ab}{4} \right)^n \frac{1}{n!}, \quad n \ge 0, \qquad B_0(n) = 1, \quad B_k(n) = \frac{(n+1)_k (n+3/2)_k}{k!(2n+2)_k}, \quad k \ge 1, \quad n \ge 0$$

Now we have from the Table of Integrals, Chapter 1,

$$M(\mu, a, x) = \int_x^{\infty} \frac{erfc(av)}{v^{\mu}} dv = \frac{1}{(\mu-1)x^{\mu-1}} \left[erfc(ax) - \frac{ax}{\sqrt{\pi}} E_{\mu/2}(a^2 x^2) \right], \qquad \mu \ne 1$$

and we can write

$$I = e^{2ab} \sqrt{\frac{b}{a}} \sum_{n=0}^{\infty} A_n \sum_{k=0}^{\infty} B_k(n) M(2n+2k+2, u_m, u(t)/u_m), \qquad u_m = 2\sqrt{ab}, \qquad u(t) = a\sqrt{t} + b/\sqrt{t}.$$

Now let l = n + k or n = l - k and, in place of summing horizontally or vertically in the (k,n) lattice, we sum along left sloping diagonals with a slope of -1:

$$I_{27}(a,b,t) = e^{2ab} \sqrt{\frac{b}{a}} \sum_{l=0}^{\infty} M(2l+2, u_m, u(t)/u_m) \sum_{k=0}^{l} A_{l-k} B_k(l-k), \qquad u_m = 2\sqrt{ab}$$

where

$$M(2l+2, u_m, u(t)/u_m) \equiv \int_{u(t)/u_m}^{\infty} \frac{erfc(u_m v)}{v^{2l+2}} dv = \frac{1}{(2l+1)} \left(\frac{u_m}{u(t)} \right)^{2l+1} \left[erfc(u(t)) - \frac{u(t)}{\sqrt{\pi}} E_{(l+1)}(u^2(t)) \right]$$

Since u_m is less than u(t) for t< t_m and u(t) goes to infinity for t to zero, the ratio $u_m / u(t)$ is less than 1 and the convergence appears to be geometric.

The recurrence relation

$$B_0(n) \equiv 1, \qquad B_k(n) = B_{k-1}(n) \frac{(n+k)}{k} \frac{(n+k+1/2)}{(2n+k+1)}, \quad k \ge 1, \qquad n \ge 0$$

with n = l - k will generate a triangular lattice in the (k,n) set with diagonal values for $B_k(l-k)$ *for* $k \ge 1$.To generate these diagonals , we set $B_0(n) \equiv 1$ for k=0 (values on the n axis). To start, we apply the recurrence for k=1 and n=0 to get $B_1(0)$=0.75. Now we have $B_0(1) \equiv 1$ and

$B_1(0)$=0.75 for the first diagonal for l=1. For l=2, we set $B_0(2) \equiv 1$, generate $B_1(1)$=1.25 from $B_0(1)$=1 and $B_2(0)$=0.625 from $B_1(0)$=0.75. Similarly for l =3,4,…. The table for l=3 is

1			
1	1.75		
1	1.25	1.3125	
1	0.75	0.6250	0.54688

We also generate the A_n sequence by recurrence:

$$A_0 = 1, \qquad A_n = A_{n-1}\left(\frac{ab}{4}\right)/n\ , \quad n \ge 1.$$

<u>Computation of $\mathbf{J_{27}(a,b,t)}$</u> <u>for</u> $\mathbf{t \le t_m}$

Since $\mathrm{I}_{27}(\mathrm{a,b,t})$ and $\mathrm{J}_{27}(\mathrm{a,b,t})$ are similar, we can apply the technique outlined above to $\mathrm{J}_{27}(\mathrm{a,b,t})$. The results are:

$$J_{27}(a,b,t) \ = \frac{e^{2ab}}{4}\left(\frac{b}{a}\right)^{3/2}\sum_{l=0}^{\infty} M(2l+4,u_m,u(t)/u_m)\sum_{k=0}^{l} A_{l-k}B_k(l-k),$$

$$M(2l+4,u_m,u(t)/u_m) \equiv \int_{u(t)/u_m}^{\infty} \frac{erfc(u_m v)}{v^{2l+4}}dv = \frac{1}{(2l+3)}\left(\frac{u_m}{u(t)}\right)^{2l+3}\left[erfc(u(t)) - \frac{u(t)}{\sqrt{\pi}}E_{(l+2)}(u^2(t))\right]$$

with $u_m = 2\sqrt{ab}$, $u(t) = a\sqrt{t} + b/\sqrt{t}$ and

$$A_n = \left(\frac{ab}{4}\right)^n \frac{1}{n!}, \quad n \ge 0, \qquad B_0(n) = 1, \quad B_k(n) = \frac{(n+2)_k (n+5/2)_k}{k!(2n+4)_k}, \quad k \ge 1, \quad n \ge 0$$

and a corresponding recurrence

$$B_0(n) \equiv 1, \qquad B_k(n) = B_{k-1}(n)\cdot\frac{(n+k+1)}{k}\cdot\frac{(n+k+3/2)}{(2n+k+3)}, \quad k \ge 1, \qquad n \ge 0 \qquad .$$

<u>Numerical Experiments</u>

The formulas for $\mathrm{I}_{27}(\mathrm{a,b,t})$ and $\mathrm{J}_{27}(\mathrm{a,b,t})$ for $t \le t_m$ were compared with direct quadratures on the original integrals using DGAUS8. The results for a and b over a modest range (<5) indicate that convergence is best on $t < t_m/2$ with about 300 terms for (relative) errors $O(10^{-13})$ in the worst cases. As t exceeds $t_m/2$ there is a rapid increase in the number of terms to get errors $O(10^{-13})$. Put another way, for a fixed number of terms, the accuracy decreases rapidly when t exceeds $t_m/2$. A closer analysis (using the geometric convergence) with $u_m/u(t) \le 1/2$ shows that $t \le 0.072 t_m$ is necessary to achieve this accuracy in about 50 terms. This indicates that the expression is best for small t. This probably could have been foreseen since the derivative du/dt at $t = t_m$ is quite shallow and the ratio $u_m/u(t)$ stays close to 1 or a long distance from $t = t_m$.

The formula connecting $I_{27}(a,b,t)$ and $J_{27}(a,b,t)$ which was developed in the first section was also verified numerically using the series forms for $I_{27}(a,b,t)$ and $J_{27}(a,b,t)$ for t< t_m.

Case 2, $t>t_m$.

The plan for computing $I_{27}(a,b,t)$ for t> t_m was to compute $I_{27}(a,b,t_m)$ from Case 1 and continue the integration from t_m to t,

$$I_{27}(a,b,t) = I_{27}(a,b,t_m) + e^{2ab}\int_{t_m}^{t} \frac{e^{a^2\tau} erfc(a\sqrt{\tau}+b/\sqrt{\tau})}{\sqrt{\tau}} d\tau \quad , \quad t>t_m$$

but the numerical experiments from Case 1 show that the computation of $I_{27}(a,b,t_m)$ is not generally feasible. Nevertheless, we do the analysis on the integral for t> t_m,

$$I = e^{2ab}\int_{t_m}^{t} \frac{e^{a^2\tau} erfc(a\sqrt{\tau}+b/\sqrt{\tau})}{\sqrt{\tau}} d\tau .$$

We proceed as before with the branch for $t > t_m$,

$$\sqrt{\tau} = \frac{u+\sqrt{u^2-u_m^2}}{2a}, \qquad \tau = \frac{1}{4a^2}[u+\sqrt{u^2-u_m^2}]^2$$

and get

$$I = 2e^{2ab}\int_{t_m}^{t} \frac{e^{(u+\sqrt{u^2-u_m^2})^2/4} erfc(u)\tau}{a\tau - b} d\tau .$$

Now, keeping in mind $u_m^2 = 4ab$,

$$a\tau - b = -b - \frac{u_m^2}{4a} + \frac{u}{2a}(u+\sqrt{u^2-u_m^2}) = \sqrt{u^2-u_m^2}\,(u+\sqrt{u^2-u_m^2})/(2a)$$

and

$$I = \frac{e^{2ab}}{a}\int_{t_m}^{t} \frac{e^{(u+\sqrt{u^2-u_m^2})^2/4} erfc(u)(u+\sqrt{u^2-u_m^2})}{\sqrt{u^2-u_m^2}} du .$$

Then, with a change of variables $u = u_m v$,

$$I = \frac{u_m e^{2ab}}{a}\int_{1}^{u(t)/u_m} \frac{e^{u_m^2 v^2(1+\sqrt{1-1/v^2})^2/4} erfc(u_m v)(1+\sqrt{1-1/v^2})}{\sqrt{1-1/v^2}} du .$$

Expanding out the exponential, we get

$$I = \frac{u_m e^{2ab}}{a}\sum_{n=0}^{\infty}\left(\frac{u_m^2}{4}\right)^n \frac{1}{n!}\int_1^{\frac{u(t)}{u_m}} \frac{v^{2n} erfc(u_m v)(1+\sqrt{1-1/v^2})^{2n+1}}{\sqrt{1-1/v^2}} dv .$$

Now, we manipulate the form

$$\frac{(1+\sqrt{1-z})^{2n+1}}{\sqrt{1-z}} = \frac{1}{\sqrt{1-z}}\sum_{k=0}^{2n+1}\binom{2n+1}{k}(\sqrt{1-z})^k$$

$$= \frac{\mathbf{1}}{\sqrt{1-z}}\sum_{k=0}^{n}\binom{2n+1}{2k}(1-z)^k + \sum_{k=0}^{n}\binom{2n+1}{2k+1}(1-z)^k$$

Now with

$$(1-z)^a = F(-a,b,b,z), \qquad |z|<1$$

and the hypergeometric series for F we get

$$\frac{\mathbf{1}}{\sqrt{1-z}}\sum_{k=0}^{n}\binom{2n+1}{2k}(1-z)^k = \sum_{k=0}^{n} C_k(n)\sum_{m=0}^{\infty} A_m(k)z^m$$

$$\sum_{k=0}^{n}\binom{2n+1}{2k+1}(1-z)^k = \sum_{k=0}^{n} D_k(n)\sum_{m=0}^{\infty} B_m(k)z^m$$

where

$$C_k(n) = \binom{2n+1}{2k}, \qquad D_k(n) = \binom{2n+1}{2k+1}, \quad k = 0, n \qquad z=1/v^2,$$

$$A_m(k) = \frac{(1/2-k)_m}{m!}, \quad m \ge 0, \quad B_m(k) = \frac{(-k)_m}{m!}, \quad m \ge 1, \qquad A_0(k) \equiv 1, \; B_0(k) \equiv 1 \;\; \mathit{for}\ k=0$$

Exchanging the order of summation, we have

$$I = 2\sqrt{\frac{b}{a}}e^{2ab}\sum_{n=0}^{\infty}\frac{(ab)^n}{n!}\sum_{m=0}^{\infty}[R(m,n)+S(m,n)]\int_1^{u(t)/u_m}\frac{erfc(u_m v)}{v^{2m-2n}}dv$$

where

$$R(m,n) = \sum_{k=0}^{n} C_k(n)A_m(k), \qquad S(m,n)=\sum_{k=0}^{n} D_k(n)B_m(k).$$

We also have (Table of Integrals, Chapter 1)

$$M(\mu,a,x) = \int_x^{\infty}\frac{erfc(av)}{v^{\mu}}dv = \begin{bmatrix} \frac{1}{(\mu-1)x^{\mu-1}}[erfc(ax) - \frac{ax}{\sqrt{\pi}}E_{\mu/2}(a^2x^2)] \;, & \mu \ge 2 \text{ and } \mu<0 \\ G(ax) & \mu=1 \\ ierfc(ax)/a & \mu=0 \end{bmatrix}$$

or

$$I = 2\sqrt{\frac{b}{a}}e^{2ab}\sum_{n=0}^{\infty}\frac{(ab)^n}{n!}\sum_{m=0}^{\infty}[R(m,n)+S(m,n)][M(2m-2n,u_m,1) - M(2m-2n,u_m,u(t)/u_m)]$$

We notice that 2m-2n can be negative for n greater than m, hence the necessity to specify M for negative indices. For positive indices the exponential integral can be computed from DEXINT and DHEXINT for m-n even and odd. However we need negative indices when m-n is negative and these can be generated by backward recurrence from $E_0(x)=exp(-x)/x$ and $E_{1/2}(x) = \sqrt{\pi}\ erfc(\sqrt{x})/\sqrt{x}$ for m-n even and m-n odd. Backward recurrence in this application is numerically stable.

In order to program this formula for numerical evaluation, we split out the special cases for n=0 and m=0 where the notation may deviate from the cases for n and m positive.

n=0 Cases

$$[R(0,0)+S(0,0)][M(0,u_m,1)-M(0,u_m,\frac{u(t)}{u_m})]+\sum_{m=1}^{\infty}[R(m,0)+S(m,0)][M(2m,u_m,1)-M(2m,u_m,\frac{u(t)}{u_m})]$$

n=0, m=0

$$R(0,0)=\sum_{k=0}^{n}C_k(0)A_0(k)=\begin{pmatrix}1\\0\end{pmatrix}\cdot 1,\quad since\quad C_k(0)=\begin{pmatrix}1\\2k\end{pmatrix},\quad and\quad A_0(k)\equiv 1$$

$$S(0,0)=\sum_{k=0}^{n}D_k(0)B_0(k)=\begin{pmatrix}1\\1\end{pmatrix}\cdot 1,\quad since\quad D_k(0)=\begin{pmatrix}1\\2k+1\end{pmatrix},\quad and\quad B_0(k)\equiv 1$$

$$R(0,0)+S(0,0)=2$$

n=0, m>0

$$R(m,0)=\sum_{k=0}^{n}C_k(0)A_m(k)=\left[\begin{pmatrix}1\\2k\end{pmatrix}\cdot\frac{(1/2-k)_m}{m!}\right]_{k=0}=\frac{(1/2)_m}{m!},\quad m>0$$

$$S(m,0)=\sum_{k=0}^{n}D_k(0)B_m(k)=\left[\begin{pmatrix}1\\2k+1\end{pmatrix}\cdot\frac{(-k)_m}{m!}\right]_{k=0}=1\cdot\begin{bmatrix}1\\0\end{bmatrix}\begin{matrix}m=0\\m>0\end{matrix}$$

$$R(m,0)+S(m,0)=\frac{(1/2)_m}{m!}+\begin{bmatrix}1\\0\end{bmatrix}=\begin{bmatrix}2\\\frac{(1/2)_m}{m!}\end{bmatrix}\begin{matrix}m=0\\m>0\end{matrix}$$

n>0, m=0 Cases

$$R(0,n)=\sum_{k=0}^{n}C_k(n)A_0(k)=\sum_{k=0}^{n}C_k(n)\cdot 1$$

$$S(0,n)=\sum_{k=0}^{n}D_k(n)B_0(k)=\sum_{k=0}^{n}D_k(n)\cdot 1$$

$$R(0,n)+S(0,n)=\sum_{k=0}^{n}[C_k(n)+D_k(n)]=2^{2n+1}\quad \textit{(sum of even and odd binomial coefficients)}$$

Acceleration of Convergence for n=0 Case

The n=0 sum is given by (R(m,0)+S(m,0) gives an extra integral for m=0. n=0 in the original expansion gives the same leading term in the following)

$$S_{n=0}=\int_1^{u(t)/u_m}erfc(u_m v)dv+\int_1^{u(t)/u_m}\frac{erfc(u_m v)}{\sqrt{1-1/v^2}}dv\,.$$

We integrate by parts:

$$x=erfc(u_m v)\qquad dy=\frac{v}{\sqrt{v^2-1}}dv$$

$$dx=-\frac{2u_m}{\sqrt{\pi}}e^{-u_m^2v^2}dv\qquad y=\sqrt{v^2-1}$$

$$\int_1^{u(t)/u_m}\frac{erfc(u_m v)}{\sqrt{1-1/v^2}}dv=\left[\sqrt{v^2-1}\,erfc(u_m v)\right]_1^{u(t)/u_m}+\frac{2u_m}{\sqrt{\pi}}\int_1^{u(t)/u_m}e^{-u_m^2v^2}\sqrt{v^2-1}dv$$

and change the variable $w=v^2$ to get

$$\int_1^{u(t)/u_m} \frac{erfc(u_m v)}{\sqrt{1-1/v^2}} dv = \left[\sqrt{\frac{u^2(t)}{u_m^2}-1}\ erfc(u(t))\right] + \frac{u_m}{\sqrt{\pi}} \int_1^{[u(t)/u_m]^2} e^{-u_m^2 w} \sqrt{1-1/w}\ dw$$

Expanding out the radical $\sqrt{1-1/w}$ and adding in the extra term, we have for n=0,

$$S_{n=0} = \left[\sqrt{\frac{u^2(t)}{u_m^2}-1}\ erfc(u(t))\right] + \frac{ierfc(u_m) - ierfc(u(t))}{u_m} + \frac{u_m}{\sqrt{\pi}}\left[\frac{e^{-u_m^2} - e^{-u^2(t)}}{u_m^2}\right] +$$

$$+ \frac{u_m}{\sqrt{\pi}} \sum_{k=1}^{\infty} \frac{(\text{-}1/2)_k}{k!} \left[E_k(u_m^2) - \left[\frac{u_m^2}{u^2(t)}\right]^{k-1} E_k(u^2(t)) \right]$$

Note that,

$$\frac{(-1/2)_k}{k!} = \frac{\Gamma(k-1/2)}{\Gamma(-1/2)\Gamma(k+1)} \sim -\frac{k^{-3/2}}{2\sqrt{\pi}} \quad and \quad E_k(x) \sim \frac{e^{-x}}{x+k} \qquad for\ k \to \infty$$

and this makes the overall convergence $O(k^{-5/2})$ and the truncation error $O(k^{-3/2})$.

<u>Numerical Experiments</u>

The numerical experiments were disappointing. Barring an error, accurate answers for I on t>t_m were obtained only for very small a and b. One can surmise from the appearance of positive moments of $erfc(av)$ for n>m that high cancellation of digits is likely to occur .

<u>Recurrence Relation and Series For</u> $\mathbf{Y_n = \int_0^t \tau^{n-1/2} U(a,b,\tau) d\tau}$, n=0,1,...

We noted before that integration by parts on this integral for n=0 gave the relationship between $\mathrm{I}_{27}(a,b,t)$ and $\mathrm{J}_{27}(a,b,t)$. We repeat the process for n>0 and arrive at a recurrence relation:

$$u = U(a,b,\tau) \qquad\qquad dv = \tau^{n-1/2} d\tau$$

$$du = \left[-\frac{2}{\sqrt{\pi}} e^{-b^2/\tau} \left(\frac{a}{2\sqrt{\tau}} - \frac{b}{2\tau^{3/2}}\right) + a^2 U(a,b,\tau) \right] d\tau \qquad\qquad v = \frac{\tau^{n+1/2}}{n+1/2}$$

and

$$\mathrm{Y_n}(a,b,t) = \frac{t^{n+1/2}}{n+1/2} U(a,b,t) + \frac{a t^{n+1}}{(n+1/2)\sqrt{\pi}} E_{n+2}\left(\frac{b^2}{t}\right) - \frac{b t^n}{(n+1/2)\sqrt{\pi}} E_{n+1}\left(\frac{b^2}{t}\right) - \frac{a^2}{(n+1/2)} \mathrm{Y_{n+1}}(a,b,t)$$

for n=0,1,2,… From this relation we can increase the index by one and get $\mathrm{Y_n}(a,b,t)$ in terms of $\mathrm{Y_{n+2}}(a,b,t)$, etc. This amounts to backward recurrence on the recurrence relation and develops a series in $a\sqrt{t}$. Notice that the terms involve powers of $a\sqrt{t}$ in the numerator and a build up of factorials in the denominator. Explicitly, let

$$Y_n = R_n - \frac{a^2}{(n+1/2)} Y_{n+1}$$

where

$$\mathrm{R_n}(a,b,t) = \frac{t^n}{(n+1/2)}\left(\sqrt{t}\,U(a,b,t) + \frac{at}{\sqrt{\pi}}E_{n+2}\left(\frac{b^2}{t}\right) - \frac{b}{\sqrt{\pi}}E_{n+1}\left(\frac{b^2}{t}\right)\right).$$

Then,

$$Y_0 = R_0 - \frac{a^2}{(1/2)}Y_1 \;\text{ and }\; Y_1 = R_1 - \frac{a^2}{(3/2)}Y_2 \;\text{ giving }\; Y_0 = R_0 - \frac{a^2}{(1/2)}R_1 + \frac{a^4}{(1/2)(3/2)}Y_2 .$$

Induction gives

$$I_{27}(a,b,t) = Y_0 = \sum_{n=0}^{\infty}\frac{(-1)^n R_n a^{2n}}{(1/2)_n},$$

with

$$\frac{R_n a^{2n}}{(1/2)_n} = \frac{(a^2t)^n}{(1/2)_{n+1}}\left(\sqrt{t}\,U(a,b,t) + \frac{at}{\sqrt{\pi}}E_{n+2}\left(\frac{b^2}{t}\right) - \frac{b}{\sqrt{\pi}}E_{n+1}\left(\frac{b^2}{t}\right)\right).$$

Similarly,

$$J_{27}(a,b,t) = Y_1 = \sum_{n=1}^{\infty}\frac{(-1)^{n-1} R_n a^{2(n-1)}}{(3/2)_{n-1}}$$

$$\frac{R_n a^{2(n-1)}}{(3/2)_{n-1}} = \frac{a^{2n-2}(t)^n}{(3/2)_n}\left(\sqrt{t}\,U(a,b,t) + \frac{at}{\sqrt{\pi}}E_{n+2}\left(\frac{b^2}{t}\right) - \frac{b}{\sqrt{\pi}}E_{n+1}\left(\frac{b^2}{t}\right)\right).$$

Thus the series is best for a relatively small $a\sqrt{t}$.

Numerical Experiments on the recurrence for $Y_n = \int_0^t \tau^{n-1/2}U(a,b,\tau)d\tau$, n=0,1,...

The formulae for Y_0 and Y_1 were coded and evaluated for a and b up to 6 and t up to $t = 1.2t_m$, $t_m = b/a$, and compared with a direct quadrature . Relative errors of $O(10^{-13})$ were obtained in most cases of interest where the magnitude was greater than 0.001 in less than 120 terms; otherwise decimal place accuracy prevailed for answers of lower magnitude down to unit roundoff $O(10^{-14})$. One can predict losses of significance because for a^2t large, the terms get large and alternate in sign to produce small answers, meaning that high cancellation occurs.

Identification of Leading Sums and Standardization of Series

The formulae for $I_{27}(a,b,t)$ and $J_{27}(a,b,t)$ can be written as

$$\mathrm{I}_{27}(a,b,t) = \sqrt{t}\,U(a,b,t)\sum_{n=0}^{\infty}\frac{(-a^2t)^n}{(1/2)_{n+1}} + \sum_{n=0}^{\infty}\frac{(-a^2t)^n}{(1/2)_{n+1}}\left[\frac{at}{\sqrt{\pi}}E_{n+2}\left(\frac{b^2}{t}\right) - \frac{b}{\sqrt{\pi}}E_{n+1}\left(\frac{b^2}{t}\right)\right]$$

$$J_{27}(a,b,t) = \sqrt{t}\,U(a,b,t)\sum_{n=1}^{\infty}\frac{(-a^2)^{n-1}(t)^n}{(3/2)_n} + \sum_{n=1}^{\infty}\frac{(-a^2)^{n-1}(t)^n}{(3/2)_n}\left[\frac{at}{\sqrt{\pi}}E_{n+2}\left(\frac{b^2}{t}\right) - \frac{b}{\sqrt{\pi}}E_{n+1}\left(\frac{b^2}{t}\right)\right].$$

The leading sums

$$\sum_{n=0}^{\infty}\frac{(-a^2t)^n}{(1/2)_{n+1}} = 2\sum_{n=0}^{\infty}\frac{(-a^2t)^n}{(3/2)_n}, \qquad \sum_{n=1}^{\infty}\frac{(-a^2)^{n-1}(t)^n}{(3/2)_n} = t\sum_{n=1}^{\infty}\frac{(-a^2t)^{n-1}}{(3/2)_n} = -\frac{1}{a^2}\sum_{n=1}^{\infty}\frac{(-a^2t)^n}{(3/2)_n}$$

can be identified by writing these sums in terms of the confluent hypergeometric function

$$\phi(a,c,x) = \sum_{k=0}^{\infty}\frac{(a)_k}{(c)_k}\frac{x^k}{k!} \qquad \text{and} \qquad \sum_{n=0}^{\infty}\frac{(-a^2t)^n}{(3/2)_n} = \phi(1,3/2,-a^2t)$$

since $(1)_k = k!$. But , [A&S (7.1.21) p.298]

$$erf\,(ix) = \frac{2ix}{\sqrt{\pi}}e^{x^2}\phi(1,3/2,-x^2) \qquad or \qquad \phi(1,3/2,-x^2) = -i\frac{\sqrt{\pi}}{2x}e^{-x^2}erf\,(ix)$$

Notice that

$$erf\,(iy) = \frac{2}{\sqrt{\pi}}\int_0^{iy}e^{-t^2}dt = \frac{2i}{\sqrt{\pi}}\int_0^{y}e^{w^2}dw \qquad (\,path\ is\ t = iw\,).$$

Then,

$$\phi(1,3/2,-x^2) = \sum_{n=0}^{\infty}\frac{(-x^2)^n}{(3/2)_n} = \frac{F(x)}{x}, \quad where \quad F(x) = e^{-x^2}\int_0^{x}e^{w^2}dw$$

and F(x) is known as Dawson’s integral[A&S (7.1.16) p.298]. The leading sums become

$$\sum_{n=0}^{\infty}\frac{(-a^2t)^n}{(1/2)_{n+1}} = 2\frac{F(x)}{x}, \qquad \sum_{n=1}^{\infty}\frac{(-a^2)^{n-1}(t)^n}{(3/2)_n} = \frac{1}{a^2}[1-\frac{F(x)}{x}] \qquad x=a\sqrt{t}\,.$$

From above, we also have

$$F(x) = x\phi(1,3/2,-x^2) = -i\frac{\sqrt{\pi}}{2}e^{-x^2}erf\,(ix)$$

and Dawson’s integral can be computed using the complex error function subroutine ZERF in file ZPLXSUBS.FOR Finally,

$$\mathrm{I}_{27}(a,b,t) = \frac{2}{a}U(a,b,t)F(a\sqrt{t}) + 2\sum_{n=0}^{\infty}\frac{(-a^2t)^n}{(3/2)_n}\left[\frac{at}{\sqrt{\pi}}E_{n+2}\left(\frac{b^2}{t}\right) - \frac{b}{\sqrt{\pi}}E_{n+1}\left(\frac{b^2}{t}\right)\right]$$

$$J_{27}(a,b,t) = \frac{\sqrt{t}}{a^2}U(a,b,t)\left[1-\frac{F(a\sqrt{t})}{a\sqrt{t}}\right] - \frac{1}{a^2}\sum_{n=1}^{\infty}\frac{(-a^2t)^n}{(3/2)_n}\left[\frac{at}{\sqrt{\pi}}E_{n+2}\left(\frac{b^2}{t}\right) - \frac{b}{\sqrt{\pi}}E_{n+1}\left(\frac{b^2}{t}\right)\right].$$

Folder 28

Reciprocal Relations

$$\mathbf{\frac{1}{\sqrt{\pi}} E_{n+1/2}(x^2) = \sum_{k=1}^{n} A(n,k)\, i^{2k\text{-}1}erfc(x)\ , \qquad n \geq 1,}$$

$$\mathbf{i^{2n\text{-}1}erfc(x) = \sum_{k=1}^{n} B(n,k)[E_{k+1/2}(x^2)/\sqrt{\pi}\,], \qquad n \geq 1\,,}$$

$$\mathbf{\frac{2x}{\sqrt{\pi}} E_{n+1/2}(x^2) = \sum_{k=0}^{n} A(n+1,k+1)\, i^{2k}erfc(x)\ , \qquad n \geq 0,}$$

$$\mathbf{i^{2n}erfc(x) = 2x\sum_{k=0}^{n} B(n+1,k+1)[E_{k+1/2}(x^2)/\sqrt{\pi}\,], \qquad n \geq 0\,,}$$

with application to $\mathbf{I_3^c(a,b,c,T)}$ of Folder 7

Donald E. Amos, June 2005

Summary

The formula

$$\frac{1}{\sqrt{\pi}} E_{n+1/2}(x^2) = \sum_{k=1}^{n} A(n,k)\ i^{2k\text{-}1}erfc(x)\ , \qquad\qquad n \geq 1$$

generalizes the known relations

$$\frac{1}{\sqrt{\pi}} E_{1/2}(x^2) = \frac{erfc(x)}{x}\ , \quad \frac{1}{\sqrt{\pi}} E_{3/2}(x^2) = 2\,ierfc(x)\,,$$

$$\frac{1}{\sqrt{\pi}} E_{5/2}(x^2) = 4x\,i^2 erfc(x) + 4\,i^3 erfc(x) = 2\,ierfc(x) - 8\,i^3 erfc(x)$$

where $E_{5/2}(x^2)/\sqrt{\pi}$ is derived in Folder 24. The inverse relation

$$i^{2n\text{-}1}erfc(x) = \sum_{k=1}^{n} B(n,k)\left[\,E_{k+1/2}(x^2)/\sqrt{\pi}\,\right], \qquad n \geq 1$$

is obtained by solving the lower triangular system for the odd order iterated co-error functions. Explicitly,

$$A(n,k) = 2\,(\text{-}4)^{k\text{-}1}(k\text{-}1)!\,C_{k\text{-}1}^{n\text{-}1} = \frac{(-1)^{k-1}2^{2k\text{-}1}(n-1)!}{(n\text{-}k)!}\ , \qquad 1 \leq\ k\ \leq\ n, \qquad n \geq 1$$

$$B(n,k) = \frac{(\text{-}1)^{k\text{-}1}C_{k\text{-}1}^{n\text{-}1}}{2^{2n\text{-}1}(n\text{-}1)!} = \frac{(\text{-}1)^{k\text{-}1}}{2^{2n\text{-}1}(k-1)!(n-k)!}\ , \qquad 1 \leq\ k\ \leq\ n, \qquad n \geq 1$$

The expansions for co-error functions of even orders are obtained by differentiation of the odd order expansions.

The application to the numerical evaluation of $I_3^c(a,b,c,T)$ comes about through the computation of $R_k(a,d,T)$ in the formulae

$$I_3^c(a,b,c,T) \equiv \int_T^{\infty} e^{\text{-}c^2w^2} erfc(aw)erfc(bw)dw = erfc(aT)I_5(c,b,T) - S(a,b,c,T)$$

$$S(a,b,c,T) = \frac{2a}{\sqrt{\pi}}\int_T^{\infty}\ e^{-a^2w^2}I_5(c,b,w)dw = \frac{a}{\pi d}\sum_{k=0}^{\infty}\ C_k\left(\frac{c^2}{d^2}\right)^k R_k\,,$$

where

$$R_k = \int_T^{\infty} e^{-a^2w^2} E_{k+3/2}(d^2w^2)dw, \qquad k \ge 0,$$
$$d^2 = b^2 + c^2 \qquad\qquad C_k = (1/2)_k/k!,$$

a , $c \le b$ (Case I in Folder 7a) and I_5 is the I integral of Folder 5. The full computation for $I_3^c(a,b,c,T)$ is described in Folder 7a in Cases I and II with refinements for $I_3(a,b,c,T)$ in Folder 7f. Cases not covered in Case I are addressed in Case II where a formula is presented which reduces the computation to that of Case I.

Derivation of the Reciprocal Relations

Notice that

$$\frac{1}{\sqrt{\pi}} E_{3/2}(x^2) = 2\,ierfc(x)$$

gives

$$n=1, \qquad\qquad A(1,1)=2$$

and

$$\frac{1}{\sqrt{\pi}} E_{5/2}(x^2) = 4x\,i^2erfc(x) + 4\,i^3erfc(x) = 2\,ierfc(x) - 8\,i^3erfc(x)$$

from Folder 24 gives

$$n=2, \qquad\qquad A(2,1)=2, \quad A(2,2)=-8.$$

We proceed by induction. Multiply both sides of

$$\frac{1}{\sqrt{\pi}} E_{n+1/2}(x^2) = \sum_{k=1}^{n} A(n,k)\; i^{2k\text{-}1}erfc(x)\;, \qquad\qquad n \ge 1$$

by 2x. Then,

$$\frac{2x}{\sqrt{\pi}} E_{n+1/2}(x^2) = \sum_{k=1}^{n} A(n,k)\; 2x\,i^{2k\text{-}1}erfc(x)\;, \qquad\qquad n \ge 1$$

and the recurrence

$$2n\,i^n erfc(x) + 2x\,i^{n-1}erfc(x) = i^{n-2}erfc(x)$$

with n=2k

$$2x\,i^{2k-1}erfc(x) = i^{2k-2}erfc(x) - 4k\,i^{2k}\,erfc(x)$$

gives

$$\frac{2x}{\sqrt{\pi}} E_{n+1/2}(x^2) = \sum_{k=1}^{n} A(n,k)\,i^{2k\text{-}2}erfc(x) - \sum_{k=1}^{n} 4kA(n,k)\,i^{2k}\,erfc(x)$$

or

$$\frac{2x}{\sqrt{\pi}} E_{n+1/2}(x^2) = A(n,1)\,erfc(x) + \sum_{k=2}^{n}\left[A(n,k) - 4(k-1)A(n,k-1)\right] i^{2k\text{-}2}erfc(x) - 4nA(n,n)\; i^{2n}\,erfc(x)\,.$$

Now we integrate on $[x,\infty)$ to obtain

$$\frac{1}{\sqrt{\pi}} E_{n+3/2}(x^2) = A(n,1)\,ierfc(x) + \sum_{k=2}^{n}\left[A(n,k) - 4(k-1)A(n,k-1)\right] i^{2k\text{-}1}erfc(x) - 4nA(n,n)\; i^{2n+1}erfc(x)\,.$$

Since both sides represent $E_{n+3/2}(x^2)/\sqrt{\pi}$, we equate like terms to get

$$A(n+1,1) = A(n,1) = 2 \qquad k=1$$
$$A(n+1,k) = A(n,k) - 4(k-1)A(n,k-1) \qquad 2 \le k \le n\ , \qquad n \ge 2$$
$$A(n+1,n+1) = -4nA(n,n) \qquad k = n+1$$

or, with a shift of index, the complete recurrence is

$$A(1,1) = 2 \qquad n=1,$$
$$A(2,1) = 2, \qquad A(2,2) = -8. \qquad n=2,$$
$$A(n,1) = A(n-1,1) = 2 \qquad k=1$$
$$A(n,k) = A(n-1,k) - 4(k-1)A(n-1,k-1) \qquad 2 \le k \le n-1\ , \qquad n \ge 3.$$
$$A(n,n) = -4(n-1)A(n-1,n-1) \qquad k=n$$

It is easy to deduce the special cases

$$A(n,1) = 2 \qquad \mathrm{n} \ge 1$$
$$A(n,n) = 2(-4)^{n-1}(n-1)! \qquad \mathrm{n} \ge 1\ .$$
$$A(n,2) = -8(n-1), \qquad \mathrm{n} \ge 2$$

TABLE 28 shows values of A(n,k), $1 \le k \le n$, for n as high as 20. Notice that all coefficients in the triangular matrix for A are integers, but for n>12, some integers are truncated when generated in double precision arithmetic. If we let

$$A(n,k) = 2(-4)^{k-1}(k-1)!\, D(n,k)$$

then D(n,k) satisfies the relation for binomial coefficients,

$$D(n,k) = D(n-1,k) + D(n-1,k-1)$$

and with A(1,1)=2, we have

$$A(n,k) = 2(-4)^{k-1}(k-1)!\, C_{k-1}^{n-1} = \frac{2(-4)^{k-1}(n-1)!}{(n-k)!} \ , \qquad 1 \le k \le n \quad , \quad n \ge 1\ .$$

The numerical computations comparing this formula with results from subroutines DINERFC and DEXINT give errors $O(10^{-14})$ up to n=12; but the results degrade, starting with the worst error at x=0 and diminishing for higher values of x with full precision restored for x values not too far from 1. At n=20 we have errors $O(10^{-9})$ for x near zero.

Because the matrix A is triangular, we can also solve for $i^n erfc(x)$, n odd, recursively in terms of $E_{k+1/2}(x^2)/\sqrt{\pi}$, $1 \le k \le n$, by solving for the diagonal term,

$$n=1 \qquad ierfc(x) = \frac{1}{2\sqrt{\pi}} E_{3/2}(x^2)$$
$$n=2 \qquad \mathrm{i}^3 erfc(x) = \frac{1}{8\sqrt{\pi}}\left[E_{3/2}(x^2) - E_{5/2}(x^2)\right]$$
$$n=3 \qquad \mathrm{i}^5 erfc(x) = \frac{1}{64\sqrt{\pi}}\left[E_{3/2}(x^2) - 2E_{5/2}(x^2) + E_{7/2}(x^2)\right]$$
$$\vdots$$

$$i^{2n\text{-}1}erfc(x) = \frac{1}{2^{2n-1}(n-1)!}\sum_{k=1}^{n}(-1)^{k-1}C_{k-1}^{n-1}\left[E_{k+1/2}(x^2)/\sqrt{\pi}\right], \qquad n \ge 1,$$

where C_{k-1}^{n-1} are binomial coefficients. If we define

$$B(n,k) = \frac{(-1)^{k-1}C_{k-1}^{n-1}}{2^{2n-1}(n-1)!} = \frac{(-1)^{k-1}}{2^{2n-1}(k-1)!(n-k)!}, \qquad 1 \le k \le n$$

we can write the reciprocal relations in the form of systems of equations

$$\hat{e} = A\hat{i} \;\;,\;\; \hat{i} = B\hat{e}$$

Where $\hat{i}$ and $\hat{e}$ are vectors with components of iterated co-error functions and exponential integrals, and A and B are the lower triangular matrices formed from A(n,k) and B(n,k) of a size of one's choosing, say N. Then we should have

AB=I.

Specifically, we compute the inner product of the nth row of A with the kth column of B

$$\sum_{j=1}^{k-1} A(n,j)\left[B(j,k)\right] + \sum_{j=k}^{n} A(n,j)B(j,k) + \sum_{j=n+1}^{N} [A(n,j)]B(j,k), \qquad k < n$$

$$\sum_{j=1}^{n-1} A(n,j)\left[B(j,n)\right] + A(n,n)B(n,n) + \sum_{j=n+1}^{N} [A(n,j)]B(j,n), \qquad k = n$$

$$\sum_{j=1}^{k-1} A(n,j)\left[B(j,k)\right] + \sum_{j=k}^{N} \left[A(n,j)\right]B(j,k), \qquad k > n$$

taking into account [A(r,s)]=0 and [B(r,s)]=0 for s>r and using the closed form expressions for A(r,s) and B(r,s) for $s \le r$. Then,

$$0 + \frac{(n\text{-}1)!}{(k\text{-}1)!}\sum_{j=k}^{n}\frac{(-1)^{j+k}}{(n-j)!(j-k)!} + 0 = \frac{(n\text{-}1)!}{(k\text{-}1)!(n-k)!}\sum_{m=0}^{n-k}(-1)^m C_m^{n-k} = 0, \qquad k < n$$

$$0 + 2(-4)^{n-1}(n-1)!\frac{(-1)^{n-1}}{2^{2n-1}(n-1)!0!} + 0 = 1, \qquad k = n$$

$$0 + 0 = 0, \qquad k > n.$$

This shows that the extrapolation of the first few values of B(n,k) to the general case is correct.

Numerical experiments confirm this reciprocal relation, though substantial losses of significance occur for n>5 with relative errors $O(10^{-5})$ at n=10 for larger values of x (x exceeding 5).

If we replace n by n+1 and differentiate with respect to x, we get the expansions for even order co-error functions,

$$\frac{2x}{\sqrt{\pi}}E_{n+1/2}(x^2) = \sum_{k=0}^{n} A(n+1,k+1)\, i^{2k}erfc(x) \;, \qquad n \ge 0$$

$$i^{2n}erfc(x) = \frac{x}{2^{2n}n!\sqrt{\pi}}\sum_{k=0}^{n}(\text{-}1)^k C_k^n E_{k+1/2}(x^2), \qquad n \ge 1.$$

The results from numerical experiments on these formulae do not differ significantly from the results of the odd order expansions. We chose to write the inverse expansion in the form

$$i^{2n}erfc(x) = \frac{1}{2^{2n}n!\sqrt{\pi}}\left[\sqrt{\pi}\; erfc(x) + x\sum_{k=1}^{n}(\text{-}1)^k C_k^n E_{k+1/2}(x^2)\right], \qquad n \ge 1$$

or

$$i^{2n} erfc(x) = 2\left[B(n+1,1)\ erfc(x) + x\sum_{k=1}^{n} B(n+1,k+1)\left[E_{k+1/2}(x^2)/\sqrt{\pi}\right]\right], \qquad n \ge 1$$

to remove the x in the denominator of the k=0 term.

If we take x=0 in the expansions for $E_{n+1/2}(x^2)/\sqrt{\pi}$, and use

$$E_\nu(0) = \frac{1}{\nu-1}, \qquad \nu > 1, \qquad and \qquad i^n erfc(0) = \frac{1}{2^n \Gamma(n/2+1)}, \qquad n \ge 0$$

we get check computations

$$\frac{1}{(n-1/2)\sqrt{\pi}} = \sum_{k=1}^{n} \frac{A(n,k)}{2^{2k-1}\Gamma(k+1/2)}, \qquad n \ge 1$$

$$0 = \sum_{k=0}^{n} \frac{A(n+1,k+1)}{2^{2k}\Gamma(k+1)}, \qquad n \ge 1$$

Application of $\mathbf{E_{n+1/2}(x^2)}$ to $\mathbf{I_3^c(a,b,c,T)}$ of Folder 7a

In Folder 7a, the results for $I_3(a,b,c,T)$ and $I_3^c(a,b,c,T)$ depend upon the evaluation of the quantity

$$R_k = \int_T^\infty e^{-a^2w^2} E_{k+3/2}(d^2w^2)dw, \qquad k \ge 0, \qquad d^2 = b^2+c^2$$

and summing the series

$$S(a,b,c,T) = \frac{2a}{\sqrt{\pi}}\int_T^\infty e^{-a^2w^2} I_5(c,b,w)dw = \frac{a}{\pi d}\sum_{k=0}^{\infty} C_k \left(\frac{c^2}{d^2}\right)^k R_k(a,d,T), \qquad C_k = (1/2)_k/k!$$

where for Case I, a , $c \le b$. All other cases are reduced to Case I. Then, using the formula for exponential integrals of half odd orders,

$$R_k(a,d,T) = \sqrt{\pi}\sum_{m=1}^{k+1} A(k+1,m)\, y_{2m-1}(a,d,T),$$

our problem is converted to the evaluation of

$$y_n(a,d,T) = \int_T^\infty e^{-a^2w^2} i^n erfc(dw)\mathrm{d}w$$

for odd indices. We derive a recurrence relation for $y_n(a,d,T)$ using the recurrence for the iterated co-error functions:

$$2n\, i^n erfc(x) + 2x\, i^{n-1} erfc(x) = i^{n-2} erfc(x), \qquad n \ge 1.$$

Then,

$$2ny_n + 2d\int_T^\infty we^{-a^2w^2} i^{n-1} erfc(dw)\mathrm{d}w = y_{n-2}$$

and integration by parts gives

$$v_{n-1} \equiv \int_T^\infty we^{-a^2w^2} i^{n-1} erfc(dw)\mathrm{d}w = \frac{e^{-a^2T^2}}{2a^2} i^{n-1} erfc(dT) - \frac{d}{2a^2} y_{n-2}.$$

The final recurrence is

$$y_{n-2} = \frac{a^2}{a^2+d^2}\left[2ny_n + \frac{d}{a^2} e^{-a^2T^2} i^{n-1} erfc(dT)\right].$$

For n=1, we can compute y_{-1} directly since $i^{-1}erfc(x)=\dfrac{2e^{-x^2}}{\sqrt{\pi}}$ and

$$y_{-1}(a,d,T)=\frac{2}{\sqrt{\pi}}\int_T^\infty e^{-(a^2+d^2)w^2}\mathrm{d}w=\frac{1}{\sqrt{a^2+d^2}}erfc(T\sqrt{a^2+d^2}).$$

Then with n=1 in the recurrence, we have

$$y_1=\frac{1}{2a^2}\left[\sqrt{a^2+d^2}\,erfc(T\sqrt{a^2+d^2})-de^{-a^2T^2}erfc(dT)\right].$$

In order to avoid losses of significance when a is small, we write y_1 in the form

$$\begin{aligned}y_1&=\frac{1}{2a^2}\left\{(\sqrt{a^2+d^2}-d)\,erfc(T\sqrt{a^2+d^2})+d\left[erfc(T\sqrt{a^2+d^2})-e^{-a^2T^2}erfc(dT)\right]\right\}\\&=\frac{erfc(T\sqrt{a^2+d^2})}{2(d+\sqrt{a^2+d^2})}+\frac{dTe^{-a^2T^2}}{2(d+\sqrt{a^2+d^2})}\sum_{k=1}^{\infty}(-1)^k\,i^k erfc(dT)(dT\phi)^{k-1}\ ,\quad dT\phi=\frac{a^2T}{(d+\sqrt{a^2+d^2})}\end{aligned}$$

by using the expansions of Folder 9. Here, aT<2.4 or $dT\phi<1$ gives rapid convergence.

Now with n=3 and y_1 computed above, we can recur forward and generate the odd indices for $y_{2m-1},\ \ m\geq 2$. Subroutine DINERFC computes sequences $i^n erfc(*),\ \ n=0,\ 1,\ 2,\ldots$ This process, however, degenerates by losing significant digits as n increases. A better result, which yields full double precision accuracy, is obtained by recurring backward from some large *odd* index N with $y_N=0$. This effectively sums a series with all positive terms in powers of $a^2/(a^2+d^2)<1$. The procedure is to select an index $N+l$, where y_l is the last y member needed for the application and with $y_{N+l}^N=0$ recur backward to index l for y_l^N; increase $N+l$ to $2N+l$ and recur backward again and compare y_l^N with y_l^{2N}, hoping that the relative error test

$$\left|y_l^N-y_l^{2N}\right|\leq\ 0.5x10^{-14}\left|y_l^{2N}\right|.$$

is satisfied (N=50 is a good choice). If this test is not satisfied, repeat the computation with $3N+l$ and compare y_l^{2N} with y_l^{3N}, etc. The super script on y denotes the fact that members of the numerical sequence depend on where you start the backward recurrence. Since $a\ ,c\leq b$, the estimates

$$\frac{a^2}{a^2+d^2}=\frac{a^2}{a^2+b^2+c^2}\leq\frac{a^2}{a^2+b^2}\leq\frac{1}{2}\ ,\qquad\left[\left(\frac{1}{2}\right)^{50}\sim 10^{-15}\right]$$

show that N=50 is a reasonable choice. We also notice that because of the relations $a\ ,c\leq b$,

$$\frac{c^2}{d^2}=\frac{c^2}{b^2+c^2}\leq\frac{1}{2}\quad\text{and}\quad C_k=\frac{\Gamma(k+1/2)}{\Gamma(k+1)\Gamma(1/2)}\sim\frac{k^{-1/2}}{\sqrt{\pi}}\quad\text{for } k\to\infty,$$

the coefficient $C_k(c^2/d^2)^k$ in the expression for S decreases rapidly.

In the previous section we noted that the sum $\sum_{m=1}^{k+1}A(k+1,m)\,i^{2m-1}erfc(dT)$ would lose significant digits for k>12. Since $y_{2m-1}(a,d,T)$ is only slightly smaller than $i^{2m-1}erfc(dT)$ one can expect similar losses in

$R_k(a,d,T) = \sqrt{\pi}\sum_{m=1}^{k+1} A(k+1,m)\, y_{2m-1}a,d,T)$. This does not necessarily doom the computation because the coefficient $C_k(c^2/d^2)^k$ in the series for S(a,b,c,T) decreases rapidly and the product will be of lower order magnitude for larger values of k. Thus when a term for a larger k value is added to the sum accumulated from the previous k values, at least some of the error in the term will be shifted off of the addition register because the smaller term is scaled to the largest exponent before addition. This scaling is accomplished by shifting all mantissa bits to the right with the left most bits being replaced by zero.

The numerical experiments support this analysis with over all errors $O(10^{-13})$ when compared with a quadrature on

$$S(a,b,c,T) = \frac{2a}{\sqrt{\pi}}\int_T^\infty e^{-a^2w^2} I_5(c,b,w)dw$$

where $I_5(c,b,w)$ is the I integral of Folder 5 which is implemented in subroutine INTEGI5.

TABLE 28

Coefficients A(n,k), $1 \le k \le n$, $1 \le n \le 20$

n = 1		
0.2000000000000000D+01		
n = 2		
0.2000000000000000D+01	-0.8000000000000000D+01	
n = 3		
0.2000000000000000D+01	-0.1600000000000000D+02	0.6400000000000000D+02
n = 4		
0.2000000000000000D+01	-0.2400000000000000D+02	0.1920000000000000D+03
-0.7680000000000000D+03		
n = 5		
0.2000000000000000D+01	-0.3200000000000000D+02	0.3840000000000000D+03
-0.3072000000000000D+04	0.1228800000000000D+05	
n = 6		
0.2000000000000000D+01	-0.4000000000000000D+02	0.6400000000000000D+03
-0.7680000000000000D+04	0.6144000000000000D+05	-0.2457600000000000D+06
n = 7		
0.2000000000000000D+01	-0.4800000000000000D+02	0.9600000000000000D+03
-0.1536000000000000D+05	0.1843200000000000D+06	-0.1474560000000000D+07
0.5898240000000000D+07		
n = 8		
0.2000000000000000D+01	-0.5600000000000000D+02	0.1344000000000000D+04
-0.2688000000000000D+05	0.4300800000000000D+06	-0.5160960000000000D+07
0.4128768000000000D+08	-0.1651507200000000D+09	
n = 9		
0.2000000000000000D+01	-0.6400000000000000D+02	0.1792000000000000D+04
-0.4300800000000000D+05	0.8601600000000000D+06	-0.1376256000000000D+08
0.1651507200000000D+09	-0.1321205760000000D+10	0.5284823040000000D+10
n = 10		
0.2000000000000000D+01	-0.7200000000000000D+02	0.2304000000000000D+04
-0.6451200000000000D+05	0.1548288000000000D+07	-0.3096576000000000D+08
0.4954521600000000D+09	-0.5945425920000000D+10	0.4756340736000000D+11
-0.1902536294400000D+12		
n = 11		
0.2000000000000000D+01	-0.8000000000000000D+02	0.2880000000000000D+04
-0.9216000000000000D+05	0.2580480000000000D+07	-0.6193152000000000D+08
0.1238630400000000D+10	-0.1981808640000000D+11	0.2378170368000000D+12
-0.1902536294400000D+13	0.7610145177600000D+13	

n = 12
```
 0.2000000000000000D+01 -0.8800000000000000D+02  0.3520000000000000D+04
-0.1267200000000000D+06  0.4055040000000000D+07 -0.1135411200000000D+09
 0.2724986880000000D+10 -0.5449973760000000D+11  0.8719958016000000D+12
-0.1046394961920000D+14  0.8371159695360000D+14 -0.3348463878144000D+15
```
n = 13
```
 0.2000000000000000D+01 -0.9600000000000000D+02  0.4224000000000000D+04
-0.1689600000000000D+06  0.6082560000000000D+07 -0.1946419200000000D+09
 0.5449973760000000D+10 -0.1307993702400000D+12  0.2615987404800000D+13
-0.4185579847680000D+14  0.5022695817216000D+15 -0.4018156653772800D+16
 0.1607262661509120D+17
```
n = 14
```
 0.2000000000000000D+01 -0.1040000000000000D+03  0.4992000000000000D+04
-0.2196480000000000D+06  0.8785920000000000D+07 -0.3162931200000000D+09
 0.1012137984000000D+11 -0.2833986355200000D+12  0.6801567252480000D+13
-0.1360313450496000D+15  0.2176501520793600D+16 -0.2611801824952320D+17
 0.2089441459961856D+18 -0.8357765839847424D+18
```
n = 15
```
 0.2000000000000000D+01 -0.1120000000000000D+03  0.5824000000000000D+04
-0.2795520000000000D+06  0.1230028800000000D+08 -0.4920115200000000D+09
 0.1771241472000000D+11 -0.5667972710400000D+12  0.1587032358912000D+14
-0.3808877661388800D+15  0.7617755322777600D+16 -0.1218840851644416D+18
 0.1462609021973299D+19 -0.1170087217578639D+20  0.4680348870314557D+20
```
n = 16
```
 0.2000000000000000D+01 -0.1200000000000000D+03  0.6720000000000000D+04
-0.3494400000000000D+06  0.1677312000000000D+08 -0.7380172800000000D+09
 0.2952069120000000D+11 -0.1062744883200000D+13  0.3400783626240000D+14
-0.9522194153472000D+15  0.2285326596833280D+17 -0.4570653193666560D+18
 0.7313045109866496D+19 -0.8775654131839795D+20  0.7020523305471836D+21
-0.2808209322188734D+22
```
n = 17
```
 0.2000000000000000D+01 -0.1280000000000000D+03  0.7680000000000000D+04
-0.4300800000000000D+06  0.2236416000000000D+08 -0.1073479680000000D+10
 0.4723310592000000D+11 -0.1889324236800000D+13  0.6801567252480000D+14
-0.2176501520793600D+16  0.6094204258222080D+17 -0.1462609021973299D+19
 0.2925218043946598D+20 -0.4680348870314557D+21  0.5616418644377469D+22
-0.4493134915501975D+23  0.1797253966200790D+24
```
n = 18
```
 0.2000000000000000D+01 -0.1360000000000000D+03  0.8704000000000000D+04
-0.5222400000000000D+06  0.2924544000000000D+08 -0.1520762880000000D+10
 0.7299661824000000D+11 -0.3211851202560000D+13  0.1284740481024000D+15
-0.4625065731686400D+16  0.1480021034139648D+18 -0.4144058895591014D+19
 0.9945741349418435D+20 -0.1989148269883687D+22  0.3182637231813899D+23
-0.3819164678176679D+24  0.3055331742541343D+25 -0.1222132697016537D+26
```
n = 19
```
 0.2000000000000000D+01 -0.1440000000000000D+03  0.9792000000000000D+04
-0.6266880000000000D+06  0.3760128000000000D+08 -0.2105671680000000D+10
 0.1094949273600000D+12 -0.5255756513280000D+13  0.2312532865843200D+15
-0.9250131463372800D+16  0.3330047326814208D+18 -0.1065615144580547D+20
 0.2983722404825530D+21 -0.7160933771581273D+22  0.1432186754316255D+24
-0.2291498806906007D+25  0.2749798568287209D+26 -0.2199838854629767D+27
 0.8799355418519068D+27
```
n = 20
```
 0.2000000000000000D+01 -0.1520000000000000D+03  0.1094400000000000D+05
-0.7441920000000000D+06  0.4762828800000000D+08 -0.2857697280000000D+10
 0.1600310476800000D+12 -0.8321614479360000D+13  0.3994374950092800D+15
-0.1757524978040832D+17  0.7030099912163328D+18 -0.2530835968378798D+20
 0.8098675098812154D+21 -0.2267629027667403D+23  0.5442309666401767D+24
-0.1088461933280353D+26  0.1741539093248566D+27 -0.2089846911898279D+28
 0.1671877529518623D+29 -0.6687510118074492D+29
```

Folder 29

Evaluation of

$$\mathbf{Y_n(a,b,T) = \int_T^\infty e^{-a^2w^2}\, i^n erfc(bw)dw, \quad a > 0, \quad b \ge 0, \quad n \ge -1}$$

With Applications to

$\mathbf{I_2^c(a,b,c,T)}$ **of Folder 9,** $\mathbf{I_5(a,b,T)}$ **of Folder 5 and** $\mathbf{I_3^c(a,b,c,T)}$ **of Folder 7**

Donald E. Amos, July 2006

Summary

Notice that this is a generalization of the I integral of Folder 5, $Y_0(a,b,T) = I_5(a,b,T)$. This integral was also encountered in Folder 28 and evaluated numerically by backward recurrence on

$$Y_{n-2} = \frac{a^2}{a^2+b^2}\left[2nY_n + \frac{b}{a^2}e^{-a^2T^2}i^{n-1}erfc(bT)\right].$$

The results of this folder give an explicit formula for Y_n, n ≥ -1.
For Case I, $a \le b$,

$$Y_n(a,b,T) = \frac{be^{-a^2T^2}}{a^2+b^2}\sum_{k=1}^{\infty}\left(\frac{4a^2}{a^2+b^2}\right)^{k-1}\frac{\Gamma(n/2+k)}{\Gamma(n/2+1)}i^{n+2k-1}erfc(bT), \quad n \ge 0$$

and for case II, a>b,

$$Y_n(a,b,T) = \frac{\sqrt{\pi}}{2a}\sum_{k=0}^{n}(-1)^k\left(\frac{b}{a}\right)^k i^k erfc(aT)\, i^{n-k}erfc(bT) + (-1)^{n+1}\left(\frac{b}{a}\right)^{n+1}Y_n(b,a,T)$$

with the special case

$$Y_{-1}(a,b,T) = \frac{erfc(T\sqrt{a^2+b^2})}{\sqrt{a^2+b^2}}.$$

Notice, as in Folder 5, we have Cases I and II where $Y_n(b,a,T)$ in Case II is computed by the formula in Case I since the first parameter is smaller than the second parameter.

We also apply Y_n for n=0 to derive a new representation for $I_5(a,b,T)$:
Case I, $a \le b$,

$$I_5(a,b,T) = Y_0(a,b,T) = \frac{be^{-a^2T^2}}{a^2+b^2}\sum_{k=1}^{\infty}\left(\frac{4a^2}{a^2+b^2}\right)^{k-1}[(k-1)!\, i^{2k-1}erfc(bT)],$$

Case II, a>b

$$I_5(a,b,T) = \frac{\sqrt{\pi}}{2a}erfc(aT)erfc(bT) - \frac{b}{a}I_5(b,a,T)$$

Fifty terms of these series gets errors $O(10^{-15})$. Subroutine DINERFC is designed to compute significant digit sequences $i^n erfc(*)$, n=0,1,….Subroutine INTEGI29 implements Cases I and II and produces significant digit sequences for $Y_n(a,b,T)$, n=0,2,4,… or n=1,3,5,…over wide ranges of parameters.

The application to $I_3^c(a,b,c,T)$ of Folder 7 follows from the representation

$$I_3^c(a,b,c,T) \equiv \int_T^\infty e^{-c^2w^2}\ erfc(aw)\,erfc(bw)dw = \mathrm{erfc}(aT)I_5(c,b,T) - S(a,b,c,T)$$

$$S(a,b,c,T) = \frac{2a}{\sqrt{\pi}}\int_T^\infty\ e^{-a^2w^2}I_5(c,b,w)dw$$

with the restriction s $a,\ c \le b$ designated as Case I. All other cases were considered in Case II which reduced the computation to Case I. Since $Y_0(a,b,T) = I_5(a,b,T)$, we can integrate the series to obtain

$$S(a,b,c,T) = \frac{2ab}{\sqrt{\pi}\sqrt{b^2+c^2}}\sum_{k=1}^\infty\ (4\gamma^2)^{k-1}[(k-1)!Y_{2k-1}(\sqrt{a^2+c^2},b,T)]\ ,\quad \gamma^2 = \frac{c^2}{b^2+c^2} \le \frac{1}{2}\ ,\qquad a\ ,\ c \le b\,.$$

<u>Derivations</u>

<u>Case I,</u> $\mathbf{a \le b}$

The explicit formula for $Y_n(a,b,T)$ for $a \le b$ is derived by solving the recurrence relation as a difference equation. Because we can solve the homogeneous equation, a variation of parameters solution for the complete solution is possible. Let

$$Y_n = \frac{b}{a^2}e^{-a^2T^2}h_n v_n$$

and substitute into the difference equation

$$Y_{n-2} = \frac{a^2}{a^2+b^2}\left[2nY_n + \frac{b}{a^2}e^{-a^2T^2}i^{n-1}erfc(bT)\right]$$

where h_n is a solution of the homogeneous equation($(-1)^n h_n$ is the other solution),

$$h_{n-2} - 2nc^2h_n = 0\,,\quad h_n = \frac{1}{c^n 2^n\Gamma(1+n/2)}\qquad \text{and}\qquad c^2 = \frac{a^2}{a^2+b^2}\,.$$

The result is

$$h_{n-2}v_{n-2} = c^2[2nh_nv_n + i^{n-1}erfc(bT)]$$

and with h_{n-2} from the homogeneous form, we get

$$v_{n-2} - v_n = \frac{i^{n-1}erfc(bT)}{2nh_n}\quad \text{or ,.with a change of index,}\ v_{m-2} - v_m = \frac{i^{m-1}erfc(bT)}{2mh_m}$$

Now we let m=n+2, m=n+4,…,m=n+2k,…,m=n+2N and sum on k

$$v_n - v_{n+2N} = \sum_{k=1}^N\frac{i^{n+2k-1}erfc(bT)}{2(n+2k)h_{n+2k}}\,.$$

Our next goal is to show that $v_{n+2N} \to 0$ as . $N \to \infty$. Expressing v_n in terms of Y_n we get

$$v_n = \frac{a^2e^{a^2T^2}}{bh_n}\int_{\mathrm{T}}^\infty \mathrm{e}^{-\mathrm{a}^2\mathrm{w}^2}\,\mathrm{i}^\mathrm{n}\,\mathrm{erfc(bw)dw}$$

and we estimate the integral,

$$|v_n| \le \frac{a^2}{b^2}i^{n+1}erfc(bT)\ c^n 2^n\Gamma(\frac{n}{2}+1) \le \frac{a^2}{b^2}\frac{c^n2^n\Gamma(\frac{n}{2}+1)}{2^{n+1}\Gamma(\frac{n}{2}+\frac{3}{2})} = \frac{a^2c^n}{2b^2}\frac{\Gamma(\frac{n}{2}+1)}{\Gamma(\frac{n}{2}+\frac{3}{2})} \sim \frac{a^2}{2b^2}\frac{c^n}{\sqrt{n/2}},\quad n\to\infty$$

using $i^n erfc(bT) \le i^n erfc(0) = 1/[2^n\Gamma(n/2+1)]$. Since c<1, we have the result $v_n \to 0$ as $n \to \infty$. Then we can write the sum as a series,

$$Y_n(a,b,T) = \frac{be^{-a^2T^2}}{a^2}\frac{1}{2^n c^n \Gamma(n/2+1)}\sum_{k=1}^{\infty}\frac{i^{n+2k-1}erfc(bT)}{2(n+2k)h_{n+2k}} = \frac{be^{-a^2T^2}}{a^2}\sum_{k=1}^{\infty}(2c)^{2k}\frac{[\Gamma(n/2+k+1)i^{n+2k-1}erfc(bT)]}{2(n+2k)\Gamma(n/2+1)}$$

or

$$Y_n(a,b,T) = \frac{be^{-a^2T^2}}{a^2+b^2}\sum_{k=1}^{\infty}\left(\frac{4a^2}{a^2+b^2}\right)^{k-1}\frac{\Gamma(n/2+k)}{\Gamma(n/2+1)}i^{n+2k-1}erfc(bT).$$

We have already estimated the truncation error at k=N above in v_n by replacing n in $|v_n|$ with n+2N. Since for $a \le b$, $c^2 \le 1/2$, one can truncate at about 50 terms to achieve relative errors $O(10^{-15})$. Actually, h_n should have been carried with an arbitrary constant, but replacing $i^n erfc(bw)$ with its asymptotic behavior (leading term) for large w (large T) in the integral representation matches the asymptotic behavior of the first term of the series expansion for large T. This implies that the arbitrary constant should be 1, as we have assumed.

Case II, `a > b`,

In this case, we integrate repeatedly by parts to decrease the order n on $i^n erfc(bw)$. m+1 applications yield

$$Y_n(a,b,T) = \frac{\sqrt{\pi}}{2a}\sum_{k=0}^{m}(-1)^k\left(\frac{b}{a}\right)^k i^k erfc(aT)\, i^{n-k}erfc(bT) + (-1)^{m+1}\frac{\sqrt{\pi}}{2}\left(\frac{b}{a}\right)^{m+1}\int_T^{\infty} i^m erfc(aw)\, i^{n-m-1}erfc(bw)dw$$

and with m=n we get n-m-1=-1, and $i^{-1}erfc(bw) = 2e^{-b^2w^2}/\sqrt{\pi}$. Then the final result for $a > b$ is

$$Y_n(a,b,T) = \frac{\sqrt{\pi}}{2a}\sum_{k=0}^{n}(-1)^k\left(\frac{b}{a}\right)^k i^k erfc(aT)\, i^{n-k}erfc(bT) + (-1)^{n+1}\left(\frac{b}{a}\right)^{n+1} Y_n(b,a,T).$$

Implementation of Cases I and II is carried out in subroutine INTEGI29 for sequences of odd and even n.

Application of $\mathbf{Y_0(a,b,T)}$ to $\mathbf{I_5(a,b,T)}$

n=0 in Cases I and II give the relations for the I function found in Folder 5 with a new formula for $a \le b$,

$$I_5(a,b,T) = Y_0(a,b,T) = \frac{be^{-a^2T^2}}{a^2+b^2}\sum_{k=1}^{\infty}\left(\frac{4a^2}{a^2+b^2}\right)^{k-1}[(k-1)!\,i^{2k-1}erfc(bT)], \qquad a \le b.$$

Application of $\mathbf{Y_0(a,b,T)}$ to $\mathbf{I_3^c(a,b,c,T)}$ of Folder 7

The formula

$$I_3^c(a,b,c,T) \equiv \int_T^{\infty} e^{-c^2w^2}\, erfc(aw)\,erfc(bw)dw = \mathrm{erfc}(aT)I_5(c,b,T) - S(a,b,c,T)$$

$$S(a,b,c,T) = \frac{2a}{\sqrt{\pi}}\int_T^{\infty} e^{-a^2w^2} I_5(c,b,w)dw$$

was derived in Folder 7 as Case I with the restrictions $a,\ c \le b$. All other cases were considered in Case II which reduced the computation to Case I. Since $Y_0(a,b,T) = I_5(a,b,T)$, we can integrate the series to obtain

$$S(a,b,c,T)=\frac{2ab}{\sqrt{\pi}\sqrt{b^2+c^2}}\sum_{k=1}^{\infty}(4\gamma^2)^{k-1}[(k-1)!Y_{2k-1}(\sqrt{a^2+c^2},b,T)]\ ,\quad \gamma^2=\frac{c^2}{b^2+c^2}\le\frac{1}{2}\ ,\qquad a\ ,\ c\le b\,.$$

The convergence of the series follows from the estimates

$$(4\gamma^2)^{k-1}[\Gamma(k)Y_{2k-1}(\sqrt{a^2+c^2},b,T)]\le(2\gamma)^{2k-2}\Gamma(k)\frac{e^{-(a^2+c^2)T^2}}{b}i^{2k}erfc(bT)\le(2\gamma)^{2k-2}\frac{e^{-(a^2+c^2)T^2}}{b}\frac{\Gamma(k)}{2^{2k}\Gamma(k+1)}$$

$$\le\gamma^{2k-2}\frac{e^{-(a^2+c^2)T^2}}{4bk}=O\left(\frac{\gamma^{2k-2}}{k}\right),\qquad \gamma^2=\frac{c^2}{b^2+c^2},\ \ \gamma^2\le\frac{1}{2}$$

As before, one can expect errors $O(10^{-15})$ with truncation of the series at k=N=50. The computation of S(a,b,c,T) by means of $Y_{2k-1}(\sqrt{a^2+c^2},b,T)$ is a better choice than the computation in Folder 28 because the risk of losses of significance by small differences of large numbers is less. Numerical experiments comparing the quadrature of S with the above series for S are consistent with this analysis. Subroutine INTEGI29 can be used to generate the sequences of odd order functions $Y_{2k-1}(\sqrt{a^2+c^2},b,T)$ for this application.

Special Cases

Integral of $\mathbf{I_5(a,b,T)}$

Integration of this relation for $I_5(a,b,T)$ for Case I, $a\le b$, gives

$$\int_T^{\infty}I_5(a,b,w)dw=\frac{b}{a^2+b^2}\sum_{k=1}^{\infty}\left(\frac{4a^2}{a^2+b^2}\right)^{k-1}[(k-1)!\,Y_{2k-1}(a,b,T)],\qquad a\le b$$

and integration of Case II, $a>b$, in Folder 5 gives

$$\int_T^{\infty}I_5(a,b,w)dw=\frac{\sqrt{\pi}}{2a}\int_T^{\infty}erfc(aw)erfc(bw)dw-\frac{b}{a}\int_T^{\infty}I_5(b,a,w)dw\qquad a>b$$

where the $I_5(b,a,T)$ integral on the right is computed above since the first parameter is less than the second parameter. The integral

$$I_2^c(a,b,c,T)=\int_T^{\infty}erfc(aw)erfc(bw)dw$$

is computed in Folder 9, but we can get an alternate form from

$$Y_1(a,b,T)=\int_T^{\infty}e^{-a^2w^2}\,\mathrm{ierfc(bw)dw}$$

by integration by parts

$$Y_1(a,b,T)=\int_T^{\infty}e^{-a^2w^2}\,\mathrm{ierfc(bw)dw}=\frac{\sqrt{\pi}}{2a}erfc(aT)\,ierfc(bT)-\frac{\sqrt{\pi}}{2}\frac{b}{a}\int_T^{\infty}erfc(aw)erfc(bw)dw$$

or

$$I_2^c(a,b,c,T)=\int_T^{\infty}erfc(aw)erfc(bw)dw=\frac{1}{b}\left[erfc(aT)\,ierfc(bT)-\frac{2a}{\sqrt{\pi}}Y_1(a,b,T\right]\ ,$$

with

$$Y_1(a,b,T)=\frac{be^{-a^2T^2}}{a^2+b^2}\sum_{k=1}^{\infty}\left(\frac{4a^2}{a^2+b^2}\right)^{k-1}\frac{\Gamma(1/2+k)}{\Gamma(3/2)}i^{2k}erfc(bT),\quad \frac{\Gamma(1/2+k)}{\Gamma(3/2)}=2\,(1/2)_k,\ \ a\le b$$

For this computation we can always choose $a \le b$ since the integral is symmetric in a and b and a can be chosen to be the smaller of the two parameters.

b=a in Case II

Although we chose to take $a \le b$ in Case I and $a > b$ in Case II to get rapid convergence, the results apply for any a and b. Thus with b=a in Case II, we get

$$Y_n(a,a,T) = \frac{\sqrt{\pi}}{2a}\sum_{k=0}^{n}(-1)^k\, i^k erfc(aT)\, i^{n-k} erfc(aT) + (-1)^{n+1}\, Y_n(a,a,T)$$

and this yields

$$[1-(-1)^{n+1}]Y_n(a,a,T) = \frac{\sqrt{\pi}}{2a}\sum_{k=0}^{n}(-1)^k\, i^k erfc(aT)\, i^{n-k} erfc(aT)$$

or

$$2Y_n(a,a,T) = \frac{\sqrt{\pi}}{2a}\sum_{k=0}^{n}(-1)^k\, i^k erfc(aT)\, i^{n-k} erfc(aT), \qquad n\ even$$

and

$$\frac{\sqrt{\pi}}{2a}\sum_{k=0}^{n}(-1)^k\, i^k erfc(aT)\, i^{n-k} erfc(aT) = 0, \qquad n\ odd$$

Representations of $\mathbf{erfc(T\sqrt{a^2+b^2})}$ and $\mathbf{ierfc(T\sqrt{a^2+b^2})}$

If we let n=-1 in the general formula for $Y_n(a,b,T)$, we get

$$Y_{-1}(a,b,T) = \frac{erfc(T\sqrt{a^2+b^2})}{\sqrt{a^2+b^2}} = \frac{be^{-a^2T^2}}{a^2+b^2}\sum_{k=1}^{\infty}\left(\frac{4a^2}{a^2+b^2}\right)^{k-1}\frac{\Gamma(-1/2+k)}{\sqrt{\pi}} i^{2k-2} erfc(bT)$$

and integration again gives

$$\frac{ierfc(T\sqrt{a^2+b^2})}{a^2+b^2} = \frac{b}{a^2+b^2}\sum_{k=1}^{\infty}\left(\frac{4a^2}{a^2+b^2}\right)^{k-1}\frac{\Gamma(-1/2+k)}{\sqrt{\pi}} Y_{2k-2}(a,b,T).$$

Since these expressions are symmetric in a and b, we can, without loss of generality, take a as the smaller of the two parameters.

Evaluation of $\mathbf{Y_n(a,b,0)}$

If we use $i^n erfc(0) = 1/[2^n\Gamma(1+n/2)]$ in the expression for $Y_n(a,b,0)$ we get

$$\begin{aligned}Y_n(a,b,0) &= \frac{b}{a^2+b^2}\sum_{k=1}^{\infty}\left(\frac{4a^2}{a^2+b^2}\right)^{k-1}\frac{1}{2^{n+2k-1}\Gamma(k+(n+1)/2)} \\ &= \frac{b}{a^2+b^2}\frac{1}{2^{n+1}\Gamma(1+(n+1)/2)}\sum_{k=0}^{\infty}\left(\frac{a^2}{a^2+b^2}\right)^{k}\frac{1}{(1+(n+1)/2)_k}\end{aligned}$$

Now this last series can be identified as a confluent hypergeometric function $\phi(1,1+\alpha,x)$ with $x = a^2/(a^2+b^2)$ and $\alpha = (n+1)/2$. Because the first parameter is 1, we can further identify this hypergeometric function as an incomplete gamma function $\alpha x^{-\alpha}e^x\gamma(\alpha,x)$. Then, for $a \le b$

$$Y_n(a,b,0) = \frac{b}{2^{n+1} a^2 \Gamma((n+1)/2)} \left(\frac{a^2 + b^2}{a^2} \right)^{(n-1)/2} e^{a^2/(a^2+b^2)} \gamma((n+1)/2\ ,\ a^2/(a^2+b^2))$$

The expression for a>b reduces to

$$Y_n(a,b,0) = \frac{\sqrt{\pi}}{2^{n+1} a} \sum_{k=0}^{n} (-1)^k \left(\frac{b}{a} \right)^k \frac{1}{\Gamma(1+k/2)\Gamma(1+(n-k)/2)} + (-1)^{n+1} \left(\frac{b}{a} \right)^{n+1} Y_n(b,a,0)\,.$$

4 . Fortran Codes

This chapter presents the callable routines developed in the course of the research on the integrals of Chapter 3. Executable programs are found in driver files denoted by BECKDRVR.FOR, AMOSDRVR.FOR or RESEARCH.FOR. The main library containing supporting subroutines is formed from AMOSSUBS.FOR+BECKSUBS.FOR. The driver program and the main library must be compiled and linked to form an executable unit.

The files listed below are stored as text files on a disk which is attached at the end of this handbook.

Disk Contents

Callable Subroutines and Functions in File AMOSSUBS.FOR

	CALLABLE SUBROUTINE:	**FOLDER**
Q	DGAUS8(FUN, A, B, ERR, ANS, IERR)	SLATEC
$Q=\sum Q_{k_k}$	DQUAD8(DQFUN,INIT,X1,SIG,REL,X2,QANS,IERR)	21
$E_{N+k}(x)$	DEXINT(X, N, KODE, M, TOL, EN, NZ, IERR)	SLATEC
$E_{1/2+k}(x)$	DHEXINT(X, FNH, KODE, M, TOL, EN, NZ, IERR)	18
$G_{N+k}(x)$	DGEXINT(X, N, KODE, M, TOL, GN, EN, NZ, IERR)	18
$G_{1/2+k}(x)$	DGHEXINT(X, FNH, KODE, M, TOL, GN, EN, NZ, IERR)	18
$ierfc(x)$	DIERFC(X,KODE,ANS,IERR)	10,16
$i^{N_0+k}erfc(x)$	DINERFC(X,N0,KODE,N,REL,Y,NZ)	SLATEC
$\frac{(-1)^{k+1}}{k!}\psi^{(k)}(x)$	DPSIFN(X, N, KODE, M, ANS, NZ, IERR)	SLATEC
	XERROR(MESS,NMESS,NERR,LEVEL)	SLATEC
	XERRWV(MESSG,NMESSG,NERR,LEVEL,NI,I1,I2,NR,R1,R2)	SLATEC
	FDUMP	SLATEC

	CALLABLE FUNCTION:	**FOLDER**
$erf(x), erfc(x)$	DOUBLE PRECISION FUNCTION DRERF(X,KODE,NZ)	AMOSLIB
$F(x)$	DOUBLE PRECISION FUNCTION DFERF(X,REL,IERR)	16
$G(x)$	DOUBLE PRECISION FUNCTION DGERFC(X,KODE,REL,IERR)	16
$H_{23}(x)$	DOUBLE PRECISION FUNCTION DHERFC(X)	23
$\psi(N)$	DOUBLE PRECISION FUNCTION DPSIXN(N)	SLATEC
$\ln\Gamma(z)$	DOUBLE PRECISION FUNCTION DGAMLN(Z,IERR)	SLATEC
	DOUBLE PRECISION FUNCTION D1MACH(I)	SLATEC
	INTEGER FUNCTION I1MACH(I)	SLATEC
	REAL FUNCTION R1MACH(I)	SLATEC

Callable Subroutines and Functions in File BECKSUBS.FOR

CALLABLE SUBROUTINE:	FOLDER
INTEGI1(A,B,T,KODE,REL,ANSI1,IERR)	1,2,10
INTEGI2(A,B,T,KODE,ANSI2,IERR)	9
INTEGI9(A,B,T,KODE,ANSI9,IERR)	9
INTEGI3(A,B,C,T,ANSI3,IERR,KFORM)	7
INTEGI5(A,B,X,ANSI5,IERR)	5
INTEGJ5(A,B,X,ANSJ5,IERR)	5
INTEGV5(A,B,X,ANSV5,IERR)	5
INTEGI6(A,B,T,KODE,ANSI6,IERR)	3,6,15
INTEGP(A,B,T,KODE,REL,PANS,IERR)	11
INTEGQ(A,B,T,REL,QANS,IERR)	11
INTEGW3(A,B,T,KODE,REL,ANSW3,IERR)	10
INTEGI21(A,B,C,T,KODE,ANSI21,I21ERR,KFORM)	21
INTEGJ21(A,B,C,T,ANSJ21,J21ERR,KFORM)	21
INTEGI22(A,B,C,T,ANSI22,I22ERR,KFORM)	22
INTEGJ22(A,B,C,T,ANSJ22,J22ERR,KFORM)	22
INTEGI29(A,B,T,N0,NN,YN,IERR)	29
INTEGS1(A,B,C,T,TOL,S1,IERR,KFORM)	21
INTEGS2(A,B,C,T,TOL,S2,IERR,KFORM)	22
GNSEQ(A,B,CAPT,M,REL,YN)	21

CALLABLE FUNCTION:	FOLDER
DVOFT(A,B,T,REL,IERR,KFORM)	21
PHIZ(Z)	5

Executable Programs in File BECKDRVR.FOR

PROGRAM:	OUTPUT:	FOLDER
PROGRAM I1COMP	I1COMP.TXT	1,2,10
PROGRAM I2COMP	I2COMP.TXT	9
PROGRAM I9COMP	I9COMP.TXT	9
PROGRAM I3COMP	I3COMP.TXT	7
PROGRAM I5COMP	I5COMP.TXT	5
PROGRAM J5COMP	J5COMP.TXT	5
PROGRAM V5COMP	V5COMP.TXT	5
PROGRAM I6COMP	I6COMP.TXT	3,6,15
PROGRAM W3COMP	W3COMP.TXT	10
PROGRAM PCOMP	PCOMP.TXT	11
PROGRAM QCOMP	QCOMP.TXT	11
PROGRAM I21COMP	I21COMP.TXT	21
PROGRAM J21COMP	J21COMP.TXT	21
PROGRAM I22COMP	I22COMP.TXT	22
PROGRAM J22COMP	J22COMP.TXT	22
PROGRAM I29COMP	I29COMP.TXT	29
PROGRAM GNCOMP	GNCOMP.TXT	21
PROGRAM VTCOMP	VTCOMP.TXT	21

Executable Programs in File AMOSDRVR.FOR

PROGRAM:	OUTPUT:	FOLDER
PROGRAM GECOMP	GECOMP.TXT	18
PROGRAM GHECOMP	GHECOMP.TXT	18
PROGRAM DFCOMP	DFCOMP.TXT	16
PROGRAM DGCOMP	DGCOMP.TXT	16
PROGRAM HERFCOMP	HERFCOMP.TXT	23

Executable Programs in File RESEARCH.FOR

PROGRAM:	OUTPUT:	FOLDER
PROGRAM I1COMPB	I1COMPB.TXT	1,2
PROGRAM I4COMP	I4COMP.TXT	8
PROGRAM J4COMP	J4COMP.TXT	8
PROGRAM ERFINT	ERFINT.TXT	12
PROGRAM I13COMP	I13COMP.TXT	13
PROGRAM I14COMP	I14COMP.TXT	14
PROGRAM I19COMP	I19COMP.TXT	19
PROGRAM I20COMP	I20COMP.TXT	20
PROGRAM I24COMP	I24COMP.TXT	24
PROGRAM J24COMP	J24COMP.TXT	24
PROGRAM V24COMP	V24COMP.TXT	24
PROGRAM I25COMP	I25COMP.TXT	25
PROGRAM I26COMP	I26COMP.TXT	26
PROGRAM I26ACOMP	I26ACOMP.TXT	26
PROGRAM DGSCOMP	DGSCOMP.TXT	16

FILE DESCRIPTIONS

The following files contain the programs and subroutines which were used to check out formulas numerically.

BECKSUBS.FOR is a file of subroutines and functions which implement many of the formulae in developed in Chapter 3. These can be regarded as "complete", which means that these codes were constructed with high accuracy over large ranges of variables in mind and contain parameters which record input and output errors when some condition is violated.

AMOSSUBS.FOR is a file which contains codes published in ACM Collected Algorithms or the SLATEC library, codes copied from a personal archive called AMOSLIB, or codes developed as natural extensions of those in the SLATEC Library or AMOSLIB. High accuracy is the dominant consideration in the development of these codes.

BECKDRVR.FOR is a file of drivers (PROGRAM…) which exercise the subroutines of BECKSUBS.FOR, and, when successfully completed, show typical relative errors when compared with an alternate method of computation. For integrals, this alternate method is usually a direct quadrature with DGAUS8 or DQUAD8.

AMOSDRVR.FOR is a file of driver routines which exercise not previously published subroutines in AMOSSUBS.FOR in the manner of BECKDRVR.FOR.

RESEARCH.FOR is a file containing codes which implement a formula or procedure but does not contain error checking nor flags for unusual occurrences. A code with this designation is *not* to be considered "complete" or algorithmic (where accurate values over stated ranges of variables are returned).

USAGE: To execute a driver program, it must be extracted from BECKDRVR.FOR, AMOSDRVR.FOR, or RESEARCH.FOR, compiled and linked to the compiled files AMOSSUBS.FOR and BECKSUBS.FOR. Each of these files starts with a code consisting of all comment lines with text which describes the contents of the file. These information subroutines are labeled BPRGINFO, APRGINFO, RPRGINFO, AMOSINFO, and BECKINFO, respectively.

MACHINE DEPENDANT CONSTANTS: The FORTRAN code distributed with this document contains machine dependant functions

```
          INTEGER FUNCTION  I1MACH(I),    I=1,16
             REAL FUNCTION  R1MACH(I),    I=1,5
DOUBLE PRECISION FUNCTION  D1MACH(I),    I=1,5
```

which define important machine constants (File AMOSSUBS.FOR). Some of the codes which were adapted from the SLATEC library require these functions in order to compute properly. These functions have FORTRAN code which returns machine constants for a variety of machines. These are set in comment statements. To define a machine, simply remove the C in column 1 and re-comment any active FORTRAN code from a previous setting. The default settings define the IBM PC which will work for many other personal computers. The prologue of each function defines the value returned for each I. To see the numeric values which will define your machine, simply evaluate each function in a loop and print out the values.

```
      SUBROUTINE AMOSINFO
C-----------------------------------------------------------------------
C     DONALD E. AMOS, MAY 1, 2002
C
C     EACH CALLABLE ROUTINE HAS A REFERENCE IN THE PROLOGUE TO
C     APPROPRIATE FOLDERS WHICH DESCRIBE THE ANALYTICAL BASIS OF THE
C     CODE.
C
C     USAGE:
C
C     COMPILE THIS FILE AND LINK IT TO ANY PROGRAM NEEDING A SUBROUTINE
C     OR FUNCTION FROM THE FOLLOWING LIST:
C
C     CALLABLE SUBROUTINES:                                   FOLDER
C           DGAUS8(FUN, A, B, ERR, ANS, IERR)                 SLATEC
C           DQUAD8(DQFUN,INIT,X1,SIG,REL,X2,QANS,IERR)          21
C           DEXINT(X, N, KODE, M, TOL, EN, NZ, IERR)          SLATEC
C           DHEXINT(X, FNH, KODE, M, TOL, EN, NZ, IERR)         18
C           DGEXINT(X, N, KODE, M, TOL, GN, EN, NZ, IERR)       18
C           DGHEXINT(X, FNH, KODE, M, TOL, GN, EN, NZ, IERR)    18
C           DIERFC(X,KODE,ANS,IERR)                           10,16
C           DINERFC(X,N0,KODE,N,REL,Y,NZ)                     SLATEC
C           DPSIFN(X, N, KODE, M, ANS, NZ, IERR)              SLATEC
C           XERROR(MESS,NMESS,NERR,LEVEL)                     SLATEC
C           XERRWV(MESSG,NMESSG,NERR,LEVEL,NI,I1,I2,NR,R1,R2) SLATEC
C           FDUMP                                             SLATEC
C
C      CALLABLE FUNCTIONS:
C           DOUBLE PRECISION FUNCTION DRERF(X,KODE,NZ)        AMOSLIB
C           DOUBLE PRECISION FUNCTION DFERF(X,REL,IERR)         16
C           DOUBLE PRECISION FUNCTION DGERFC(X,KODE,REL,IERR)   16
C           DOUBLE PRECISION FUNCTION DHERFC(X)                 23
C           DOUBLE PRECISION FUNCTION DPSIXN(N)               SLATEC
C           DOUBLE PRECISION FUNCTION DGAMLN(Z,IERR)          SLATEC
C           DOUBLE PRECISION FUNCTION D1MACH(I)               SLATEC
C           INTEGER FUNCTION I1MACH(I)                        SLATEC
C           REAL FUNCTION R1MACH(I)                           SLATEC
C-----------------------------------------------------------------------
      END
      DOUBLE PRECISION FUNCTION DRERF(X,KODE,NZ)
C
C     WRITTEN BY D.E.AMOS AND S.L.DANIEL, JANUARY, 1975.
C                MODIFIED MAY, 1990; MARCH, 2002: JANUARY, 2003
C
C     REFERENCE: SANDIA LABORATORY REPORT SC-DR-72-0918
C
C     ABSTRACT
C         DRERF COMPUTES THE ERROR FUNCTION ERF(X) AND THE COERROR
C         FUNCTION ERFC(X)=1.-ERF(X) FOR REAL, UNRESTRICTED X. TO RETAIN
C         SIGNIFICANT DIGITS THE FORMULAE
C
C                -1                              2
C                -1+EXP(-X*X)*R/ABS(X)           2-EXP(-X*X)*R/ABS(X)
C                -1+EXP(-X*X)*S/ABS(X)           2-EXP(-X*X)*S/ABS(X)
C         ERF(X)= X*T                    ERFC(X)=1-X*T
C                 1-EXP(-X*X)*U/X                EXP(-X*X)*U/X
C                 1-EXP(-X*X)*V/X                EXP(-X*X)*V/X
C                 1                              0
C
C         ARE USED WHERE THE BREAK POINTS ARE
C
C             X= -6, -4, -2, +2, +4, +6   FOR ERF(X)
C
C         AND
C             X= -6, -4, -2, +2, +4       FOR ERFC(X)
C
C         HERE R, S, T, U AND V ARE CHEBYSHEV EXPANSIONS
```

```
C         ON APPROPRIATE INTERVALS. A SCALING OPTION, EXP(X**2)*ERFC(X)
C         FOR X.GE.0.0, IS ALSO PROVIDED.
C
C     DESCRIPTION OF ARGUMENTS
C
C         INPUT
C           X      - X,UNRESTRICTED
C           KODE   - A SELECTION PARAMETER
C                    KODE=1 RETURNS DRERF=ERF(X)
C                    KODE=2 RETURNS DRERF=ERFC(X)
C                    KODE=3 RETURNS DRERF=         ERFC(X),  X.LT.0.0
C                                   DRERF=EXP(X*X)*ERFC(X),  X.GE.0.0
C
C         OUTPUT
C           DRERF  - VALUE FOR ERF(X),ERFC(X), OR EXP(X**2)*ERFC(X)
C                    DEPENDING ON KODE
C           NZ     - UNDERFLOW FLAG
C                    NZ=0  NORMAL RETURN
C                    NZ=1  UNDERFLOW FOR KODE=2,
C                          DRERF=0.0 RETURNED
C
C     ERROR CONDITIONS
C         KODE NOT 1, 2, OR 3 IS A FATAL ERROR
C         UNDERFLOW IS A NON-FATAL ERROR, DRERF=0.0 AND NZ=1 RETURNED
C
C     RERF   USES SUBROUTINES I1MACH,D1MACH,XERROR
C-----------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION A1(24),A2(23),A3(17)
C     CHEBYSHEV COEFFICIENTS FOR [0,2]
      DATA   (A1(K),K=1,24)  /
     1      0.832316137886179500D+00,     -0.341657034193776500D+00,
     2     -0.152213336353308800D-01,      0.271881389879122100D-01,
     3     -0.443860662153016500D-02,     -0.886371605188296700D-03,
     4      0.380536428195736700D-03,     -0.571020720810514000D-05,
     5     -0.170532778625881700D-04,      0.207469149563607100D-05,
     6      0.475742503160444100D-06,     -0.119217251859814700D-06,
     7     -0.662105198579041900D-08,      0.436703352221523300D-08,
     8     -0.108586273948044500D-09,     -0.118452196128678300D-09,
     9      0.103592366410598800D-10,      0.245412015623528400D-11,
     A     -0.391710955513175500D-12,     -0.369418614881312400D-13,
     B      0.106467918695034400D-13,      0.284514287425412300D-15,
     C     -0.232515791318906500D-15,      0.453152396988929400D-17/
C     CHEBYSHEV COEFFICIENTS FOR [4,6]
      DATA (A2(K),K=1,23)  / 5.55925693425176D-01, 0.00000000000000D+00,
     1-8.09033209211571D-03, 0.00000000000000D+00, 1.67768369286852D-04,
     2 0.00000000000000D+00,-5.53104163380836D-06, 0.00000000000000D+00,
     3 2.44369364046347D-07, 0.00000000000000D+00,-1.33270043958223D-08,
     4 0.00000000000000D+00, 8.55027489170315D-10, 0.00000000000000D+00,
     5-6.25400760100857D-11, 0.00000000000000D+00, 5.10173516527965D-12,
     6 0.00000000000000D+00,-4.56694846856777D-13, 0.00000000000000D+00,
     7 4.43106374588405D-14, 0.00000000000000D+00,-4.61460815796214D-15/
C     CHEBYSHEV COEFFICIENTS FOR [2,4]
      DATA (A3(K),K=1,17)  / 5.33315442660327D-01, 1.78876062098494D-02,
     1-3.80175293816804D-03, 6.97111435056370D-04,-1.16368846095251D-04,
     2 1.81367676091642D-05,-2.67719941001743D-06, 3.77701336341082D-07,
     3-5.12491193181002D-08, 6.71870624801104D-09,-8.54020784885325D-10,
     4 1.05545351331091D-10,-1.27109174471917D-11, 1.49449942042063D-12,
     5-1.71824290388960D-13, 1.93435061097394D-14,-2.13482679031942D-15/
      DATA N1,N2,N3/24,23,17/
C-----------------------------------------------------------------------
      IF(KODE.LT.1 .OR. KODE.GT.3) GO TO 900
      NZ=0
      XX=X
      IF(KODE.GT.1) GOTO 200
C     COMPUTATION OF ERF FUNCTION   (KODE=1)
  100 CONTINUE
```

```
      IF(XX.GT.-6.0D0) GO TO 10
      DRERF=-1.0D0
      RETURN
   10 IF(XX.LT.6.0D0) GO TO 12
      DRERF=1.0D0
      RETURN
   12 CONTINUE
      IF(XX.LT.4.0D0) GO TO 33
C     INTERVAL (4,6)
      APB=0.0D0
      BMA=XX
      XD=2.0D0
      CALL DCHBYS(APB,BMA,XD,N2,A2,SUM)
      DRERF=1.0D0-SUM*DEXP(-XX*XX)/XX
      RETURN
   33 CONTINUE
      IF(XX.LE.2.0D0) GO TO 40
C     INTERVAL (2,4)
      APB=6.0D0
      BMA=2.0D0
      CALL DCHBYS(APB,BMA,XX,N3,A3,SUM)
      DRERF=1.0D0-SUM*DEXP(-XX*XX)/XX
      RETURN
   40 CONTINUE
      IF(XX.LT.-2.0D0) GO TO 50
C     INTERVAL (-2,2)
      APB=2.0D0
      BMA=2.0D0
      AXX=DABS(XX)
      CALL DCHBYS(APB,BMA,AXX,N1,A1,SUM)
      DRERF=XX*SUM
      RETURN
   50 CONTINUE
      IF(XX.LE.-4.0D0) GO TO 60
C     INTERVAL (-4,-2)
      XX=-XX
      APB=6.0D0
      BMA=2.0D0
      CALL DCHBYS(APB,BMA,XX,N3,A3,SUM)
      DRERF=-1.0D0+SUM*DEXP(-XX*XX)/XX
      RETURN
   60 CONTINUE
C     INTERVAL (-6,-4)
      XX=-XX
      APB=0.0D0
      BMA=XX
      XD=2.0D0
      CALL DCHBYS(APB,BMA,XD,N2,A2,SUM)
      DRERF=-1.0D0+SUM*DEXP(-XX*XX)/XX
      RETURN
C     COMPUTATION OF ERFC FUNCTION (KODE=2 AND KODE=3)
  200 CONTINUE
      IF(XX.GT.-6.0D0) GO TO 205
      DRERF=2.0D0
      RETURN
  205 CONTINUE
      KK=MIN(IABS(I1MACH(15)),IABS(I1MACH(16)))
      ELIM=2.303D0*(DBLE(FLOAT(KK))*D1MACH(5)-3.0D0)
      XLIM=DSQRT(ELIM)
      IF(XX.LT.XLIM) GO TO 210
C     INTERVAL (XLIM,INF)
      IF(KODE.EQ.3) THEN
        APB=0.0D0
        BMA=XX
        XD=2.0D0
        CALL DCHBYS(APB,BMA,XD,N2,A2,SUM)
        DRERF=SUM/XX
```

```
          RETURN
        ELSE
          NZ=1
          DRERF=0.0D0
          RETURN
        ENDIF
  210 CONTINUE
      IF(XX.LT.4.0D0) GO TO 215
C     INTERVAL (4,XLIM)
      APB=0.0D0
      BMA=XX
      XD=2.0D0
      CALL DCHBYS(APB,BMA,XD,N2,A2,SUM)
      IF(KODE.EQ.3) THEN
        DRERF=SUM/XX
        RETURN
      ELSE
        DRERF=SUM*DEXP(-XX*XX)/XX
        RETURN
      ENDIF
  215 CONTINUE
      IF(XX.GT.2.0D0) GO TO 220
      IF(XX.LT.-2.0D0) GO TO 225
C     INTERVAL (-2,2)
      APB=2.0D0
      BMA=2.0D0
      AXX=DABS(XX)
      CALL DCHBYS(APB,BMA,AXX,N1,A1,SUM)
      SUM=XX*SUM
      IF(X.GT.0.0D0 .AND. KODE.EQ.3) THEN
        DRERF=(1.0D0-SUM)*DEXP(XX*XX)
        RETURN
      ELSE
        DRERF=1.0D0-SUM
        RETURN
      ENDIF
  220 CONTINUE
C     INTERVAL (2,4)
      APB=6.0D0
      BMA=2.0D0
      CALL DCHBYS(APB,BMA,XX,N3,A3,SUM)
      IF(KODE.EQ.3) THEN
        DRERF=SUM/XX
        RETURN
      ELSE
        DRERF=SUM*DEXP(-XX*XX)/XX
        RETURN
      ENDIF
  225 CONTINUE
      IF(XX.GT.-4.0D0) GO TO 230
C     INTERVAL (-INF,-4)
      IF(XX.LT.-6.0D0) THEN
        DRERF=2.0D0
        RETURN
      ENDIF
      XX=-XX
      APB=0.0D0
      BMA=XX
      XD=2.0D0
      CALL DCHBYS(APB,BMA,XD,N2,A2,SUM)
      DRERF=2.0D0-SUM*DEXP(-XX*XX)/XX
      RETURN
  230 CONTINUE
C     INTERVAL (-4,-2)
      XX=-XX
      APB=6.0D0
      BMA=2.0D0
```

```
      CALL DCHBYS(APB,BMA,XX,N3,A3,SUM)
      DRERF=2.0D0-SUM*DEXP(-XX*XX)/XX
      RETURN
  900 CONTINUE
      CALL XERROR(' IN DRERF, KODE NOT 1, 2, OR 3',30,1,2)
      RETURN
      END
      SUBROUTINE DGAUS8(FUN, A, B, ERR, ANS, IERR)
C
C     WRITTEN BY R.E. JONES
C
C     ABSTRACT  *** A DOUBLE PRECISION ROUTINE ***
C        DGAUS8 INTEGRATES REAL FUNCTIONS OF ONE VARIABLE OVER FINITE
C        INTERVALS USING AN ADAPTIVE 8-POINT LEGENDRE-GAUSS ALGORITHM.
C        DGAUS8 IS INTENDED PRIMARILY FOR HIGH ACCURACY INTEGRATION
C        OR INTEGRATION OF SMOOTH FUNCTIONS.
C
C        THE MAXIMUM NUMBER OF SIGNIFICANT DIGITS OBTAINABLE IN ANS
C        IS THE SMALLER OF 18 AND THE NUMBER OF DIGITS CARRIED IN
C        DOUBLE PRECISION ARITHMETIC.
C
C        DGAUS8 CALLS I1MACH, D1MACH, XERROR
C
C     DESCRIPTION OF ARGUMENTS
C
C        INPUT--* FUN,A,B,ERR ARE DOUBLE PRECISION *
C        FUN - NAME OF EXTERNAL FUNCTION TO BE INTEGRATED. THIS NAME
C              MUST BE IN AN EXTERNAL STATEMENT IN THE CALLING PROGRAM.
C              FUN MUST BE A DOUBLE PRECISION FUNCTION OF ONE DOUBLE
C              PRECISION ARGUMENT. THE VALUE OF THE ARGUMENT TO FUN
C              IS THE VARIABLE OF INTEGRATION WHICH RANGES FROM A TO
C              B.
C        A   - LOWER LIMIT OF INTEGRAL
C        B   - UPPER LIMIT OF INTEGRAL (MAY BE LESS THAN A)
C        ERR - IS A REQUESTED PSEUDORELATIVE ERROR TOLERANCE.  NORMALLY
C              PICK A VALUE OF DABS(ERR) SO THAT DTOL.LT.DABS(ERR).LE.
C              1.0D-3 WHERE DTOL IS THE LARGER OF 1.0D-18 AND THE
C              DOUBLE PRECISION UNIT ROUNDOFF = D1MACH(4). ANS WILL
C              NORMALLY HAVE NO MORE ERROR THAN DABS(ERR) TIMES THE
C              INTEGRAL OF THE ABSOLUTE VALUE OF FUN(X). USUALLY,
C              SMALLER VALUES OF ERR YIELD MORE ACCURACY AND REQUIRE
C              MORE FUNCTION EVALUATIONS.
C
C              A NEGATIVE VALUE FOR ERR CAUSES AN ESTIMATE OF THE
C              ABSOLUTE ERROR IN ANS TO BE RETURNED IN ERR. NOTE THAT
C              ERR MUST BE A VARIABLE (NOT A CONSTANT) IN THIS CASE.
C              NOTE ALSO THAT THE USER MUST RESET THE VALUE OF ERR
C              BEFORE MAKING ANY MORE CALLS THAT USE THE VARIABLE ERR.
C
C        OUTPUT--* ERR,ANS ARE DOUBLE PRECISION *
C        ERR - WILL BE AN ESTIMATE OF THE ABSOLUTE ERROR IN ANS IF THE
C              INPUT VALUE OF ERR WAS NEGATIVE.  (ERR IS UNCHANGED IF
C              THE INPUT VALUE OF ERR WAS NONNEGATIVE.)  THE ESTIMATED
C              ERROR IS SOLELY FOR INFORMATION TO THE USER AND SHOULD
C              NOT BE USED AS A CORRECTION TO THE COMPUTED INTEGRAL.
C        ANS - COMPUTED VALUE OF INTEGRAL
C        IERR- A STATUS CODE
C            --NORMAL CODES
C               1 ANS MOST LIKELY MEETS REQUESTED ERROR TOLERANCE,
C                 OR A=B.
C              -1 A AND B ARE TOO NEARLY EQUAL TO ALLOW NORMAL
C                 INTEGRATION.  ANS IS SET TO ZERO.
C            --ABNORMAL CODE
C               2 ANS PROBABLY DOES NOT MEET REQUESTED ERROR TOLERANCE.
C***END PROLOGUE
      EXTERNAL FUN
      INTEGER ICALL, IERR, K, KML, KMX, L, LMN, LMX, LR, MXL, NBITS,
```

```
     * NIB, NLMN, NLMX
      INTEGER I1MACH
      DOUBLE PRECISION A,AA,AE,ANIB,ANS,AREA,B,C,CE,DTOL,EE,EF,
     * EPS, ERR, EST, GL, GLR, GR, HH, SQ2, TOL, VL, VR, W1, W2, W3,
     * W4, X1, X2, X3, X4, X, H
      DOUBLE PRECISION D1MACH, G8, FUN
      DIMENSION AA(30), HH(30), LR(30), VL(30), GR(30)
      DATA X1, X2, X3, X4/
     1     1.83434642495649805D-01,     5.25532409916328986D-01,
     2     7.96666477413626740D-01,     9.60289856497536232D-01/
      DATA W1, W2, W3, W4/
     1     3.62683783378361983D-01,     3.13706645877887287D-01,
     2     2.22381034453374471D-01,     1.01228536290376259D-01/
      DATA ICALL  /  0  /
      DATA SQ2/1.41421356D0/
      DATA NLMN/1/,KMX/5000/,KML/6/
      G8(X,H)=H*((W1*(FUN(X-X1*H) + FUN(X+X1*H))
     1           +W2*(FUN(X-X2*H) + FUN(X+X2*H)))
     2          +(W3*(FUN(X-X3*H) + FUN(X+X3*H))
     3           +W4*(FUN(X-X4*H) + FUN(X+X4*H))))
C
C     INITIALIZE
C
      IF (ICALL.NE.0) CALL XERROR('DGAUS8- DGAUS8 CALLED RECURSIVELY. RE
     *CURSIVE CALLS ARE ILLEGAL IN FORTRAN.',74, 7, 2)
      ICALL = 1
      DTOL=D1MACH(4)
      DTOL=DMAX1(DTOL,1.0D-18)
      IF(DABS(ERR).LT.DTOL) GO TO 150
      K = I1MACH(14)
      ANIB = D1MACH(5)*DBLE(FLOAT(K))/0.30102000D0
      NBITS = INT(SNGL(ANIB))
      NLMX = (NBITS*5)/8
      ANS = 0.0D0
      IERR = 1
      CE = 0.0D0
      IF (A.EQ.B) GO TO 140
      LMX = NLMX
      LMN = NLMN
      IF (B.EQ.0.0D0) GO TO 10
      IF (DSIGN(1.0D0,B)*A.LE.0.0D0) GO TO 10
      C = DABS(1.0D0-A/B)
      IF (C.GT.0.1D0) GO TO 10
      IF (C.LE.0.0D0) GO TO 140
      ANIB = 0.5D0 - DLOG(C)/0.69314718D0
      NIB = INT(SNGL(ANIB))
      LMX = MIN0(NLMX,NBITS-NIB-7)
      IF (LMX.LT.1) GO TO 130
      LMN = MIN0(LMN,LMX)
   10 TOL = DMAX1(DABS(ERR),2.0D0**(5-NBITS))/2.0D0
      IF (ERR.EQ.0.0D0) TOL = DSQRT(D1MACH(4))
      EPS = TOL
      HH(1) = (B-A)/4.0D0
      AA(1) = A
      LR(1) = 1
      L = 1
      EST = G8(AA(L)+2.0D0*HH(L),2.0D0*HH(L))
      K = 8
      AREA = DABS(EST)
      EF = 0.5D0
      MXL = 0
C
C     COMPUTE REFINED ESTIMATES, ESTIMATE THE ERROR, ETC.
C
   20 GL = G8(AA(L)+HH(L),HH(L))
      GR(L) = G8(AA(L)+3.0D0*HH(L),HH(L))
      K = K + 16
```

```
      AREA = AREA + (DABS(GL)+DABS(GR(L))-DABS(EST))
C     IF (L.LT.LMN) GO TO 11
      GLR = GL + GR(L)
      EE = DABS(EST-GLR)*EF
      AE = DMAX1(EPS*AREA,TOL*DABS(GLR))
      IF (EE-AE) 40, 40, 50
   30 MXL = 1
   40 CE = CE + (EST-GLR)
      IF (LR(L)) 60, 60, 80
C
C     CONSIDER THE LEFT HALF OF THIS LEVEL
C
   50 IF (K.GT.KMX) LMX = KML
      IF (L.GE.LMX) GO TO 30
      L = L + 1
      EPS = EPS*0.5D0
      EF = EF/SQ2
      HH(L) = HH(L-1)*0.5D0
      LR(L) = -1
      AA(L) = AA(L-1)
      EST = GL
      GO TO 20
C
C     PROCEED TO RIGHT HALF AT THIS LEVEL
C
   60 VL(L) = GLR
   70 EST = GR(L-1)
      LR(L) = 1
      AA(L) = AA(L) + 4.0D0*HH(L)
      GO TO 20
C
C     RETURN ONE LEVEL
C
   80 VR = GLR
   90 IF (L.LE.1) GO TO 120
      L = L - 1
      EPS = EPS*2.0D0
      EF = EF*SQ2
      IF (LR(L)) 100, 100, 110
  100 VL(L) = VL(L+1) + VR
      GO TO 70
  110 VR = VL(L+1) + VR
      GO TO 90
C
C      EXIT
C
  120 ANS = VR
      IF ((MXL.EQ.0) .OR. (DABS(CE).LE.2.0D0*TOL*AREA)) GO TO 140
      IERR = 2
      CALL XERROR('DGAUS8- ANS IS PROBABLY INSUFFICIENTLY ACCURATE.',
     * 48, 3, 1)
      GO TO 140
  130 IERR = -1
      CALL XERROR('DGAUS8- THE FOLLOWING TEMPORARY DIAGNOSTIC WILL APPEA
     *R ONLY ONCE.  A AND B ARE TOO NEARLY EQUAL TO ALLOW NORMAL INTEGRA
     *TION.  ANS IS SET TO ZERO, AND IERR=-1.', 158, 1, -1)
  140 ICALL = 0
      IF (ERR.LT.0.0D0) ERR = CE
      RETURN
  150 CONTINUE
      CALL XERROR('DGAUS8- ABS(ERR) IS TOO SMALL.', 30, 2, 1)
      RETURN
      END
*DECK I1MACH
      INTEGER FUNCTION I1MACH(I)
C***BEGIN PROLOGUE  I1MACH
C***DATE WRITTEN   750101   (YYMMDD)
```

```
C***REVISION DATE  890213   (YYMMDD)
C***CATEGORY NO.  R1
C***KEYWORDS  LIBRARY=SLATEC,TYPE=INTEGER(I1MACH-I),MACHINE CONSTANTS
C***AUTHOR  FOX, P. A., (BELL LABS)
C           HALL, A. D., (BELL LABS)
C           SCHRYER, N. L., (BELL LABS)
C***PURPOSE  Returns integer machine dependent constants
C***DESCRIPTION
C
C     I1MACH can be used to obtain machine-dependent parameters
C     for the local machine environment.  It is a function
C     subroutine with one (input) argument, and can be called
C     as follows, for example
C
C          K = I1MACH(I)
C
C     where I=1,...,16.  The (output) value of K above is
C     determined by the (input) value of I.  The results for
C     various values of I are discussed below.
C
C  I/O unit numbers.
C    I1MACH( 1) = the standard input unit.
C    I1MACH( 2) = the standard output unit.
C    I1MACH( 3) = the standard punch unit.
C    I1MACH( 4) = the standard error message unit.
C
C  Words.
C    I1MACH( 5) = the number of bits per integer storage unit.
C    I1MACH( 6) = the number of characters per integer storage unit.
C
C  Integers.
C    assume integers are represented in the S-digit, base-A form
C
C               sign ( X(S-1)*A**(S-1) + ... + X(1)*A + X(0) )
C
C               where 0 .LE. X(I) .LT. A for I=0,...,S-1.
C    I1MACH( 7) = A, the base.
C    I1MACH( 8) = S, the number of base-A digits.
C    I1MACH( 9) = A**S - 1, the largest magnitude.
C
C  Floating-Point Numbers.
C    Assume floating-point numbers are represented in the T-digit,
C    base-B form
C               sign (B**E)*( (X(1)/B) + ... + (X(T)/B**T) )
C
C               where 0 .LE. X(I) .LT. B for I=1,...,T,
C               0 .LT. X(1), and EMIN .LE. E .LE. EMAX.
C    I1MACH(10) = B, the base.
C
C  Single-Precision
C    I1MACH(11) = T, the number of base-B digits.
C    I1MACH(12) = EMIN, the smallest exponent E.
C    I1MACH(13) = EMAX, the largest exponent E.
C
C  Double-Precision
C    I1MACH(14) = T, the number of base-B digits.
C    I1MACH(15) = EMIN, the smallest exponent E.
C    I1MACH(16) = EMAX, the largest exponent E.
C
C  To alter this function for a particular environment,
C  the desired set of DATA statements should be activated by
C  removing the C from column 1.  Also, the values of
C  I1MACH(1) - I1MACH(4) should be checked for consistency
C  with the local operating system.
C
C***REFERENCES  FOX P.A., HALL A.D., SCHRYER N.L.,*FRAMEWORK FOR A
C                 PORTABLE LIBRARY*, ACM TRANSACTIONS ON MATHEMATICAL
```

```
C                  SOFTWARE, VOL. 4, NO. 2, JUNE 1978, PP. 177-188.
C***ROUTINES CALLED  (NONE)
C***END PROLOGUE  I1MACH
C
      INTEGER IMACH(16),OUTPUT
      SAVE IMACH
      EQUIVALENCE (IMACH(4),OUTPUT)
C
C     MACHINE CONSTANTS FOR THE AMIGA
C     ABSOFT COMPILER
C
C     DATA IMACH(1) /    5 /
C     DATA IMACH(2) /    6 /
C     DATA IMACH(3) /    5 /
C     DATA IMACH(4) /    6 /
C     DATA IMACH(5) /   32 /
C     DATA IMACH(6) /    4 /
C     DATA IMACH(7) /    2 /
C     DATA IMACH(8) /   31 /
C     DATA IMACH(9) / 2147483647 /
C     DATA IMACH(10)/    2 /
C     DATA IMACH(11)/   24 /
C     DATA IMACH(12)/ -126 /
C     DATA IMACH(13)/  127 /
C     DATA IMACH(14)/   53 /
C     DATA IMACH(15)/ -1022 /
C     DATA IMACH(16)/  1023 /
C
C     MACHINE CONSTANTS FOR THE APOLLO
C
C     DATA IMACH(1) /    5 /
C     DATA IMACH(2) /    6 /
C     DATA IMACH(3) /    6 /
C     DATA IMACH(4) /    6 /
C     DATA IMACH(5) /   32 /
C     DATA IMACH(6) /    4 /
C     DATA IMACH(7) /    2 /
C     DATA IMACH(8) /   31 /
C     DATA IMACH(9) / 2147483647 /
C     DATA IMACH(10)/    2 /
C     DATA IMACH(11)/   24 /
C     DATA IMACH(12)/ -125 /
C     DATA IMACH(13)/  129 /
C     DATA IMACH(14)/   53 /
C     DATA IMACH(15)/ -1021 /
C     DATA IMACH(16)/  1025 /
C
C     MACHINE CONSTANTS FOR THE BURROUGHS 1700 SYSTEM
C
C     DATA IMACH( 1) /    7 /
C     DATA IMACH( 2) /    2 /
C     DATA IMACH( 3) /    2 /
C     DATA IMACH( 4) /    2 /
C     DATA IMACH( 5) /   36 /
C     DATA IMACH( 6) /    4 /
C     DATA IMACH( 7) /    2 /
C     DATA IMACH( 8) /   33 /
C     DATA IMACH( 9) / Z1FFFFFFFF /
C     DATA IMACH(10) /    2 /
C     DATA IMACH(11) /   24 /
C     DATA IMACH(12) / -256 /
C     DATA IMACH(13) /  255 /
C     DATA IMACH(14) /   60 /
C     DATA IMACH(15) / -256 /
C     DATA IMACH(16) /  255 /
C
C     MACHINE CONSTANTS FOR THE BURROUGHS 5700 SYSTEM
```

```
C
C     DATA IMACH( 1) /    5 /
C     DATA IMACH( 2) /    6 /
C     DATA IMACH( 3) /    7 /
C     DATA IMACH( 4) /    6 /
C     DATA IMACH( 5) /   48 /
C     DATA IMACH( 6) /    6 /
C     DATA IMACH( 7) /    2 /
C     DATA IMACH( 8) /   39 /
C     DATA IMACH( 9) / O0007777777777777 /
C     DATA IMACH(10) /    8 /
C     DATA IMACH(11) /   13 /
C     DATA IMACH(12) /  -50 /
C     DATA IMACH(13) /   76 /
C     DATA IMACH(14) /   26 /
C     DATA IMACH(15) /  -50 /
C     DATA IMACH(16) /   76 /
C
C     MACHINE CONSTANTS FOR THE BURROUGHS 6700/7700 SYSTEMS
C
C     DATA IMACH( 1) /    5 /
C     DATA IMACH( 2) /    6 /
C     DATA IMACH( 3) /    7 /
C     DATA IMACH( 4) /    6 /
C     DATA IMACH( 5) /   48 /
C     DATA IMACH( 6) /    6 /
C     DATA IMACH( 7) /    2 /
C     DATA IMACH( 8) /   39 /
C     DATA IMACH( 9) / O0007777777777777 /
C     DATA IMACH(10) /    8 /
C     DATA IMACH(11) /   13 /
C     DATA IMACH(12) /  -50 /
C     DATA IMACH(13) /   76 /
C     DATA IMACH(14) /   26 /
C     DATA IMACH(15) / -32754 /
C     DATA IMACH(16) /  32780 /
C
C     MACHINE CONSTANTS FOR THE CDC 170/180 SERIES USING NOS/VE
C
C     DATA IMACH( 1) /      5 /
C     DATA IMACH( 2) /      6 /
C     DATA IMACH( 3) /      7 /
C     DATA IMACH( 4) /      6 /
C     DATA IMACH( 5) /     64 /
C     DATA IMACH( 6) /      8 /
C     DATA IMACH( 7) /      2 /
C     DATA IMACH( 8) /     63 /
C     DATA IMACH( 9) / 9223372036854775807 /
C     DATA IMACH(10) /      2 /
C     DATA IMACH(11) /     47 /
C     DATA IMACH(12) /  -4095 /
C     DATA IMACH(13) /   4094 /
C     DATA IMACH(14) /     94 /
C     DATA IMACH(15) /  -4095 /
C     DATA IMACH(16) /   4094 /
C
C     MACHINE CONSTANTS FOR THE CDC 6000/7000 SERIES
C
C     DATA IMACH( 1) /     5 /
C     DATA IMACH( 2) /     6 /
C     DATA IMACH( 3) /     7 /
C     DATA IMACH( 4) /6LOUTPUT/
C     DATA IMACH( 5) /    60 /
C     DATA IMACH( 6) /    10 /
C     DATA IMACH( 7) /     2 /
C     DATA IMACH( 8) /    48 /
C     DATA IMACH( 9) / 00007777777777777777B /
```

```
C     DATA IMACH(10) /    2 /
C     DATA IMACH(11) /   47 /
C     DATA IMACH(12) / -929 /
C     DATA IMACH(13) / 1070 /
C     DATA IMACH(14) /   94 /
C     DATA IMACH(15) / -929 /
C     DATA IMACH(16) / 1069 /
C
C     MACHINE CONSTANTS FOR THE CELERITY C1260
C
C     DATA IMACH(1) /    5 /
C     DATA IMACH(2) /    6 /
C     DATA IMACH(3) /    6 /
C     DATA IMACH(4) /    0 /
C     DATA IMACH(5) /   32 /
C     DATA IMACH(6) /    4 /
C     DATA IMACH(7) /    2 /
C     DATA IMACH(8) /   31 /
C     DATA IMACH(9) / Z'7FFFFFFF' /
C     DATA IMACH(10)/    2 /
C     DATA IMACH(11)/   24 /
C     DATA IMACH(12)/ -126 /
C     DATA IMACH(13)/  127 /
C     DATA IMACH(14)/   53 /
C     DATA IMACH(15)/ -1022 /
C     DATA IMACH(16)/  1023 /
C
C     MACHINE CONSTANTS FOR THE CONVEX C-1
C
C     DATA IMACH( 1) /      5/
C     DATA IMACH( 2) /      6/
C     DATA IMACH( 3) /      7/
C     DATA IMACH( 4) /      6/
C     DATA IMACH( 5) /     32/
C     DATA IMACH( 6) /      4/
C     DATA IMACH( 7) /      2/
C     DATA IMACH( 8) /     31/
C     DATA IMACH( 9) /2147483647/
C     DATA IMACH(10) /      2/
C     DATA IMACH(11) /     24/
C     DATA IMACH(12) /   -128/
C     DATA IMACH(13) /    127/
C     DATA IMACH(14) /     53/
C     DATA IMACH(15) /  -1024/
C     DATA IMACH(16) /   1023/
C
C     MACHINE CONSTANTS FOR THE CRAY-1
C
C     DATA IMACH( 1) /   100 /
C     DATA IMACH( 2) /   101 /
C     DATA IMACH( 3) /   102 /
C     DATA IMACH( 4) /   101 /
C     DATA IMACH( 5) /    64 /
C     DATA IMACH( 6) /     8 /
C     DATA IMACH( 7) /     2 /
C     DATA IMACH( 8) /    63 /
C     DATA IMACH( 9) /  777777777777777777777B /
C     DATA IMACH(10) /     2 /
C     DATA IMACH(11) /    47 /
C     DATA IMACH(12) / -8189 /
C     DATA IMACH(13) /  8190 /
C     DATA IMACH(14) /    94 /
C     DATA IMACH(15) / -8099 /
C     DATA IMACH(16) /  8190 /
C
C     MACHINE CONSTANTS FOR THE DATA GENERAL ECLIPSE S/200
C
```

```
C     DATA IMACH( 1) /    11 /
C     DATA IMACH( 2) /    12 /
C     DATA IMACH( 3) /     8 /
C     DATA IMACH( 4) /    10 /
C     DATA IMACH( 5) /    16 /
C     DATA IMACH( 6) /     2 /
C     DATA IMACH( 7) /     2 /
C     DATA IMACH( 8) /    15 /
C     DATA IMACH( 9) /32767 /
C     DATA IMACH(10) /    16 /
C     DATA IMACH(11) /     6 /
C     DATA IMACH(12) /   -64 /
C     DATA IMACH(13) /    63 /
C     DATA IMACH(14) /    14 /
C     DATA IMACH(15) /   -64 /
C     DATA IMACH(16) /    63 /
C
C     MACHINE CONSTANTS FOR THE ELXSI 6400
C
C     DATA IMACH( 1) /      5/
C     DATA IMACH( 2) /      6/
C     DATA IMACH( 3) /      6/
C     DATA IMACH( 4) /      6/
C     DATA IMACH( 5) /     32/
C     DATA IMACH( 6) /      4/
C     DATA IMACH( 7) /      2/
C     DATA IMACH( 8) /     32/
C     DATA IMACH( 9) /2147483647/
C     DATA IMACH(10) /      2/
C     DATA IMACH(11) /     24/
C     DATA IMACH(12) /   -126/
C     DATA IMACH(13) /    127/
C     DATA IMACH(14) /     53/
C     DATA IMACH(15) /  -1022/
C     DATA IMACH(16) /   1023/
C
C     MACHINE CONSTANTS FOR THE HARRIS 220
C
C     DATA IMACH( 1) /       5 /
C     DATA IMACH( 2) /       6 /
C     DATA IMACH( 3) /       0 /
C     DATA IMACH( 4) /       6 /
C     DATA IMACH( 5) /      24 /
C     DATA IMACH( 6) /       3 /
C     DATA IMACH( 7) /       2 /
C     DATA IMACH( 8) /      23 /
C     DATA IMACH( 9) / 8388607 /
C     DATA IMACH(10) /       2 /
C     DATA IMACH(11) /      23 /
C     DATA IMACH(12) /    -127 /
C     DATA IMACH(13) /     127 /
C     DATA IMACH(14) /      38 /
C     DATA IMACH(15) /    -127 /
C     DATA IMACH(16) /     127 /
C
C     MACHINE CONSTANTS FOR THE HONEYWELL 600/6000 SERIES
C
C     DATA IMACH( 1) /     5 /
C     DATA IMACH( 2) /     6 /
C     DATA IMACH( 3) /    43 /
C     DATA IMACH( 4) /     6 /
C     DATA IMACH( 5) /    36 /
C     DATA IMACH( 6) /     6 /
C     DATA IMACH( 7) /     2 /
C     DATA IMACH( 8) /    35 /
C     DATA IMACH( 9) / O377777777777 /
C     DATA IMACH(10) /     2 /
```

```
C      DATA IMACH(11) /   27 /
C      DATA IMACH(12) / -127 /
C      DATA IMACH(13) /  127 /
C      DATA IMACH(14) /   63 /
C      DATA IMACH(15) / -127 /
C      DATA IMACH(16) /  127 /
C
C      MACHINE CONSTANTS FOR THE HP 2100
C      3 WORD DOUBLE PRECISION OPTION WITH FTN4
C
C      DATA IMACH(1) /      5/
C      DATA IMACH(2) /      6 /
C      DATA IMACH(3) /      4 /
C      DATA IMACH(4) /      1 /
C      DATA IMACH(5) /     16 /
C      DATA IMACH(6) /      2 /
C      DATA IMACH(7) /      2 /
C      DATA IMACH(8) /     15 /
C      DATA IMACH(9) /  32767 /
C      DATA IMACH(10)/      2 /
C      DATA IMACH(11)/     23 /
C      DATA IMACH(12)/   -128 /
C      DATA IMACH(13)/    127 /
C      DATA IMACH(14)/     39 /
C      DATA IMACH(15)/   -128 /
C      DATA IMACH(16)/    127 /
C
C      MACHINE CONSTANTS FOR THE HP 2100
C      4 WORD DOUBLE PRECISION OPTION WITH FTN4
C
C      DATA IMACH(1) /      5 /
C      DATA IMACH(2) /      6 /
C      DATA IMACH(3) /      4 /
C      DATA IMACH(4) /      1 /
C      DATA IMACH(5) /     16 /
C      DATA IMACH(6) /      2 /
C      DATA IMACH(7) /      2 /
C      DATA IMACH(8) /     15 /
C      DATA IMACH(9) /  32767 /
C      DATA IMACH(10)/      2 /
C      DATA IMACH(11)/     23 /
C      DATA IMACH(12)/   -128 /
C      DATA IMACH(13)/    127 /
C      DATA IMACH(14)/     55 /
C      DATA IMACH(15)/   -128 /
C      DATA IMACH(16)/    127 /
C
C      MACHINE CONSTANTS FOR THE HP 9000
C
C      DATA IMACH(1)  /    5 /
C      DATA IMACH(2)  /    6 /
C      DATA IMACH(3)  /    6 /
C      DATA IMACH(3)  /    7 /
C      DATA IMACH(5)  /   32 /
C      DATA IMACH(6)  /    4 /
C      DATA IMACH(7)  /    2 /
C      DATA IMACH(8)  /   32 /
C      DATA IMACH(9)  /2147483647 /
C      DATA IMACH(10) /    2 /
C      DATA IMACH(11) /   24 /
C      DATA IMACH(12) / -126 /
C      DATA IMACH(13) /  127 /
C      DATA IMACH(14) /   53 /
C      DATA IMACH(15) /-1015 /
C      DATA IMACH(16) / 1017 /
C
C      MACHINE CONSTANTS FOR THE IBM 360/370 SERIES,
```

```
C     THE XEROX SIGMA 5/7/9, THE SEL SYSTEMS 85/86, AND
C     THE PERKIN ELMER (INTERDATA) 7/32.
C
C     DATA IMACH( 1) /    5 /
C     DATA IMACH( 2) /    6 /
C     DATA IMACH( 3) /    7 /
C     DATA IMACH( 4) /    6 /
C     DATA IMACH( 5) /   32 /
C     DATA IMACH( 6) /    4 /
C     DATA IMACH( 7) /   16 /
C     DATA IMACH( 8) /   31 /
C     DATA IMACH( 9) / Z7FFFFFFF /
C     DATA IMACH(10) /   16 /
C     DATA IMACH(11) /    6 /
C     DATA IMACH(12) /  -64 /
C     DATA IMACH(13) /   63 /
C     DATA IMACH(14) /   14 /
C     DATA IMACH(15) /  -64 /
C     DATA IMACH(16) /   63 /
C
C     MACHINE CONSTANTS FOR THE IBM PC
C
      DATA IMACH( 1) /      5 /
      DATA IMACH( 2) /      6 /
      DATA IMACH( 3) /      0 /
      DATA IMACH( 4) /      0 /
      DATA IMACH( 5) /     32 /
      DATA IMACH( 6) /      4 /
      DATA IMACH( 7) /      2 /
      DATA IMACH( 8) /     31 /
      DATA IMACH( 9) / 2147483647 /
      DATA IMACH(10) /      2 /
      DATA IMACH(11) /     24 /
      DATA IMACH(12) /   -125 /
      DATA IMACH(13) /    127 /
      DATA IMACH(14) /     53 /
      DATA IMACH(15) /  -1021 /
      DATA IMACH(16) /   1023 /
C
C     MACHINE CONSTANTS FOR THE IBM RS 6000
C
C     DATA IMACH( 1) /          5 /
C     DATA IMACH( 2) /          6 /
C     DATA IMACH( 3) /          6 /
C     DATA IMACH( 4) /          0 /
C     DATA IMACH( 5) /         32 /
C     DATA IMACH( 6) /          4 /
C     DATA IMACH( 7) /          2 /
C     DATA IMACH( 8) /         31 /
C     DATA IMACH( 9) / 2147483647 /
C     DATA IMACH(10) /          2 /
C     DATA IMACH(11) /         24 /
C     DATA IMACH(12) /       -125 /
C     DATA IMACH(13) /        128 /
C     DATA IMACH(14) /         53 /
C     DATA IMACH(15) /      -1021 /
C     DATA IMACH(16) /       1024 /
C
C     MACHINE CONSTANTS FOR THE PDP-10 (KA PROCESSOR)
C
C     DATA IMACH( 1) /    5 /
C     DATA IMACH( 2) /    6 /
C     DATA IMACH( 3) /    5 /
C     DATA IMACH( 4) /    6 /
C     DATA IMACH( 5) /   36 /
C     DATA IMACH( 6) /    5 /
C     DATA IMACH( 7) /    2 /
```

```
C     DATA IMACH( 8) /   35 /
C     DATA IMACH( 9) / "377777777777 /
C     DATA IMACH(10) /    2 /
C     DATA IMACH(11) /   27 /
C     DATA IMACH(12) / -128 /
C     DATA IMACH(13) /  127 /
C     DATA IMACH(14) /   54 /
C     DATA IMACH(15) / -101 /
C     DATA IMACH(16) /  127 /
C
C     MACHINE CONSTANTS FOR THE PDP-10 (KI PROCESSOR)
C
C     DATA IMACH( 1) /    5 /
C     DATA IMACH( 2) /    6 /
C     DATA IMACH( 3) /    5 /
C     DATA IMACH( 4) /    6 /
C     DATA IMACH( 5) /   36 /
C     DATA IMACH( 6) /    5 /
C     DATA IMACH( 7) /    2 /
C     DATA IMACH( 8) /   35 /
C     DATA IMACH( 9) / "377777777777 /
C     DATA IMACH(10) /    2 /
C     DATA IMACH(11) /   27 /
C     DATA IMACH(12) / -128 /
C     DATA IMACH(13) /  127 /
C     DATA IMACH(14) /   62 /
C     DATA IMACH(15) / -128 /
C     DATA IMACH(16) /  127 /
C
C     MACHINE CONSTANTS FOR PDP-11 FORTRAN SUPPORTING
C     32-BIT INTEGER ARITHMETIC.
C
C     DATA IMACH( 1) /    5 /
C     DATA IMACH( 2) /    6 /
C     DATA IMACH( 3) /    5 /
C     DATA IMACH( 4) /    6 /
C     DATA IMACH( 5) /   32 /
C     DATA IMACH( 6) /    4 /
C     DATA IMACH( 7) /    2 /
C     DATA IMACH( 8) /   31 /
C     DATA IMACH( 9) / 2147483647 /
C     DATA IMACH(10) /    2 /
C     DATA IMACH(11) /   24 /
C     DATA IMACH(12) / -127 /
C     DATA IMACH(13) /  127 /
C     DATA IMACH(14) /   56 /
C     DATA IMACH(15) / -127 /
C     DATA IMACH(16) /  127 /
C
C     MACHINE CONSTANTS FOR PDP-11 FORTRAN SUPPORTING
C     16-BIT INTEGER ARITHMETIC.
C
C     DATA IMACH( 1) /    5 /
C     DATA IMACH( 2) /    6 /
C     DATA IMACH( 3) /    5 /
C     DATA IMACH( 4) /    6 /
C     DATA IMACH( 5) /   16 /
C     DATA IMACH( 6) /    2 /
C     DATA IMACH( 7) /    2 /
C     DATA IMACH( 8) /   15 /
C     DATA IMACH( 9) / 32767 /
C     DATA IMACH(10) /    2 /
C     DATA IMACH(11) /   24 /
C     DATA IMACH(12) / -127 /
C     DATA IMACH(13) /  127 /
C     DATA IMACH(14) /   56 /
C     DATA IMACH(15) / -127 /
```

```
C     DATA IMACH(16) /  127 /
C
C     MACHINE CONSTANTS FOR THE SILICON GRAPHICS IRIS
C
C     DATA IMACH( 1) /     5 /
C     DATA IMACH( 2) /     6 /
C     DATA IMACH( 3) /     6 /
C     DATA IMACH( 4) /     0 /
C     DATA IMACH( 5) /    32 /
C     DATA IMACH( 6) /     4 /
C     DATA IMACH( 7) /     2 /
C     DATA IMACH( 8) /    31 /
C     DATA IMACH( 9) / 2147483647 /
C     DATA IMACH(10) /     2 /
C     DATA IMACH(11) /    23 /
C     DATA IMACH(12) /  -126 /
C     DATA IMACH(13) /   127 /
C     DATA IMACH(14) /    52 /
C     DATA IMACH(15) / -1022 /
C     DATA IMACH(16) /  1023 /
C
C     MACHINE CONSTANTS FOR THE SUN
C
C     DATA IMACH(1) /    5 /
C     DATA IMACH(2) /    6 /
C     DATA IMACH(3) /    6 /
C     DATA IMACH(4) /    6 /
C     DATA IMACH(5) /   32 /
C     DATA IMACH(6) /    4 /
C     DATA IMACH(7) /    2 /
C     DATA IMACH(8) /   31 /
C     DATA IMACH(9) /2147483647 /
C     DATA IMACH(10)/    2 /
C     DATA IMACH(11)/   24 /
C     DATA IMACH(12)/ -125 /
C     DATA IMACH(13)/  128 /
C     DATA IMACH(14)/   53 /
C     DATA IMACH(15)/ -1021 /
C     DATA IMACH(16)/  1024 /
C
C     MACHINE CONSTANTS FOR THE UNIVAC 1100 SERIES FTN COMPILER
C
C
C     DATA IMACH( 1) /    5 /
C     DATA IMACH( 2) /    6 /
C     DATA IMACH( 3) /    1 /
C     DATA IMACH( 4) /    6 /
C     DATA IMACH( 5) /   36 /
C     DATA IMACH( 6) /    4 /
C     DATA IMACH( 7) /    2 /
C     DATA IMACH( 8) /   35 /
C     DATA IMACH( 9) / O377777777777 /
C     DATA IMACH(10) /    2 /
C     DATA IMACH(11) /   27 /
C     DATA IMACH(12) / -128 /
C     DATA IMACH(13) /  127 /
C     DATA IMACH(14) /   60 /
C     DATA IMACH(15) /-1024 /
C     DATA IMACH(16) / 1023 /
C
C     MACHINE CONSTANTS FOR THE VAX 11/780
C
C     DATA IMACH(1) /    5 /
C     DATA IMACH(2) /    6 /
C     DATA IMACH(3) /    5 /
C     DATA IMACH(4) /    6 /
C     DATA IMACH(5) /   32 /
```

```
C     DATA IMACH(6) /    4 /
C     DATA IMACH(7) /    2 /
C     DATA IMACH(8) /   31 /
C     DATA IMACH(9) /2147483647 /
C     DATA IMACH(10)/    2 /
C     DATA IMACH(11)/   24 /
C     DATA IMACH(12)/ -127 /
C     DATA IMACH(13)/  127 /
C     DATA IMACH(14)/   56 /
C     DATA IMACH(15)/ -127 /
C     DATA IMACH(16)/  127 /
C
C     MACHINE CONSTANTS FOR THE VAX 11/780, G-FLOAT OPTION
C
C     DATA IMACH(1) /    5 /
C     DATA IMACH(2) /    6 /
C     DATA IMACH(3) /    5 /
C     DATA IMACH(4) /    6 /
C     DATA IMACH(5) /   32 /
C     DATA IMACH(6) /    4 /
C     DATA IMACH(7) /    2 /
C     DATA IMACH(8) /   31 /
C     DATA IMACH(9) /2147483647 /
C     DATA IMACH(10)/    2 /
C     DATA IMACH(11)/   24 /
C     DATA IMACH(12)/ -127 /
C     DATA IMACH(13)/  127 /
C     DATA IMACH(14)/   53 /
C     DATA IMACH(15)/ -1022 /
C     DATA IMACH(16)/  1023 /
C
C     MACHINE CONSTANTS FOR THE Z80 MICROPROCESSOR
C
C     DATA IMACH( 1) /      1/
C     DATA IMACH( 2) /      1/
C     DATA IMACH( 3) /      0/
C     DATA IMACH( 4) /      1/
C     DATA IMACH( 5) /     16/
C     DATA IMACH( 6) /      2/
C     DATA IMACH( 7) /      2/
C     DATA IMACH( 8) /     15/
C     DATA IMACH( 9) / 32767/
C     DATA IMACH(10) /      2/
C     DATA IMACH(11) /     24/
C     DATA IMACH(12) /   -127/
C     DATA IMACH(13) /    127/
C     DATA IMACH(14) /     56/
C     DATA IMACH(15) /   -127/
C     DATA IMACH(16) /    127/
C
C
C***FIRST EXECUTABLE STATEMENT  I1MACH
      IF (I .LT. 1  .OR.  I .GT. 16) GO TO 10
C
      I1MACH = IMACH(I)
      RETURN
C
   10 CONTINUE
      WRITE (UNIT = OUTPUT, FMT = 9000)
 9000 FORMAT ('1ERROR    1 IN I1MACH - I OUT OF BOUNDS')
C
C     CALL FDUMP
C
C
      STOP
      END
*DECK R1MACH
```

```
      REAL FUNCTION R1MACH(I)
C***BEGIN PROLOGUE  R1MACH
C***DATE WRITTEN   790101   (YYMMDD)
C***REVISION DATE  890213   (YYMMDD)
C***CATEGORY NO.  R1
C***KEYWORDS  LIBRARY=SLATEC,TYPE=SINGLE PRECISION(R1MACH-S D1MACH-D),
C             MACHINE CONSTANTS
C***AUTHOR  FOX, P. A., (BELL LABS)
C           HALL, A. D., (BELL LABS)
C           SCHRYER, N. L., (BELL LABS)
C***PURPOSE  Returns single precision machine dependent constants
C***DESCRIPTION
C
C   R1MACH can be used to obtain machine-dependent parameters
C   for the local machine environment.  It is a function
C   subroutine with one (input) argument, and can be called
C   as follows, for example
C
C        A = R1MACH(I)
C
C   where I=1,...,5.  The (output) value of A above is
C   determined by the (input) value of I.  The results for
C   various values of I are discussed below.
C
C   Single-Precision Machine Constants
C   R1MACH(1) = B**(EMIN-1), the smallest positive magnitude.
C   R1MACH(2) = B**EMAX*(1 - B**(-T)), the largest magnitude.
C   R1MACH(3) = B**(-T), the smallest relative spacing.
C   R1MACH(4) = B**(1-T), the largest relative spacing.
C   R1MACH(5) = LOG10(B)
C
C   Assume single precision numbers are represented in the T-digit,
C   base-B form
C
C              sign (B**E)*( (X(1)/B) + ... + (X(T)/B**T) )
C
C   where 0 .LE. X(I) .LT. B for I=1,...,T, 0 .LT. X(1), and
C   EMIN .LE. E .LE. EMAX.
C
C   The values of B, T, EMIN and EMAX are provided in I1MACH as
C   follows:
C   I1MACH(10) = B, the base.
C   I1MACH(11) = T, the number of base-B digits.
C   I1MACH(12) = EMIN, the smallest exponent E.
C   I1MACH(13) = EMAX, the largest exponent E.
C
C   To alter this function for a particular environment,
C   the desired set of DATA statements should be activated by
C   removing the C from column 1.  Also, the values of
C   R1MACH(1) - R1MACH(4) should be checked for consistency
C   with the local operating system.
C
C***REFERENCES  FOX, P.A., HALL, A.D., SCHRYER, N.L, *FRAMEWORK FOR
C                 A PORTABLE LIBRARY*, ACM TRANSACTIONS ON MATHE-
C                 MATICAL SOFTWARE, VOL. 4, NO. 2, JUNE 1978,
C                 PP. 177-188.
C***ROUTINES CALLED  XERROR
C***END PROLOGUE  R1MACH
C
      INTEGER SMALL(2)
      INTEGER LARGE(2)
      INTEGER RIGHT(2)
      INTEGER DIVER(2)
      INTEGER LOG10(2)
C
      REAL RMACH(5)
      SAVE RMACH
```

```
C
      EQUIVALENCE (RMACH(1),SMALL(1))
      EQUIVALENCE (RMACH(2),LARGE(1))
      EQUIVALENCE (RMACH(3),RIGHT(1))
      EQUIVALENCE (RMACH(4),DIVER(1))
      EQUIVALENCE (RMACH(5),LOG10(1))
C
C     MACHINE CONSTANTS FOR THE AMIGA
C     ABSOFT FORTRAN COMPILER USING THE 68020/68881 COMPILER OPTION
C
C     DATA SMALL(1) / Z'00800000' /
C     DATA LARGE(1) / Z'7F7FFFFF' /
C     DATA RIGHT(1) / Z'33800000' /
C     DATA DIVER(1) / Z'34000000' /
C     DATA LOG10(1) / Z'3E9A209B' /
C
C     MACHINE CONSTANTS FOR THE AMIGA
C     ABSOFT FORTRAN COMPILER USING SOFTWARE FLOATING POINT
C
C     DATA SMALL(1) / Z'00800000' /
C     DATA LARGE(1) / Z'7EFFFFFF' /
C     DATA RIGHT(1) / Z'33800000' /
C     DATA DIVER(1) / Z'34000000' /
C     DATA LOG10(1) / Z'3E9A209B' /
C
C     MACHINE CONSTANTS FOR THE APOLLO
C
C     DATA SMALL(1) / 16#00800000 /
C     DATA LARGE(1) / 16#7FFFFFFF /
C     DATA RIGHT(1) / 16#33800000 /
C     DATA DIVER(1) / 16#34000000 /
C     DATA LOG10(1) / 16#3E9A209B /
C
C     MACHINE CONSTANTS FOR THE BURROUGHS 1700 SYSTEM
C
C     DATA RMACH(1) / Z400800000 /
C     DATA RMACH(2) / Z5FFFFFFFF /
C     DATA RMACH(3) / Z4E9800000 /
C     DATA RMACH(4) / Z4EA800000 /
C     DATA RMACH(5) / Z500E730E8 /
C
C     MACHINE CONSTANTS FOR THE BURROUGHS 5700/6700/7700 SYSTEMS
C
C     DATA RMACH(1) / O1771000000000000 /
C     DATA RMACH(2) / O0777777777777777 /
C     DATA RMACH(3) / O1311000000000000 /
C     DATA RMACH(4) / O1301000000000000 /
C     DATA RMACH(5) / O1157163034761675 /
C
C     MACHINE CONSTANTS FOR THE CDC 170/180 SERIES USING NOS/VE
C
C     DATA RMACH(1) / Z"3001800000000000" /
C     DATA RMACH(2) / Z"4FFEFFFFFFFFFFFE" /
C     DATA RMACH(3) / Z"3FD2800000000000" /
C     DATA RMACH(4) / Z"3FD3800000000000" /
C     DATA RMACH(5) / Z"3FFF9A209A84FBCF" /
C
C     MACHINE CONSTANTS FOR THE CDC 6000/7000 SERIES
C
C     DATA RMACH(1) / 00564000000000000000B /
C     DATA RMACH(2) / 37767777777777777776B /
C     DATA RMACH(3) / 16414000000000000000B /
C     DATA RMACH(4) / 16424000000000000000B /
C     DATA RMACH(5) / 17164642023241175720B /
C
C     MACHINE CONSTANTS FOR THE CELERITY C1260
C
```

```
C     DATA SMALL(1) / Z'00800000' /
C     DATA LARGE(1) / Z'7F7FFFFF' /
C     DATA RIGHT(1) / Z'33800000' /
C     DATA DIVER(1) / Z'34000000' /
C     DATA LOG10(1) / Z'3E9A209B' /
C
C     MACHINE CONSTANTS FOR THE CONVEX C-1
C
C     DATA SMALL(1) / '00800000'X /
C     DATA LARGE(1) / '7FFFFFFF'X /
C     DATA RIGHT(1) / '34800000'X /
C     DATA DIVER(1) / '35000000'X /
C     DATA LOG10(1) / '3F9A209B'X /
C
C     MACHINE CONSTANTS FOR THE CRAY-1
C
C     DATA RMACH(1) / 200034000000000000000B /
C     DATA RMACH(2) / 577767777777777777776B /
C     DATA RMACH(3) / 377224000000000000000B /
C     DATA RMACH(4) / 377234000000000000000B /
C     DATA RMACH(5) / 377774642023241175720B /
C
C     MACHINE CONSTANTS FOR THE DATA GENERAL ECLIPSE S/200
C
C     NOTE - IT MAY BE APPROPRIATE TO INCLUDE THE FOLLOWING CARD -
C     STATIC RMACH(5)
C
C     DATA SMALL /    20K,       0 /
C     DATA LARGE / 77777K, 177777K /
C     DATA RIGHT / 35420K,       0 /
C     DATA DIVER / 36020K,       0 /
C     DATA LOG10 / 40423K,  42023K /
C
C     MACHINE CONSTANTS FOR THE HARRIS 220
C
C     DATA SMALL(1), SMALL(2) / '20000000, '00000201 /
C     DATA LARGE(1), LARGE(2) / '37777777, '00000177 /
C     DATA RIGHT(1), RIGHT(2) / '20000000, '00000352 /
C     DATA DIVER(1), DIVER(2) / '20000000, '00000353 /
C     DATA LOG10(1), LOG10(2) / '23210115, '00000377 /
C
C     MACHINE CONSTANTS FOR THE HONEYWELL 600/6000 SERIES
C
C     DATA RMACH(1) / O402400000000 /
C     DATA RMACH(2) / O376777777777 /
C     DATA RMACH(3) / O714400000000 /
C     DATA RMACH(4) / O716400000000 /
C     DATA RMACH(5) / O776464202324 /
C
C     MACHINE CONSTANTS FOR THE HP 2100
C     3 WORD DOUBLE PRECISION WITH FTN4
C
C     DATA SMALL(1), SMALL(2) / 40000B,       1 /
C     DATA LARGE(1), LARGE(2) / 77777B, 177776B /
C     DATA RIGHT(1), RIGHT(2) / 40000B,    325B /
C     DATA DIVER(1), DIVER(2) / 40000B,    327B /
C     DATA LOG10(1), LOG10(2) / 46420B,  46777B /
C
C     MACHINE CONSTANTS FOR THE HP 2100
C     4 WORD DOUBLE PRECISION WITH FTN4
C
C     DATA SMALL(1), SMALL(2) / 40000B,       1 /
C     DATA LARGE91), LARGE(2) / 77777B, 177776B /
C     DATA RIGHT(1), RIGHT(2) / 40000B,    325B /
C     DATA DIVER(1), DIVER(2) / 40000B,    327B /
C     DATA LOG10(1), LOG10(2) / 46420B,  46777B /
C
```

```
C     MACHINE CONSTANTS FOR THE HP 9000
C
C     DATA SMALL(1) / 00004000000B /
C     DATA LARGE(1) / 17677777777B /
C     DATA RIGHT(1) / 06340000000B /
C     DATA DIVER(1) / 06400000000B /
C     DATA LOG10(1) / 07646420233B /
C
C     MACHINE CONSTANTS FOR THE ELXSI 6400
C       ASSUMING REAL*4 IS THE DEFAULT REAL
C
C     DATA SMALL(1) / '00800000'X /
C     DATA LARGE(1) / '7F7FFFFF'X /
C     DATA RIGHT(1) / '33800000'X /
C     DATA DIVER(1) / '34000000'X /
C     DATA LOG10(1) / '3E9A209B'X /
C
C     MACHINE CONSTANTS FOR THE IBM 360/370 SERIES,
C     THE XEROX SIGMA 5/7/9, THE SEL SYSTEMS 85/86  AND
C     THE PERKIN ELMER (INTERDATA) 7/32.
C
C     DATA RMACH(1) / Z00100000 /
C     DATA RMACH(2) / Z7FFFFFFF /
C     DATA RMACH(3) / Z3B100000 /
C     DATA RMACH(4) / Z3C100000 /
C     DATA RMACH(5) / Z41134413 /
C
C     MACHINE CONSTANTS FOR THE IBM PC
C
      DATA SMALL(1) /     8420761 /
      DATA LARGE(1) /  2139081118 /
      DATA RIGHT(1) /   863997169 /
      DATA DIVER(1) /   872385777 /
      DATA LOG10(1) /  1050288283 /
C
C     MACHINE CONSTANTS FOR THE IBM RS 6000
C
C     DATA SMALL(1) / Z'00800000' /
C     DATA LARGE(1) / Z'7F7FFFFF' /
C     DATA RIGHT(1) / Z'33800000' /
C     DATA DIVER(1) / Z'34000000' /
C     DATA LOG10(1) / Z'3E9A209B' /
C
C     MACHINE CONSTANTS FOR THE PDP-10 (KA OR KI PROCESSOR)
C
C     DATA RMACH(1) / "000400000000 /
C     DATA RMACH(2) / "377777777777 /
C     DATA RMACH(3) / "146400000000 /
C     DATA RMACH(4) / "147400000000 /
C     DATA RMACH(5) / "177464202324 /
C
C     MACHINE CONSTANTS FOR PDP-11 FORTRAN SUPPORTING
C     32-BIT INTEGERS (EXPRESSED IN INTEGER AND OCTAL).
C
C     DATA SMALL(1) /    8388608 /
C     DATA LARGE(1) / 2147483647 /
C     DATA RIGHT(1) /  880803840 /
C     DATA DIVER(1) /  889192448 /
C     DATA LOG10(1) / 1067065499 /
C
C     DATA RMACH(1) / O00040000000 /
C     DATA RMACH(2) / O17777777777 /
C     DATA RMACH(3) / O06440000000 /
C     DATA RMACH(4) / O06500000000 /
C     DATA RMACH(5) / O07746420233 /
C
C     MACHINE CONSTANTS FOR PDP-11 FORTRAN SUPPORTING
```

```
C     16-BIT INTEGERS  (EXPRESSED IN INTEGER AND OCTAL).
C
C     DATA SMALL(1), SMALL(2) /   128,     0 /
C     DATA LARGE(1), LARGE(2) / 32767,    -1 /
C     DATA RIGHT(1), RIGHT(2) / 13440,     0 /
C     DATA DIVER(1), DIVER(2) / 13568,     0 /
C     DATA LOG10(1), LOG10(2) / 16282,  8347 /
C
C     DATA SMALL(1), SMALL(2) / O000200, O000000 /
C     DATA LARGE(1), LARGE(2) / O077777, O177777 /
C     DATA RIGHT(1), RIGHT(2) / O032200, O000000 /
C     DATA DIVER(1), DIVER(2) / O032400, O000000 /
C     DATA LOG10(1), LOG10(2) / O037632, O020233 /
C
C     MACHINE CONSTANTS FOR THE SILICON GRAPHICS IRIS
C
c     data rmach(1) / 1.17549 424 e-38 /
c     data rmach(2) / 3.40282 356 e+38 /
c     data rmach(3) / 1.19209 290 e-07 /
c     data rmach(4) / 2.38418 579 e-07 /
c     data rmach(5) / 0.30103 001 /
C
C     DATA SMALL(1) / Z'00800000' /
C     DATA LARGE(1) / Z'7F7FFFFF' /
C     DATA RIGHT(1) / Z'34000000' /
C     DATA DIVER(1) / Z'34800000' /
C     DATA LOG10(1) / Z'3E9A209B' /
C
C     MACHINE CONSTANTS FOR THE SUN
C
C     DATA SMALL(1) / Z'00800000' /
C     DATA LARGE(1) / Z'7F7FFFFF' /
C     DATA RIGHT(1) / Z'33800000' /
C     DATA DIVER(1) / Z'34000000' /
C     DATA LOG10(1) / Z'3E9A209B' /
C
C     MACHINE CONSTANTS FOR THE UNIVAC 1100 SERIES
C
C     DATA RMACH(1) / O000400000000 /
C     DATA RMACH(2) / O377777777777 /
C     DATA RMACH(3) / O146400000000 /
C     DATA RMACH(4) / O147400000000 /
C     DATA RMACH(5) / O177464202324 /
C
C     MACHINE CONSTANTS FOR THE VAX 11/780
C     (EXPRESSED IN INTEGER AND HEXADECIMAL)
C     THE HEX FORMAT BELOW MAY NOT BE SUITABLE FOR UNIX SYSTEMS
C     THE INTEGER FORMAT SHOULD BE OK FOR UNIX SYSTEMS
C
C     DATA SMALL(1) /       128 /
C     DATA LARGE(1) /    -32769 /
C     DATA RIGHT(1) /     13440 /
C     DATA DIVER(1) /     13568 /
C     DATA LOG10(1) / 547045274 /
C
C     DATA SMALL(1) / Z00000080 /
C     DATA LARGE(1) / ZFFFF7FFF /
C     DATA RIGHT(1) / Z00003480 /
C     DATA DIVER(1) / Z00003500 /
C     DATA LOG10(1) / Z209B3F9A /
C
C     MACHINE CONSTANTS FOR THE Z80 MICROPROCESSOR
C
C     DATA SMALL(1), SMALL(2) /     0,    256/
C     DATA LARGE(1), LARGE(2) /    -1,   -129/
C     DATA RIGHT(1), RIGHT(2) /     0,  26880/
C     DATA DIVER(1), DIVER(2) /     0,  27136/
```

```
C     DATA LOG10(1), LOG10(2) /  8347,  32538/
C
C
C***FIRST EXECUTABLE STATEMENT  R1MACH
      IF (I .LT. 1  .OR.  I .GT. 5)
     1   CALL XERROR ('R1MACH -- I OUT OF BOUNDS', 25, 1, 2)
C
      R1MACH = RMACH(I)
      RETURN
C
      END
*DECK D1MACH
      DOUBLE PRECISION FUNCTION D1MACH(I)
C***BEGIN PROLOGUE  D1MACH
C***DATE WRITTEN   750101   (YYMMDD)
C***REVISION DATE  890213   (YYMMDD)
C***CATEGORY NO.  R1
C***KEYWORDS  LIBRARY=SLATEC,TYPE=DOUBLE PRECISION(R1MACH-S D1MACH-D),
C             MACHINE CONSTANTS
C***AUTHOR  FOX, P. A., (BELL LABS)
C           HALL, A. D., (BELL LABS)
C           SCHRYER, N. L., (BELL LABS)
C***PURPOSE  Returns double precision machine dependent constants
C***DESCRIPTION
C
C   D1MACH can be used to obtain machine-dependent parameters
C   for the local machine environment.  It is a function
C   subprogram with one (input) argument, and can be called
C   as follows, for example
C
C        D = D1MACH(I)
C
C   where I=1,...,5.  The (output) value of D above is
C   determined by the (input) value of I.  The results for
C   various values of I are discussed below.
C
C   D1MACH( 1) = B**(EMIN-1), the smallest positive magnitude.
C   D1MACH( 2) = B**EMAX*(1 - B**(-T)), the largest magnitude.
C   D1MACH( 3) = B**(-T), the smallest relative spacing.
C   D1MACH( 4) = B**(1-T), the largest relative spacing.
C   D1MACH( 5) = LOG10(B)
C
C   Assume double precision numbers are represented in the T-digit,
C   base-B form
C
C              sign (B**E)*( (X(1)/B) + ... + (X(T)/B**T) )
C
C   where 0 .LE. X(I) .LT. B for I=1,...,T, 0 .LT. X(1), and
C   EMIN .LE. E .LE. EMAX.
C
C   The values of B, T, EMIN and EMAX are provided in I1MACH as
C   follows:
C   I1MACH(10) = B, the base.
C   I1MACH(14) = T, the number of base-B digits.
C   I1MACH(15) = EMIN, the smallest exponent E.
C   I1MACH(16) = EMAX, the largest exponent E.
C
C   To alter this function for a particular environment,
C   the desired set of DATA statements should be activated by
C   removing the C from column 1.  Also, the values of
C   D1MACH(1) - D1MACH(4) should be checked for consistency
C   with the local operating system.
C
C***REFERENCES  FOX P.A., HALL A.D., SCHRYER N.L.,*FRAMEWORK FOR A
C                 PORTABLE LIBRARY*, ACM TRANSACTIONS ON MATHEMATICAL
C                 SOFTWARE, VOL. 4, NO. 2, JUNE 1978, PP. 177-188.
C***ROUTINES CALLED  XERROR
```

```
C***END PROLOGUE  D1MACH
C
      INTEGER SMALL(4)
      INTEGER LARGE(4)
      INTEGER RIGHT(4)
      INTEGER DIVER(4)
      INTEGER LOG10(4)
C
      DOUBLE PRECISION DMACH(5)
      SAVE DMACH
C
      EQUIVALENCE (DMACH(1),SMALL(1))
      EQUIVALENCE (DMACH(2),LARGE(1))
      EQUIVALENCE (DMACH(3),RIGHT(1))
      EQUIVALENCE (DMACH(4),DIVER(1))
      EQUIVALENCE (DMACH(5),LOG10(1))
C
C     MACHINE CONSTANTS FOR THE AMIGA
C     ABSOFT FORTRAN COMPILER USING THE 68020/68881 COMPILER OPTION
C
C     DATA SMALL(1), SMALL(2) / Z'00100000', Z'00000000' /
C     DATA LARGE(1), LARGE(2) / Z'7FEFFFFF', Z'FFFFFFFF' /
C     DATA RIGHT(1), RIGHT(2) / Z'3CA00000', Z'00000000' /
C     DATA DIVER(1), DIVER(2) / Z'3CB00000', Z'00000000' /
C     DATA LOG10(1), LOG10(2) / Z'3FD34413', Z'509F79FF' /
C
C     MACHINE CONSTANTS FOR THE AMIGA
C     ABSOFT FORTRAN COMPILER USING SOFTWARE FLOATING POINT
C
C     DATA SMALL(1), SMALL(2) / Z'00100000', Z'00000000' /
C     DATA LARGE(1), LARGE(2) / Z'7FDFFFFF', Z'FFFFFFFF' /
C     DATA RIGHT(1), RIGHT(2) / Z'3CA00000', Z'00000000' /
C     DATA DIVER(1), DIVER(2) / Z'3CB00000', Z'00000000' /
C     DATA LOG10(1), LOG10(2) / Z'3FD34413', Z'509F79FF' /
C
C     MACHINE CONSTANTS FOR THE APOLLO
C
C     DATA SMALL(1), SMALL(2) / 16#00100000, 16#00000000 /
C     DATA LARGE(1), LARGE(2) / 16#7FFFFFFF, 16#FFFFFFFF /
C     DATA RIGHT(1), RIGHT(2) / 16#3CA00000, 16#00000000 /
C     DATA DIVER(1), DIVER(2) / 16#3CB00000, 16#00000000 /
C     DATA LOG10(1), LOG10(2) / 16#3FD34413, 16#509F79FF /
C
C     MACHINE CONSTANTS FOR THE BURROUGHS 1700 SYSTEM
C
C     DATA SMALL(1) / ZC00800000 /
C     DATA SMALL(2) / Z000000000 /
C     DATA LARGE(1) / ZDFFFFFFFF /
C     DATA LARGE(2) / ZFFFFFFFFF /
C     DATA RIGHT(1) / ZCC5800000 /
C     DATA RIGHT(2) / Z000000000 /
C     DATA DIVER(1) / ZCC6800000 /
C     DATA DIVER(2) / Z000000000 /
C     DATA LOG10(1) / ZD00E730E7 /
C     DATA LOG10(2) / ZC77800DC0 /
C
C     MACHINE CONSTANTS FOR THE BURROUGHS 5700 SYSTEM
C
C     DATA SMALL(1) / O1771000000000000 /
C     DATA SMALL(2) / O0000000000000000 /
C     DATA LARGE(1) / O0777777777777777 /
C     DATA LARGE(2) / O0007777777777777 /
C     DATA RIGHT(1) / O1461000000000000 /
C     DATA RIGHT(2) / O0000000000000000 /
C     DATA DIVER(1) / O1451000000000000 /
C     DATA DIVER(2) / O0000000000000000 /
C     DATA LOG10(1) / O1157163034761674 /
```

```
C     DATA LOG10(2) / O0006677466732724 /
C
C     MACHINE CONSTANTS FOR THE BURROUGHS 6700/7700 SYSTEMS
C
C     DATA SMALL(1) / O1771000000000000 /
C     DATA SMALL(2) / O7770000000000000 /
C     DATA LARGE(1) / O0777777777777777 /
C     DATA LARGE(2) / O7777777777777777 /
C     DATA RIGHT(1) / O1461000000000000 /
C     DATA RIGHT(2) / O0000000000000000 /
C     DATA DIVER(1) / O1451000000000000 /
C     DATA DIVER(2) / O0000000000000000 /
C     DATA LOG10(1) / O1157163034761674 /
C     DATA LOG10(2) / O0006677466732724 /
C
C     MACHINE CONSTANTS FOR THE CDC 170/180 SERIES USING NOS/VE
C
C     DATA SMALL(1) / Z"3001800000000000" /
C     DATA SMALL(2) / Z"3001000000000000" /
C     DATA LARGE(1) / Z"4FFEFFFFFFFFFFFE" /
C     DATA LARGE(2) / Z"4FFE000000000000" /
C     DATA RIGHT(1) / Z"3FD2800000000000" /
C     DATA RIGHT(2) / Z"3FD2000000000000" /
C     DATA DIVER(1) / Z"3FD3800000000000" /
C     DATA DIVER(2) / Z"3FD3000000000000" /
C     DATA LOG10(1) / Z"3FFF9A209A84FBCF" /
C     DATA LOG10(2) / Z"3FFFF7988F8959AC" /
C
C     MACHINE CONSTANTS FOR THE CDC 6000/7000 SERIES
C
C     DATA SMALL(1) / 00564000000000000000B /
C     DATA SMALL(2) / 00000000000000000000B /
C     DATA LARGE(1) / 37757777777777777777B /
C     DATA LARGE(2) / 37157777777777777777B /
C     DATA RIGHT(1) / 15624000000000000000B /
C     DATA RIGHT(2) / 00000000000000000000B /
C     DATA DIVER(1) / 15634000000000000000B /
C     DATA DIVER(2) / 00000000000000000000B /
C     DATA LOG10(1) / 17164642023241175717B /
C     DATA LOG10(2) / 16367571421742254654B /
C
C     MACHINE CONSTANTS FOR THE CELERITY C1260
C
C     DATA SMALL(1), SMALL(2) / Z'00100000', Z'00000000' /
C     DATA LARGE(1), LARGE(2) / Z'7FEFFFFF', Z'FFFFFFFF' /
C     DATA RIGHT(1), RIGHT(2) / Z'3CA00000', Z'00000000' /
C     DATA DIVER(1), DIVER(2) / Z'3CB00000', Z'00000000' /
C     DATA LOG10(1), LOG10(2) / Z'3FD34413', Z'509F79FF' /
C
C     MACHINE CONSTANTS FOR THE CONVEX C-1
C
C     DATA SMALL(1), SMALL(2) / '00100000'X,'00000000'X /
C     DATA LARGE(1), LARGE(2) / '7FFFFFFF'X,'FFFFFFFF'X /
C     DATA RIGHT(1), RIGHT(2) / '3CC00000'X,'00000000'X /
C     DATA DIVER(1), DIVER(2) / '3CD00000'X,'00000000'X /
C     DATA LOG10(1), LOG10(2) / '3FF34413'X,'509F79FF'X /
C
C     MACHINE CONSTANTS FOR THE CRAY-1
C
C     DATA SMALL(1) / 201354000000000000000B /
C     DATA SMALL(2) / 000000000000000000000B /
C     DATA LARGE(1) / 577767777777777777777B /
C     DATA LARGE(2) / 000007777777777777774B /
C     DATA RIGHT(1) / 376434000000000000000B /
C     DATA RIGHT(2) / 000000000000000000000B /
C     DATA DIVER(1) / 376444000000000000000B /
C     DATA DIVER(2) / 000000000000000000000B /
```

```
C     DATA LOG10(1) / 377774642023241175717B /
C     DATA LOG10(2) / 000007571421742254654B /
C
C     MACHINE CONSTANTS FOR THE DATA GENERAL ECLIPSE S/200
C
C     NOTE - IT MAY BE APPROPRIATE TO INCLUDE THE FOLLOWING CARD -
C     STATIC DMACH(5)
C
C     DATA SMALL /    20K, 3*0 /
C     DATA LARGE / 77777K, 3*177777K /
C     DATA RIGHT / 31420K, 3*0 /
C     DATA DIVER / 32020K, 3*0 /
C     DATA LOG10 / 40423K, 42023K, 50237K, 74776K /
C
C     MACHINE CONSTANTS FOR THE ELXSI 6400
C     (ASSUMING REAL*8 IS THE DEFAULT DOUBLE PRECISION)
C
C     DATA SMALL(1), SMALL(2) / '00100000'X,'00000000'X /
C     DATA LARGE(1), LARGE(2) / '7FEFFFFF'X,'FFFFFFFF'X /
C     DATA RIGHT(1), RIGHT(2) / '3CB00000'X,'00000000'X /
C     DATA DIVER(1), DIVER(2) / '3CC00000'X,'00000000'X /
C     DATA LOG10(1), LOG10(2) / '3FD34413'X,'509F79FF'X /
C
C     MACHINE CONSTANTS FOR THE HARRIS 220
C
C     DATA SMALL(1), SMALL(2) / '20000000, '00000201 /
C     DATA LARGE(1), LARGE(2) / '37777777, '37777577 /
C     DATA RIGHT(1), RIGHT(2) / '20000000, '00000333 /
C     DATA DIVER(1), DIVER(2) / '20000000, '00000334 /
C     DATA LOG10(1), LOG10(2) / '23210115, '10237777 /
C
C     MACHINE CONSTANTS FOR THE HONEYWELL 600/6000 SERIES
C
C     DATA SMALL(1), SMALL(2) / O402400000000, O000000000000 /
C     DATA LARGE(1), LARGE(2) / O376777777777, O777777777777 /
C     DATA RIGHT(1), RIGHT(2) / O604400000000, O000000000000 /
C     DATA DIVER(1), DIVER(2) / O606400000000, O000000000000 /
C     DATA LOG10(1), LOG10(2) / O776464202324, O117571775714 /
C
C     MACHINE CONSTANTS FOR THE HP 2100
C     THREE WORD DOUBLE PRECISION OPTION WITH FTN4
C
C     DATA SMALL(1), SMALL(2), SMALL(3) / 40000B,       0,       1 /
C     DATA LARGE(1), LARGE(2), LARGE(3) / 77777B, 177777B, 177776B /
C     DATA RIGHT(1), RIGHT(2), RIGHT(3) / 40000B,       0,    265B /
C     DATA DIVER(1), DIVER(2), DIVER(3) / 40000B,       0,    276B /
C     DATA LOG10(1), LOG10(2), LOG10(3) / 46420B,  46502B,  77777B /
C
C     MACHINE CONSTANTS FOR THE HP 2100
C     FOUR WORD DOUBLE PRECISION OPTION WITH FTN4
C
C     DATA SMALL(1), SMALL(2) /  40000B,       0 /
C     DATA SMALL(3), SMALL(4) /       0,       1 /
C     DATA LARGE(1), LARGE(2) /  77777B, 177777B /
C     DATA LARGE(3), LARGE(4) / 177777B, 177776B /
C     DATA RIGHT(1), RIGHT(2) /  40000B,       0 /
C     DATA RIGHT(3), RIGHT(4) /       0,    225B /
C     DATA DIVER(1), DIVER(2) /  40000B,       0 /
C     DATA DIVER(3), DIVER(4) /       0,    227B /
C     DATA LOG10(1), LOG10(2) /  46420B,  46502B /
C     DATA LOG10(3), LOG10(4) /  76747B, 176377B /
C
C     MACHINE CONSTANTS FOR THE HP 9000
C
C     DATA SMALL(1), SMALL(2) / 00040000000B, 00000000000B /
C     DATA LARGE(1), LARGE(2) / 17737777777B, 37777777777B /
C     DATA RIGHT(1), RIGHT(2) / 07454000000B, 00000000000B /
```

```
C     DATA DIVER(1), DIVER(2) / 07460000000B, 00000000000B /
C     DATA LOG10(1), LOG10(2) / 07764642023B, 12047674777B /
C
C     MACHINE CONSTANTS FOR THE IBM 360/370 SERIES,
C     THE XEROX SIGMA 5/7/9, THE SEL SYSTEMS 85/86, AND
C     THE PERKIN ELMER (INTERDATA) 7/32.
C
C     DATA SMALL(1), SMALL(2) / Z00100000, Z00000000 /
C     DATA LARGE(1), LARGE(2) / Z7FFFFFFF, ZFFFFFFFF /
C     DATA RIGHT(1), RIGHT(2) / Z33100000, Z00000000 /
C     DATA DIVER(1), DIVER(2) / Z34100000, Z00000000 /
C     DATA LOG10(1), LOG10(2) / Z41134413, Z509F79FF /
C
C     MACHINE CONSTANTS FOR THE IBM PC
C     ASSUMES THAT ALL ARITHMETIC IS DONE IN DOUBLE PRECISION
C     ON 8088, I.E., NOT IN 80 BIT FORM FOR THE 8087.
C
      DATA SMALL(1),SMALL(2) /  2002288515,    1050897 /
      DATA LARGE(1),LARGE(2) /  1487780761, 2146426097 /
      DATA RIGHT(1),RIGHT(2) / -1209488034, 1017118298 /
      DATA DIVER(1),DIVER(2) / -1209488034, 1018166874 /
      DATA LOG10(1),LOG10(2) /  1352628735, 1070810131 /
C
C     MACHINE CONSTANTS FOR THE IBM RS 6000
C
C     DATA SMALL(1), SMALL(2) / Z'00100000', Z'00000000' /
C     DATA LARGE(1), LARGE(2) / Z'7FEFFFFF', Z'FFFFFFFF' /
C     DATA RIGHT(1), RIGHT(2) / Z'3CA00000', Z'00000000' /
C     DATA DIVER(1), DIVER(2) / Z'3CB00000', Z'00000000' /
C     DATA LOG10(1), LOG10(2) / Z'3FD34413', Z'509F79FF' /
C
C     MACHINE CONSTANTS FOR THE PDP-10 (KA PROCESSOR)
C
C     DATA SMALL(1), SMALL(2) / "033400000000, "000000000000 /
C     DATA LARGE(1), LARGE(2) / "377777777777, "344777777777 /
C     DATA RIGHT(1), RIGHT(2) / "113400000000, "000000000000 /
C     DATA DIVER(1), DIVER(2) / "114400000000, "000000000000 /
C     DATA LOG10(1), LOG10(2) / "177464202324, "144117571776 /
C
C     MACHINE CONSTANTS FOR THE PDP-10 (KI PROCESSOR)
C
C     DATA SMALL(1), SMALL(2) / "000400000000, "000000000000 /
C     DATA LARGE(1), LARGE(2) / "377777777777, "377777777777 /
C     DATA RIGHT(1), RIGHT(2) / "103400000000, "000000000000 /
C     DATA DIVER(1), DIVER(2) / "104400000000, "000000000000 /
C     DATA LOG10(1), LOG10(2) / "177464202324, "476747767461 /
C
C     MACHINE CONSTANTS FOR PDP-11 FORTRAN SUPPORTING
C     32-BIT INTEGERS (EXPRESSED IN INTEGER AND OCTAL).
C
C     DATA SMALL(1), SMALL(2) /    8388608,           0 /
C     DATA LARGE(1), LARGE(2) / 2147483647,          -1 /
C     DATA RIGHT(1), RIGHT(2) /  612368384,           0 /
C     DATA DIVER(1), DIVER(2) /  620756992,           0 /
C     DATA LOG10(1), LOG10(2) / 1067065498, -2063872008 /
C
C     DATA SMALL(1), SMALL(2) / O00040000000, O00000000000 /
C     DATA LARGE(1), LARGE(2) / O17777777777, O37777777777 /
C     DATA RIGHT(1), RIGHT(2) / O04440000000, O00000000000 /
C     DATA DIVER(1), DIVER(2) / O04500000000, O00000000000 /
C     DATA LOG10(1), LOG10(2) / O07746420232, O20476747770 /
C
C     MACHINE CONSTANTS FOR PDP-11 FORTRAN SUPPORTING
C     16-BIT INTEGERS (EXPRESSED IN INTEGER AND OCTAL).
C
C     DATA SMALL(1), SMALL(2) /    128,      0 /
C     DATA SMALL(3), SMALL(4) /      0,      0 /
```

```
C     DATA LARGE(1), LARGE(2) /  32767,     -1 /
C     DATA LARGE(3), LARGE(4) /     -1,     -1 /
C     DATA RIGHT(1), RIGHT(2) /   9344,      0 /
C     DATA RIGHT(3), RIGHT(4) /      0,      0 /
C     DATA DIVER(1), DIVER(2) /   9472,      0 /
C     DATA DIVER(3), DIVER(4) /      0,      0 /
C     DATA LOG10(1), LOG10(2) /  16282,   8346 /
C     DATA LOG10(3), LOG10(4) / -31493, -12296 /
C
C     DATA SMALL(1), SMALL(2) / O000200, O000000 /
C     DATA SMALL(3), SMALL(4) / O000000, O000000 /
C     DATA LARGE(1), LARGE(2) / O077777, O177777 /
C     DATA LARGE(3), LARGE(4) / O177777, O177777 /
C     DATA RIGHT(1), RIGHT(2) / O022200, O000000 /
C     DATA RIGHT(3), RIGHT(4) / O000000, O000000 /
C     DATA DIVER(1), DIVER(2) / O022400, O000000 /
C     DATA DIVER(3), DIVER(4) / O000000, O000000 /
C     DATA LOG10(1), LOG10(2) / O037632, O020232 /
C     DATA LOG10(3), LOG10(4) / O102373, O147770 /
C
C     MACHINE CONSTANTS FOR THE SILICON GRAPHICS IRIS
C
c     data dmach(1) / 2.22507 38585 072012 d-308 /
c     data dmach(2) / 1.79769 31348 623158 d+308 /
c     data dmach(3) / 2.22044 60492 503131 d-16  /
c     data dmach(4) / 4.44089 20985 006262 d-16  /
c     data dmach(5) / 0.30102 99956 639812       /
C
C     DATA SMALL(1), SMALL(2) / Z'00100000',Z'00000000' /
C     DATA LARGE(1), LARGE(2) / Z'7FEFFFFF',Z'FFFFFFFF' /
C     DATA RIGHT(1), RIGHT(2) / Z'3CB00000',Z'00000000' /
C     DATA DIVER(1), DIVER(2) / Z'3CC00000',Z'00000000' /
C     DATA LOG10(1), LOG10(2) / Z'3FD34413',Z'509F79FF' /
C
C     MACHINE CONSTANTS FOR THE SUN
C
C     from SLATEC CML committee - work for Sun3, Sun4, and Sparc
C
C     DATA SMALL(1), SMALL(2) / Z'00100000', Z'00000000' /
C     DATA LARGE(1), LARGE(2) / Z'7FEFFFFF', Z'FFFFFFFF' /
C     DATA RIGHT(1), RIGHT(2) / Z'3CA00000', Z'00000000' /
C     DATA DIVER(1), DIVER(2) / Z'3CB00000', Z'00000000' /
C     DATA LOG10(1), LOG10(2) / Z'3FD34413', Z'509F79FF' /
C
C     from Sun Microsystems - work for Sun 386i
C
C     DATA SMALL(1), SMALL(2) / Z'00000000', Z'00100000' /
C     DATA LARGE(1), LARGE(2) / Z'FFFFFFFF', Z'7FEFFFFF' /
C     DATA RIGHT(1), RIGHT(2) / Z'00000000', Z'3CA00000' /
C     DATA DIVER(1), DIVER(2) / Z'00000000', Z'3CB00000' /
C     DATA LOG10(1), LOG10(2) / Z'509F79FF', Z'3FD34413' /
C
C     MACHINE CONSTANTS FOR THE UNIVAC 1100 SERIES FTN COMPILER
C
C     DATA SMALL(1), SMALL(2) / O000040000000, O000000000000 /
C     DATA LARGE(1), LARGE(2) / O377777777777, O777777777777 /
C     DATA RIGHT(1), RIGHT(2) / O170540000000, O000000000000 /
C     DATA DIVER(1), DIVER(2) / O170640000000, O000000000000 /
C     DATA LOG10(1), LOG10(2) / O177746420232, O411757177572 /
C
C     MACHINE CONSTANTS FOR VAX 11/780
C     (EXPRESSED IN INTEGER AND HEXADECIMAL)
C     THE HEX FORMAT BELOW MAY NOT BE SUITABLE FOR UNIX SYSYEMS
C     THE INTEGER FORMAT SHOULD BE OK FOR UNIX SYSTEMS
C
C     DATA SMALL(1), SMALL(2) /        128,           0 /
C     DATA LARGE(1), LARGE(2) /     -32769,          -1 /
```

```
C     DATA RIGHT(1), RIGHT(2) /       9344,           0 /
C     DATA DIVER(1), DIVER(2) /       9472,           0 /
C     DATA LOG10(1), LOG10(2) /  546979738,  -805796613 /
C
C     DATA SMALL(1), SMALL(2) / Z00000080, Z00000000 /
C     DATA LARGE(1), LARGE(2) / ZFFFF7FFF, ZFFFFFFFF /
C     DATA RIGHT(1), RIGHT(2) / Z00002480, Z00000000 /
C     DATA DIVER(1), DIVER(2) / Z00002500, Z00000000 /
C     DATA LOG10(1), LOG10(2) / Z209A3F9A, ZCFF884FB /
C
C     MACHINE CONSTANTS FOR VAX 11/780 (G-FLOATING)
C     (EXPRESSED IN INTEGER AND HEXADECIMAL)
C     THE HEX FORMAT BELOW MAY NOT BE SUITABLE FOR UNIX SYSYEMS
C     THE INTEGER FORMAT SHOULD BE OK FOR UNIX SYSTEMS
C
C     DATA SMALL(1), SMALL(2) /         16,           0 /
C     DATA LARGE(1), LARGE(2) /     -32769,          -1 /
C     DATA RIGHT(1), RIGHT(2) /      15552,           0 /
C     DATA DIVER(1), DIVER(2) /      15568,           0 /
C     DATA LOG10(1), LOG10(2) /  1142112243, 2046775455 /
C
C     DATA SMALL(1), SMALL(2) / Z00000010, Z00000000 /
C     DATA LARGE(1), LARGE(2) / ZFFFF7FFF, ZFFFFFFFF /
C     DATA RIGHT(1), RIGHT(2) / Z00003CC0, Z00000000 /
C     DATA DIVER(1), DIVER(2) / Z00003CD0, Z00000000 /
C     DATA LOG10(1), LOG10(2) / Z44133FF3, Z79FF509F /
C
C
C***FIRST EXECUTABLE STATEMENT  D1MACH
      IF (I .LT. 1  .OR.  I .GT. 5)
     1   CALL XERROR ('D1MACH -- I OUT OF BOUNDS', 25, 1, 2)
C
      D1MACH = DMACH(I)
      RETURN
C
      END
      SUBROUTINE FDUMP
C***BEGIN PROLOGUE  FDUMP
C***DATE WRITTEN   790801   (YYMMDD)
C***REVISION DATE  861211   (YYMMDD)
C***CATEGORY NO.  R3
C***KEYWORDS  LIBRARY=SLATEC(XERROR),TYPE=ALL(FDUMP-A),ERROR
C***AUTHOR  JONES, R. E., (SNLA)
C***PURPOSE  Symbolic dump (should be locally written).
C***DESCRIPTION
C
C        ***Note*** Machine Dependent Routine
C        FDUMP is intended to be replaced by a locally written
C        version which produces a symbolic dump.  Failing this,
C        it should be replaced by a version which prints the
C        subprogram nesting list.  Note that this dump must be
C        printed on each of up to five files, as indicated by the
C        XGETUA routine.  See XSETUA and XGETUA for details.
C
C     Written by Ron Jones, with SLATEC Common Math Library Subcommittee
C***REFERENCES  (NONE)
C***ROUTINES CALLED  (NONE)
C***END PROLOGUE  FDUMP
C***FIRST EXECUTABLE STATEMENT  FDUMP
      RETURN
      END
      SUBROUTINE XERROR(MESS,NMESS,NERR,LEVEL)
C***BEGIN PROLOGUE  XERROR
C***DATE WRITTEN   880401   (YYMMDD)
C***REVISION DATE  880401   (YYMMDD)
C***CATEGORY NO.  R3C
C***KEYWORDS  ERROR,XERROR PACKAGE
```

```
C***AUTHOR  AMOS, D. E., (SNLA)
C***PURPOSE  TO PRINT AN ERROR MESSAGE
C***DESCRIPTION
C
C     ABSTRACT
C        XERROR IS A DUMMY SLATEC LIBRARY ROUTINE TO PRINT AN ERROR
C        MESSAGE.  THE CALL LIST IS CONSISTENT WITH THE CORRESPONDING
C        SLATEC LIBRARY SUBROUTINE.
C
C     DESCRIPTION OF PARAMETERS
C      --INPUT--
C        MESSG - THE HOLLERITH MESSAGE TO BE PROCESSED.
C        NMESSG- THE ACTUAL NUMBER OF CHARACTERS IN MESSG.
C        NERR  - THE ERROR NUMBER ASSOCIATED WITH THIS MESSAGE.
C                (A DUMMY VARIABLE NOT USED, BUT NEEDED TO BE COMPATIBLE
C                 WITH THE CORRESPONDING SLATEC ROUTINE)
C        LEVEL - ERROR CATEGORY.
C                (A DUMMY VARIABLE NOT USED, BUT NEEDED TO BE COMPATIBLE
C                 WITH THE CORRESPONDING SLATEC ROUTINE)
C
C***REFERENCES  JONES R.E., KAHANER D.K., 'XERROR, THE SLATEC ERROR-
C                 HANDLING PACKAGE', SAND82-0800, SANDIA LABORATORIES,
C                 1982.
C***ROUTINES CALLED  NONE
C***END PROLOGUE  XERROR
      CHARACTER*(*) MESS
      INTEGER NMESS,NERR,LEVEL,NN,NR,K,I,KMIN,MIN
      IF (NMESS.LE.0) THEN
        PRINT *,' IN XERROR, NMESS IS OUT OF RANGE'
      ELSE
        NN=NMESS/70
        NR=NMESS-70*NN
        IF(NR.NE.0) NN=NN+1
        K=1
        PRINT 900
  900   FORMAT(/)
        DO 10 I=1,NN
          KMIN=MIN(K+69,NMESS)
          PRINT *, MESS(K:KMIN)
          K=K+70
   10   CONTINUE
        RETURN
      END IF
      END
      SUBROUTINE XERRWV(MESSG,NMESSG,NERR,LEVEL,NI,I1,I2,NR,R1,R2)
C***BEGIN PROLOGUE  XERRWV
C***DATE WRITTEN   880401   (YYMMDD)
C***REVISION DATE  880401   (YYMMDD)
C***CATEGORY NO.  R3C
C***KEYWORDS  ERROR,XERROR PACKAGE
C***AUTHOR  AMOS, D. E., (SNLA)
C***PURPOSE  PROCESS AN ERROR MESSAGE ALLOWING 2 INTEGER AND 2 REAL
C            VALUES TO BE INCLUDED IN THE MESSAGE.
C***DESCRIPTION
C
C     ABSTRACT
C        XERRWV IS A DUMMY SLATEC LIBRARY ROUTINE TO PRINT AN ERROR
C        MESSAGE AND PRINT UP TO 2 INTEGER VARIABLES AND 2 REAL
C        VARIABLES. THE CALL LIST IS CONSISTENT WITH THE CORRESPONDING
C        SLATEC LIBRARY SUBROUTINE.
C
C     DESCRIPTION OF PARAMETERS
C      --INPUT--
C        MESSG - THE HOLLERITH MESSAGE TO BE PROCESSED.
C        NMESSG- THE ACTUAL NUMBER OF CHARACTERS IN MESSG.
C        NERR  - THE ERROR NUMBER ASSOCIATED WITH THIS MESSAGE.
C                (A DUMMY VARIABLE NOT USED, BUT NEEDED TO BE COMPATIBLE
```

```
C                  WITH THE CORRESPONDING SLATEC ROUTINE)
C         LEVEL - ERROR CATEGORY.
C                 (A DUMMY VARIABLE NOT USED, BUT NEEDED TO BE COMPATIBLE
C                  WITH THE CORRESPONDING SLATEC ROUTINE)
C         NI    - NUMBER OF INTEGER VALUES TO BE PRINTED. (0 TO 2)
C         I1    - FIRST INTEGER VALUE.
C         I2    - SECOND INTEGER VALUE.
C         NR    - NUMBER OF REAL VALUES TO BE PRINTED. (0 TO 2)
C         R1    - FIRST REAL VALUE.
C         R2    - SECOND REAL VALUE.
C
C***REFERENCES  JONES R.E., KAHANER D.K., 'XERROR, THE SLATEC ERROR-
C                  HANDLING PACKAGE', SAND82-0800, SANDIA LABORATORIES,
C                  1982.
C***ROUTINES CALLED  XERROR
C***END PROLOGUE  XERRWV
      CHARACTER*(*) MESSG
      INTEGER I1,I2,NI,NR,NMESSG,NERR,LEVEL
      REAL R1,R2
      CALL XERROR(MESSG,NMESSG,NERR,LEVEL)
      IF (NI.LT.0 .OR. NI.GT.2) THEN
        PRINT *,' IN XERRWV, NI IS OUT OF RANGE'
      ELSE
        IF (NI.NE.0) THEN
          IF (NI.EQ.1) THEN
            PRINT *,' I1 = ',I1
          ELSE
            PRINT *,' I1 , I2 =',I1,I2
          END IF
        END IF
      END IF
      IF (NR.LT.0 .OR. NR.GT.2) THEN
        PRINT *,' IN XERRWV, NR IS OUT OF RANGE'
      ELSE
        IF (NR.NE.0) THEN
          IF (NR.EQ.1) THEN
            PRINT *,' R1 =',R1
          ELSE
            PRINT *,' R1 , R2 =',R1,R2
          END IF
        END IF
      END IF
      RETURN
      END
      SUBROUTINE DEXINT(X, N, KODE, M, TOL, EN, NZ, IERR)
C***BEGIN PROLOGUE  DEXINT
C***DATE WRITTEN   820601   (YYMMDD)
C***REVISION DATE  820601   (YYMMDD)
C***CATEGORY NO.  B5E
C***KEYWORDS  EXPONENTIAL INTEGRAL
C***AUTHOR  AMOS, DONALD E., SANDIA NATIONAL LABORATORIES
C***PURPOSE  DEXINT COMPUTES THE EXPONENTIAL INTEGRAL
C***DESCRIPTION
C
C                      * A DOUBLE PRECISION ROUTINE *
C          DEXINT COMPUTES M MEMBER SEQUENCES OF EXPONENTIAL INTEGRALS
C          E(N+K,X), K=0,1,...,M-1 FOR N.GE.1 AND X.GE.0.0D0.  THE
C          EXPONENTIAL INTEGRAL IS DEFINED BY
C
C          E(N,X)=INTEGRAL ON (1,INFINITY) OF EXP(-XT)/T**N
C
C          WHERE X=0.0 AND N=1 CANNOT OCCUR SIMULTANEOUSLY. FORMULAS
C          AND NOTATION ARE FOUND IN THE NBS HANDBOOK OF MATHEMATICAL
C          FUNCTIONS (REF. 1).
C
C          EXINT IS THE SINGLE PRECISION VERSION OF DEXINT
C
```

```
C         INPUT     X AND TOL ARE DOUBLE PRECISION
C           X       X.GT.0.0D0 FOR N=1 AND  X.GE.0.0D0 FOR N.GE.2
C           N       ORDER OF THE FIRST MEMBER OF THE SEQUENCE, N.GE.1
C                   (X=0.0D0 AND N=1 IS AN ERROR)
C           KODE    A SELECTION PARAMETER FOR SCALED VALUES
C                   KODE=1   RETURNS        E(N+K,X), K=0,1,...,M-1.
C                       =2   RETURNS EXP(X)*E(N+K,X), K=0,1,...,M-1.
C           M       NUMBER OF EXPONENTIAL INTEGRALS IN THE SEQUENCE,
C                   M.GE.1
C           TOL     RELATIVE ACCURACY WANTED, ETOL.LE.TOL.LE.0.1D0
C                   ETOL = MAXIMUM OF DOUBLE PRECISION UNIT ROUNDOFF
C                   (D1MACH(4)) AND 1.0D-18
C
C         OUTPUT    EN IS A DOUBLE PRECISION VECTOR
C           EN      A VECTOR OF DIMENSION AT LEAST M CONTAINING VALUES
C                   EN(K) = E(N+K-1,X) OR EXP(X)*E(N+K-1,X), K=1,M
C                   DEPENDING ON KODE
C           NZ      UNDERFLOW INDICATOR
C                   NZ=0   A NORMAL RETURN
C                   NZ=M   X EXCEEDS XLIM AND AN UNDERFLOW OCCURS.
C                          EN(K)=0.0D0 , K=1,M RETURNED ON KODE=1
C           IERR    ERROR FLAG
C                   IERR=0, NORMAL RETURN, COMPUTATION COMPLETED
C                   IERR=1, INPUT ERROR,   NO COMPUTATION
C                   IERR=2, ERROR,         NO COMPUTATION
C                           ALGORITHM TERMINATION CONDITION NOT MET
C
C***REFERENCES  HANDBOOK OF MATHEMATICAL FUNCTIONS BY M. ABRAMOWITZ
C                 AND I. A. STEGUN, NBS AMS SERIES 55, U.S. DEPT. OF
C                 COMMERCE, 1955.
C
C               COMPUTATION OF EXPONENTIAL INTEGRALS BY D. E. AMOS,
C                 ACM TRANS. MATH SOFTWARE, 6, 1980,PP. 365-377, 420-428
C***ROUTINES CALLED  I1MACH,D1MACH,DPSIXN
C***END PROLOGUE  DEXINT
C
      DOUBLE PRECISION A, AA, AAMS, AH, AK, AT, B, BK, BT, CC, CNORM,
     * CT, EM, EMX, EN, ETOL, FNM, FX, PT, P1, P2, S, TOL, TX, X, XCUT,
     * XLIM, XTOL, Y, YT, Y1, Y2, FN, FKN, XOVR, RIX
      DOUBLE PRECISION D1MACH, DPSIXN
      INTEGER I, IC, ICASE, ICT, IERR, IK, IND, IX, I1M, JSET, K, KK,
     * KN, KODE, KS, M, ML, MU, N, ND, NM, NZ, I1P
      INTEGER I1MACH
      DIMENSION EN(M), A(99), B(99), Y(2)
C
      DATA XCUT                  /     2.0D0                 /
C
C***FIRST EXECUTABLE STATEMENT  DEXINT
      IERR=0
      NZ=0
      ETOL = DMAX1(D1MACH(4),0.5D-18)
      IF (X.LT.0.0D0) IERR=1
      IF (N.LT.1) IERR=1
      IF (KODE.LT.1 .OR. KODE.GT.2) IERR=1
      IF (M.LT.1) IERR=1
      IF (TOL.LT.ETOL .OR. TOL.GT.0.1D0) IERR=1
      IF (X.EQ.0.0D0 .AND. N.EQ.1) IERR=1
      IF(IERR.NE.0) RETURN
      I1M = -I1MACH(15)
      PT = 2.3026D0*DBLE(FLOAT(I1M))*D1MACH(5)
      XLIM = PT - 6.907755D0
      BT = PT + DBLE(FLOAT(N+M-1))
      IF (BT.GT.1000.0D0) XLIM = PT - DLOG(BT)
C-----------------------------------------------------------------------
C     SET OVERFLOW LIMIT ON X. THE UNDERFLOW LIMIT IS 1.0D0/XOVR.
C     TEST EXPONENTS ON SMALLEST AND LARGEST NUMBERS TO BE SURE THE
C     RECIPROCALS EXIST. EXPONENTS ARE I1MACH(15) AND I1MACH(16).
```

```
C-----------------------------------------------------------------------
      I1P = I1MACH(16)
      IF(I1M.LT.I1P) THEN
        XOVR = 1.0D-3/D1MACH(1)
      ELSE
        XOVR = 1.0D-3*D1MACH(2)
      ENDIF
      IF (X.GT.XOVR) GO TO 105
      IF (X.GT.XCUT) GO TO 100
C     IF (X.EQ.0.0D0 .AND. N.GT.1) GO TO 80
      IF (X.LT.1.0D0/XOVR) THEN
        IF ( N.GT.1 ) GO TO 80
        IERR=1
        RETURN
      ENDIF
C-----------------------------------------------------------------------
C     SERIES FOR E(N,X) FOR X.LE.XCUT
C-----------------------------------------------------------------------
C      TX = X + 0.5D0
C      IX = INT(SNGL(TX))
       RIX = ANINT(X)
C-----------------------------------------------------------------------
C     ICASE=1 MEANS INTEGER CLOSEST TO X IS 2 AND N=1
C     ICASE=2 MEANS INTEGER CLOSEST TO X IS 0,1, OR 2 AND N.GE.2
C-----------------------------------------------------------------------
      ICASE = 2
      IF (RIX.GT.DBLE(FLOAT(N))) ICASE = 1
      NM = N - ICASE + 1
      ND = NM + 1
      IND = 3 - ICASE
      MU = M - IND
      ML = 1
      KS = ND
      FNM = DBLE(FLOAT(NM))
      S = 0.0D0
      XTOL = 3.0D0*TOL
      IF (ND.EQ.1) GO TO 10
      XTOL = 0.3333D0*TOL
      S = 1.0D0/FNM
   10 CONTINUE
      AA = 1.0D0
      AK = 1.0D0
      IC = 35
      IF (X.LT.ETOL) IC = 1
      DO 50 I=1,IC
        AA = -AA*X/AK
        IF (I.EQ.NM) GO TO 30
        S = S - AA/(AK-FNM)
        IF (DABS(AA).LE.XTOL*DABS(S)) GO TO 20
        AK = AK + 1.0D0
        GO TO 50
   20   CONTINUE
        IF (I.LT.2) GO TO 40
        IF (ND-2.GT.I .OR. I.GT.ND-1) GO TO 60
        AK = AK + 1.0D0
        GO TO 50
   30   S = S + AA*(-DLOG(X)+DPSIXN(ND))
        XTOL = 3.0D0*TOL
   40   AK = AK + 1.0D0
   50 CONTINUE
      IF (IC.NE.1) GO TO 340
   60 IF (ND.EQ.1) S = S + (-DLOG(X)+DPSIXN(1))
      IF (KODE.EQ.2) S = S*DEXP(X)
      EN(1) = S
      EMX = 1.0D0
      IF (M.EQ.1) GO TO 70
      EN(IND) = S
```

```
      AA = DBLE(FLOAT(KS))
      IF (KODE.EQ.1) EMX = DEXP(-X)
      GO TO (220, 240), ICASE
   70 IF (ICASE.EQ.2) RETURN
      IF (KODE.EQ.1) EMX = DEXP(-X)
      EN(1) = (EMX-S)/X
      RETURN
   80 CONTINUE
      DO 90 I=1,M
        EN(I) = 1.0D0/DBLE(FLOAT(N+I-2))
   90 CONTINUE
      RETURN
C-----------------------------------------------------------------------
C     BACKWARD RECURSIVE MILLER ALGORITHM FOR
C              E(N,X)=DEXP(-X)*(X**(N-1))*U(N,N,X)
C     WITH RECURSION AWAY FROM N=INTEGER CLOSEST TO X.
C     U(A,B,X) IS THE SECOND CONFLUENT HYPERGEOMETRIC FUNCTION
C-----------------------------------------------------------------------
  100 CONTINUE
      EMX = 1.0D0
      IF (KODE.EQ.2) GO TO 130
      IF (X.LE.XLIM) GO TO 120
  105 CONTINUE
      NZ = M
      DO 110 I=1,M
        EN(I) = 0.0D0
  110 CONTINUE
      RETURN
  120 EMX = DEXP(-X)
  130 CONTINUE
C     TX = X + 0.5D0
C     IX = INT(SNGL(TX))
      RIX = ANINT(X)
      KN = N + M - 1
      FKN = DBLE(FLOAT(KN))
      IF (FKN.LE.RIX) GO TO 140
      FN = DBLE(FLOAT(N))
      IF (FN.LT.RIX .AND. RIX.LT.FKN) GO TO 170
      IF (FN.GE.RIX) GO TO 160
      GO TO 340
  140 ICASE = 1
      KS = KN
      ML = M - 1
      MU = -1
      IND = M
      IF (KN.GT.1) GO TO 180
  150 KS = 2
      ICASE = 3
      GO TO 180
  160 ICASE = 2
      IND = 1
      KS = N
      MU = M - 1
      IF (N.GT.1) GO TO 180
      IF (KN.EQ.1) GO TO 150
      RIX = 2.0D0
  170 CONTINUE
      IX=INT(SNGL(RIX))
      ICASE = 1
      KS = IX
      ML = IX - N
      IND = ML + 1
      MU = KN - IX
  180 CONTINUE
      IK = KS/2
      AH = DBLE(FLOAT(IK))
      JSET = 1 + KS - (IK+IK)
```

```
C-----------------------------------------------------------------------
C     START COMPUTATION FOR
C              EN(IND) = C*U( A , A ,X)    JSET=1
C              EN(IND) = C*U(A+1,A+1,X)    JSET=2
C     FOR AN EVEN INTEGER A.
C-----------------------------------------------------------------------
      IC = 0
      AA = AH + AH
      AAMS = AA - 1.0D0
      AAMS = AAMS*AAMS
      TX = X + X
      FX = TX + TX
      AK = AH
      XTOL = TOL
      IF (TOL.LE.1.0D-3) XTOL = 20.0D0*TOL
      CT = AAMS + FX*AH
C-----------------------------------------------------------------------
C     PREVENT OVERFLOW IN DEN OF EM = (AH+1.0D0)/((X+AA)*XTOL*SQRT(CT))
C-----------------------------------------------------------------------
      P1 = (AH+1.0D0)/((X+AA)*XTOL)
      P2 = SQRT(CT)
      IF( P1 .LT. P2/XOVR ) THEN
        Y(1) = 1.0D0/X
        Y(2) = 1.0D0/X
        GO TO 205
      ENDIF
      EM = P1/P2
      BK = AA
      CC = AH*AH
C-----------------------------------------------------------------------
C     FORWARD RECURSION FOR P(IC),P(IC+1) AND INDEX IC FOR BACKWARD
C     RECURSION
C-----------------------------------------------------------------------
      P1 = 0.0D0
      P2 = 1.0D0
  190 CONTINUE
      IF (IC.EQ.99) GO TO 340
      IC = IC + 1
      AK = AK + 1.0D0
      AT = BK/(BK+AK+CC+DBLE(FLOAT(IC)))
      BK = BK + AK + AK
      A(IC) = AT
      BT = (AK+AK+X)/(AK+1.0D0)
      B(IC) = BT
      PT = P2
      P2 = BT*P2 - AT*P1
      P1 = PT
      CT = CT + FX
      EM = EM*AT*(1.0D0-TX/CT)
      IF (EM*(AK+1.0D0).GT.P1*P1) GO TO 190
      ICT = IC
      KK = IC + 1
      BT = TX/(CT+FX)
      Y2 = (BK/(BK+CC+DBLE(FLOAT(KK))))*(P1/P2)*(1.0D0-BT+0.375D0*BT*BT)
      Y1 = 1.0D0
C-----------------------------------------------------------------------
C     BACKWARD RECURRENCE FOR
C              Y1=            C*U( A ,A,X)
C              Y2= C*(A/(1+A/2))*U(A+1,A,X)
C-----------------------------------------------------------------------
      DO 200 K=1,ICT
        KK = KK - 1
        YT = Y1
        Y1 = (B(KK)*Y1-Y2)/A(KK)
        Y2 = YT
  200 CONTINUE
C-----------------------------------------------------------------------
```

```
C     THE CONTIGUOUS RELATION
C              X*U(B,C+1,X)=(C-B)*U(B,C,X)+U(B-1,C,X)
C     WITH  B=A+1 , C=A IS USED FOR
C              Y(2) = C * U(A+1,A+1,X)
C     X IS INCORPORATED INTO THE NORMALIZING RELATION
C-----------------------------------------------------------------------
      PT = Y2/Y1
      CNORM = 1.0E0 - PT*(AH+1.0E0)/AA
      Y(1) = 1.0E0/(CNORM*AA+X)
      Y(2) = CNORM*Y(1)
  205 CONTINUE
      IF (ICASE.EQ.3) GO TO 210
      EN(IND) = EMX*Y(JSET)
      IF (M.EQ.1) RETURN
      AA = DBLE(FLOAT(KS))
      GO TO (220, 240), ICASE
C-----------------------------------------------------------------------
C     RECURSION SECTION  N*E(N+1,X) + X*E(N,X)=EMX
C-----------------------------------------------------------------------
  210 EN(1) = EMX*(1.0E0-Y(1))/X
      RETURN
  220 K = IND - 1
      DO 230 I=1,ML
        AA = AA - 1.0D0
        EN(K) = (EMX-AA*EN(K+1))/X
        K = K - 1
  230 CONTINUE
      IF (MU.LE.0) RETURN
      AA = DBLE(FLOAT(KS))
  240 K = IND
      DO 250 I=1,MU
        EN(K+1) = (EMX-X*EN(K))/AA
        AA = AA + 1.0D0
        K = K + 1
  250 CONTINUE
      RETURN
  340 CONTINUE
      IERR=2
      RETURN
      END
      DOUBLE PRECISION FUNCTION DPSIXN(N)
C***BEGIN PROLOGUE  DPSIXN
C***REFER TO  DEXINT,DBSKIN
C
C     THIS SUBROUTINE RETURNS VALUES OF PSI(X)=DERIVATIVE OF LOG
C     GAMMA(X), X.GT.0.0 AT INTEGER ARGUMENTS. A TABLE LOOK-UP IS
C     PERFORMED FOR N.LE.100, AND THE ASYMPTOTIC EXPANSION IS
C     EVALUATED FOR N.GT.100.
C
C***ROUTINES CALLED  D1MACH
C***END PROLOGUE  DPSIXN
C
      INTEGER N, K
      DOUBLE PRECISION AX, B, C, FN, RFN2, TRM, S, WDTOL
      DOUBLE PRECISION D1MACH
      DIMENSION B(6), C(100)
C
C             DPSIXN(N), N = 1,100
      DATA C(1), C(2), C(3), C(4), C(5), C(6), C(7), C(8), C(9), C(10),
     1     C(11), C(12), C(13), C(14), C(15), C(16), C(17), C(18),
     2     C(19), C(20), C(21), C(22), C(23), C(24)/
     3    -5.77215664901532861D-01,     4.22784335098467139D-01,
     4     9.22784335098467139D-01,     1.25611766843180047D+00,
     5     1.50611766843180047D+00,     1.70611766843180047D+00,
     6     1.87278433509846714D+00,     2.01564147795561000D+00,
     7     2.14064147795561000D+00,     2.25175258906672111D+00,
     8     2.35175258906672111D+00,     2.44266167997581202D+00,
```

```
     9     2.52599501330914535D+00,     2.60291809023222227D+00,
     A     2.67434666166079370D+00,     2.74101332832746037D+00,
     B     2.80351332832746037D+00,     2.86233685773922507D+00,
     C     2.91789241329478063D+00,     2.97052399224214905D+00,
     D     3.02052399224214905D+00,     3.06814303986119667D+00,
     E     3.11359758531574212D+00,     3.15707584618530734D+00/
      DATA C(25), C(26), C(27), C(28), C(29), C(30), C(31), C(32),
     1     C(33), C(34), C(35), C(36), C(37), C(38), C(39), C(40),
     2     C(41), C(42), C(43), C(44), C(45), C(46), C(47), C(48)/
     3     3.19874251285197401D+00,     3.23874251285197401D+00,
     4     3.27720405131351247D+00,     3.31424108835054951D+00,
     5     3.34995537406483522D+00,     3.38443813268552488D+00,
     6     3.41777146601885821D+00,     3.45002953053498724D+00,
     7     3.48127953053498724D+00,     3.51158256083801755D+00,
     8     3.54099432554389990D+00,     3.56956575411532847D+00,
     9     3.59734353189310625D+00,     3.62437055892013327D+00,
     A     3.65068634839381748D+00,     3.67632737403484313D+00,
     B     3.70132737403484313D+00,     3.72571761793728215D+00,
     C     3.74952714174680596D+00,     3.77278295570029433D+00,
     D     3.79551022842756706D+00,     3.81773245064978928D+00,
     E     3.83947158108457189D+00,     3.86074817682925274D+00/
      DATA C(49), C(50), C(51), C(52), C(53), C(54), C(55), C(56),
     1     C(57), C(58), C(59), C(60), C(61), C(62), C(63), C(64),
     2     C(65), C(66), C(67), C(68), C(69), C(70), C(71), C(72)/
     3     3.88158151016258607D+00,     3.90198967342789220D+00,
     4     3.92198967342789220D+00,     3.94159751656514710D+00,
     5     3.96082828579591633D+00,     3.97969621032421822D+00,
     6     3.99821472884273674D+00,     4.01639654702455492D+00,
     7     4.03425368988169777D+00,     4.05179754953082058D+00,
     8     4.06903892884116541D+00,     4.08598808138353829D+00,
     9     4.10265474805020496D+00,     4.11904819067315578D+00,
     A     4.13517722293122029D+00,     4.15105023880423617D+00,
     B     4.16667523880423617D+00,     4.18205985418885155D+00,
     C     4.19721136934036670D+00,     4.21213674247469506D+00,
     D     4.22684262482763624D+00,     4.24133537845082464D+00,
     E     4.25562109273653893D+00,     4.26970559977879245D+00/
      DATA C(73), C(74), C(75), C(76), C(77), C(78), C(79), C(80),
     1     C(81), C(82), C(83), C(84), C(85), C(86), C(87), C(88),
     2     C(89), C(90), C(91), C(92), C(93), C(94), C(95), C(96)/
     3     4.28359448866768134D+00,     4.29729311880466764D+00,
     4     4.31080663231818115D+00,     4.32413996565151449D+00,
     5     4.33729786038835659D+00,     4.35028487337536958D+00,
     6     4.36310538619588240D+00,     4.37576361404398366D+00,
     7     4.38826361404398366D+00,     4.40060929305632934D+00,
     8     4.41280441500754886D+00,     4.42485260777863319D+00,
     9     4.43675736968339510D+00,     4.44852207556574804D+00,
     A     4.46014998254249223D+00,     4.47164423541605544D+00,
     B     4.48300787177969181D+00,     4.49424382683587158D+00,
     C     4.50535493794698269D+00,     4.51634394893599368D+00,
     D     4.52721351415338499D+00,     4.53796620232542800D+00,
     E     4.54860450019776842D+00,     4.55913081598724211D+00/
      DATA C(97), C(98), C(99), C(100)/
     1     4.56954748265390877D+00,     4.57985676100442424D+00,
     2     4.59006084263707730D+00,     4.60016185273808740D+00/
C             COEFFICIENTS OF ASYMPTOTIC EXPANSION
      DATA B(1), B(2), B(3), B(4), B(5), B(6)/
     1     8.33333333333333333D-02,    -8.33333333333333333D-03,
     2     3.96825396825396825D-03,    -4.16666666666666666D-03,
     3     7.57575757575757576D-03,    -2.10927960927960928D-02/
C
      IF (N.GT.100) GO TO 10
      DPSIXN = C(N)
      RETURN
   10 CONTINUE
      WDTOL = DMAX1(D1MACH(4),1.0D-18)
      FN = DBLE(FLOAT(N))
      AX = 1.0D0
```

```
      S = -0.5D0/FN
      IF (DABS(S).LE.WDTOL) GO TO 30
      RFN2 = 1.0D0/(FN*FN)
      DO 20 K=1,6
        AX = AX*RFN2
        TRM = -B(K)*AX
        IF (DABS(TRM).LT.WDTOL) GO TO 30
        S = S + TRM
   20 CONTINUE
   30 CONTINUE
      DPSIXN = S + DLOG(FN)
      RETURN
      END
      SUBROUTINE DHEXINT(X, FNH, KODE, M, TOL, EN, NZ, IERR)
C***BEGIN PROLOGUE  DEXINT
C***DATE WRITTEN   000704   (YYMMDD)
C***REVISION DATE  020425   (YYMMDD)
C***CATEGORY NO.  B5E
C***KEYWORDS  EXPONENTIAL INTEGRAL
C***AUTHOR  AMOS, DONALD E.
C***PURPOSE  DHEXINT COMPUTES THE EXPONENTIAL INTEGRAL OF HALF ODD ORDER
C***DESCRIPTION
C
C                    * A DOUBLE PRECISION ROUTINE *
C         DHEXINT COMPUTES M MEMBER SEQUENCES OF EXPONENTIAL INTEGRALS
C         E(FNH+K,X), K=0,1,...,M-1 FOR HALF ODD ORDERS FNH.GE.0.5 AND
C         X.GE.0.0D0.  THE EXPONENTIAL INTEGRAL IS DEFINED BY
C
C         E(FNH,X)=INTEGRAL ON (1,INFINITY) OF EXP(-XT)/T**FNH
C
C         WHERE X=0.0 AND FHN=0.5 CANNOT OCCUR SIMULTANEOUSLY. FORMULAS
C         AND NOTATION ARE FOUND IN THE NBS HANDBOOK OF MATHEMATICAL
C         FUNCTIONS (REF. 1). THIS IS AN ADAPTATION OF REF. 2 TO HALF
C         ODD ORDERS. E(0.5,X)=SQRT(PI)*ERFC(Z)/Z,  Z=SQRT(X), WITH
C         FORWARD RECURRENCE IS USED IN PLACE OF THE SERIES FOR INTEGER
C         ORDERS.
C
C         INPUT     X, FNH AND TOL ARE DOUBLE PRECISION
C           X       X.GT.0.0D0 FOR FNH=0.5 AND  X.GE.0.0D0 FOR FNH.GE.1.5
C           FNH     ORDER OF THE FIRST MEMBER OF THE SEQUENCE, FNH.GE.0.5
C           KODE    A SELECTION PARAMETER FOR SCALED VALUES
C                   KODE=1   RETURNS        E(FNH+K,X), K=0,1,...,M-1.
C                       =2   RETURNS EXP(X)*E(FNH+K,X), K=0,1,...,M-1.
C           M       NUMBER OF EXPONENTIAL INTEGRALS IN THE SEQUENCE,
C                   M.GE.1
C           TOL     RELATIVE ACCURACY WANTED, ETOL.LE.TOL.LE.0.1D0
C                   ETOL = MAXIMUM OF DOUBLE PRECISION UNIT ROUNDOFF
C                   (D1MACH(4)) AND 1.0D-18
C
C         OUTPUT    EN IS A DOUBLE PRECISION VECTOR
C           EN      A VECTOR OF DIMENSION AT LEAST M CONTAINING VALUES
C                   EN(K) = E(FNH+K-1,X) OR EXP(X)*E(FNH+K-1,X), K=1,M
C                   DEPENDING ON KODE
C           NZ      UNDERFLOW INDICATOR
C                   NZ=0   A NORMAL RETURN
C                   NZ=M   X EXCEEDS XLIM AND AN UNDERFLOW OCCURS.
C                          EN(K)=0.0D0 , K=1,M RETURNED ON KODE=1
C           IERR    ERROR FLAG
C                   IERR=0, NORMAL RETURN, COMPUTATION COMPLETED
C                   IERR=1, INPUT ERROR,   NO COMPUTATION
C                   IERR=2, ERROR,         NO COMPUTATION
C                           ALGORITHM TERMINATION CONDITION NOT MET
C
C***REFERENCES  HANDBOOK OF MATHEMATICAL FUNCTIONS BY M. ABRAMOWITZ
C                 AND I. A. STEGUN, NBS AMS SERIES 55, U.S. DEPT. OF
C                 COMMERCE, 1955.
C
```

```
C                COMPUTATION OF EXPONENTIAL INTEGRALS BY D. E. AMOS,
C                  ACM TRANS. MATH SOFTWARE, 6, 1980,PP. 365-377, 420-428
C
C                FOLDER 18 BY D. E. AMOS EVALUATION OF E(NU,X), G(NU,X)=
C                  -DE(NU,X)/DNU FOR INTEGER AND HALF ODD INTEGER ORDERS
C                  WITH APPLICATION TO I(N,B,T)= INTEGRAL ON (T,INFINITY)
C                  OF EXP(-B^2*X^2)*LN(X)/X**N, N=1,2,3,...
C
C***ROUTINES CALLED  I1MACH,D1MACH,DRERF
C***END PROLOGUE  DHEXINT
C       IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
      DOUBLE PRECISION A, AA, AAMS, AH, AK, AT, B, BK, BT, CC, CNORM,
     * CT, EM, EMX, EN, ERF2, ETOL, FX, FN, FNH, FKN, PT, P1, P2, RIX,
     * RTZ, RTPI, TOL, TX, X, XCUT, XLIM, XTOL, YT, Y1, Y2, XOVR
      DOUBLE PRECISION D1MACH, DRERF
      INTEGER I, IC, ICASE, ICT, IERR, IND, IX, I1M, K, KK,
     * KODE, KODEP, M, ML, MU, N, NH, NZ, I1P
      INTEGER I1MACH
      DIMENSION EN(M), A(99), B(99)
C
      DATA XCUT                  /     2.0D0                  /
      DATA RTPI                  /1.772453850905516D0/
C
C***FIRST EXECUTABLE STATEMENT  DHEXINT
      IERR=0
      NZ=0
      ETOL = DMAX1(D1MACH(4),0.5D-18)
      IF (X.LT.0.0D0) IERR=1
      IF (FNH.LT.0.5D0) IERR=1
      FN=FNH-0.5D0
      IF (FN-DBLE(FLOAT(INT(FN))).NE.0.0D0) IERR=1
      IF (KODE.LT.1 .OR. KODE.GT.2) IERR=1
      IF (M.LT.1) IERR=1
      IF (TOL.LT.ETOL .OR. TOL.GT.0.1D0) IERR=1
      IF (X.EQ.0.0D0 .AND. FNH.EQ.0.5D0) IERR=1
      IF (IERR.NE.0) RETURN
      I1M = -I1MACH(15)
      PT = 2.3026D0*DBLE(FLOAT(I1M))*D1MACH(5)
      XLIM = PT - 6.907755D0
      BT = PT + DBLE(FLOAT(N+M-1))
      IF (BT.GT.1000.0D0) XLIM = PT - DLOG(BT)
C-----------------------------------------------------------------------
C     SET OVERFLOW LIMIT ON X. THE UNDERFLOW LIMIT IS 1.0D0/XOVR.
C     TEST EXPONENTS ON SMALLEST AND LARGEST NUMBERS TO BE SURE THE
C     RECIPROCALS EXIST. EXPONENTS ARE I1MACH(15) AND I1MACH(16).
C-----------------------------------------------------------------------
      I1P = I1MACH(16)
      IF(I1M.LT.I1P) THEN
        XOVR = 1.0D-3/D1MACH(1)
      ELSE
        XOVR = 1.0D-3*D1MACH(2)
      ENDIF
      IF (X.GT.XOVR) GO TO 105
      IF (X.LT.1.0D0/XOVR) THEN
        IF (FNH.GT.0.5D0) GOTO 80
        IERR=1
        RETURN
      ENDIF
C-----------------------------------------------------------------------
C     START SEQUENCE AT E(0.5,X) FOR X.LE.XCUT
C-----------------------------------------------------------------------
      IF (X.GT.XCUT) GOTO 100
      IF (FNH.GT.20.0D0) GOTO 100
      RTZ=DSQRT(X)
      IF (KODE.EQ.1) THEN
        EMX=DEXP(-X)
```

```
      ELSE
        EMX=1.0D0
      ENDIF
      IF (FNH.EQ.0.5D0) THEN
        NZERO=0
        KODEP=KODE+1
        ERF2=DRERF(RTZ,KODEP,NZERO)
        Y1=RTPI*ERF2/RTZ
        EN(1)=Y1
        IF (M.EQ.1) RETURN
C        FNH=1.5 HERE E(3/2,X)=2*RTPI*IERFC(SQRT(X))
        CALL DIERFC(RTZ,KODE,YT,IERR)
        Y1=(YT+YT)*RTPI
        EN(2)=Y1
        IF(M.EQ.2) RETURN
        IND = 2
        MU = M-2
        AA = 1.5D0
      ELSE
        CALL DIERFC(RTZ,KODE,YT,IERR)
        Y1=(YT+YT)*RTPI
        AA=1.5D0
        NH=INT(FN)-1
        IF (NH.LE.0) GOTO 11
        DO 10 K=1,NH
          Y2=(EMX-X*Y1)/AA
          Y1=Y2
          YI1=(Y2-X*YI1)/AA
          AA=AA+1.0D0
   10   CONTINUE
   11   CONTINUE
        EN(1)=Y1
        IF(M.EQ.1) RETURN
        IND = 1
        MU = M-1
      ENDIF
      GOTO 240
   80 CONTINUE
      DO 90 I=1,M
        EN(I) = 1.0D0/(FNH-1.0D0+DBLE(FLOAT(I-1)))
   90 CONTINUE
      RETURN
C-----------------------------------------------------------------------
C     BACKWARD RECURSIVE MILLER ALGORITHM FOR
C              E(N,X)=DEXP(-X)*(X**(N-1))*U(N,N,X)
C     WITH RECURSION AWAY FROM N=INTEGER CLOSEST TO X.
C     U(A,B,X) IS THE SECOND CONFLUENT HYPERGEOMETRIC FUNCTION
C-----------------------------------------------------------------------
  100 CONTINUE
      EMX = 1.0D0
      IF (KODE.EQ.2) GO TO 130
      IF (X.LE.XLIM) GO TO 120
  105 CONTINUE
      IF (FNH.EQ.0.5D0) THEN
        IERR=1
        RETURN
      ENDIF
      NZ = M
      DO 110 I=1,M
        EN(I) = 0.0D0
  110 CONTINUE
      RETURN
  120 EMX = DEXP(-X)
  130 CONTINUE
      MF=0
      RIX = ANINT(X)+0.5D0
      FKN = FNH+DBLE(FLOAT(M-1))
```

```
      IF (FKN.LE.RIX) GO TO 140
      IF (FNH.LT.RIX .AND. RIX.LT.FKN) GO TO 170
      IF (FNH.GE.RIX) GO TO 160
      GO TO 340
  140 ICASE = 1
C     WHOLE SEQUENCE LIES TO THE LEFT OF RIX. START AT FKS, RECUR DOWN
C     ADJUST INDEX UP TO GET BETTER CONVERGENCE AND MORE ACCURACY
C     RECUR DOWN MF TIMES (UNINDEXED) TO GET BACK TO ORIGINAL INDEX
      FKS=FKN
      IF(FKN.LT.2.5D0) THEN
        FKS=2.5D0
        MF=INT(SNGL(2.5D0-FKN))
      ENDIF
      ML = M - 1
      MU = -1
      IND = M
      GOTO 180
  160 ICASE = 2
C     WHOLE SEQUENCE LIES TO THE RIGHT OF RIX. START AT FNH, RECUR UP
      IND = 1
      FKS=FNH
      MU = M - 1
      ML = -1
      GOTO 180
  170 CONTINUE
C     RIX LIES IN THE RANGE OF THE SEQUENCE. START AT RIX, RECUR UP &
C     DOWN
      IX=INT(SNGL(RIX))
      ICASE = 1
      N = INT(FN)
      ML = IX - N
      IND = ML + 1
      FKS = FNH+DBLE(FLOAT(IND-1))
      MU = M - IND
  180 CONTINUE
       AH = FKS/2.0D0
C-----------------------------------------------------------------------
C     START COMPUTATION FOR
C              EN(IND) = C*U( A , A ,X)
C-----------------------------------------------------------------------
      IC = 0
      AA = AH + AH
      AAMS = AA - 1.0D0
      AAMS = AAMS*AAMS
      TX = X + X
      FX = TX + TX
      AK = AH
      XTOL = TOL
      IF (TOL.LE.1.0D-3) XTOL = 20.0D0*TOL
      CT = AAMS + FX*AH
C-----------------------------------------------------------------------
C     PREVENT OVERFLOW IN DEN OF EM = (AH+1.0D0)/((X+AA)*XTOL*SQRT(CT))
C-----------------------------------------------------------------------
      P1 = (AH+1.0D0)/((X+AA)*XTOL)
      P2 = SQRT(CT)
      IF( P1 .LT. P2/XOVR ) THEN
        Y1 = 1.0D0/X
        GO TO 205
      ENDIF
      EM = P1/P2
      BK = AA
      CC = AH*AH
C-----------------------------------------------------------------------
C     FORWARD RECURSION FOR P(IC),P(IC+1) AND INDEX IC FOR BACKWARD
C     RECURSION
C-----------------------------------------------------------------------
      P1 = 0.0D0
```

```
      P2 = 1.0D0
  190 CONTINUE
      IF (IC.EQ.99) GO TO 340
      IC = IC + 1
      AK = AK + 1.0D0
      AT = BK/(BK+AK+CC+DBLE(FLOAT(IC)))
      BK = BK + AK + AK
      A(IC) = AT
      BT = (AK+AK+X)/(AK+1.0D0)
      B(IC) = BT
      PT = P2
      P2 = BT*P2 - AT*P1
      P1 = PT
      CT = CT + FX
      EM = EM*AT*(1.0D0-TX/CT)
      IF (EM*(AK+1.0D0).GT.P1*P1) GO TO 190
      ICT = IC
      KK = IC + 1
      BT = TX/(CT+FX)
      Y2 = (BK/(BK+CC+DBLE(FLOAT(KK))))*(P1/P2)*(1.0D0-BT+0.375D0*BT*BT)
      Y1 = 1.0D0
C-----------------------------------------------------------------------
C     BACKWARD RECURRENCE FOR
C              Y1=             C*U( A ,A,X)
C              Y2= C*(A/(1+A/2))*U(A+1,A,X)
C-----------------------------------------------------------------------
      DO 200 K=1,ICT
        KK = KK - 1
        YT = Y1
        Y1 = (B(KK)*Y1-Y2)/A(KK)
        Y2 = YT
  200 CONTINUE
C-----------------------------------------------------------------------
C     THE CONTIGUOUS RELATION
C              X*U(B,C+1,X)=(C-B)*U(B,C,X)+U(B-1,C,X)
C     WITH  B=A+1 , C=A IS USED FOR
C              Y(2) = C * U(A+1,A+1,X)
C     X IS INCORPORATED INTO THE NORMALIZING RELATION
C-----------------------------------------------------------------------
      PT = Y2/Y1
      CNORM = 1.0D0 - PT*(AH+1.0D0)/AA
      Y1 = 1.0D0/(CNORM*AA+X)
  205 CONTINUE
      IF (MF.NE.0) THEN
        AK=FKS
        DO 206 J=1,MF
          AK=AK-1.0D0
          Y1=(1.0D0-AK*Y1)/X
  206   CONTINUE
        FKS=AK
      ENDIF
      EN(IND) = EMX*Y1
      IF (M.EQ.1) RETURN
      AA = FKS
      GO TO (220, 240), ICASE
C-----------------------------------------------------------------------
C     RECURSION SECTION  FKH*E(FKH+1,X) + X*E(FKH,X)=EMX
C-----------------------------------------------------------------------
  220 K = IND-1
      IF (ML.LE.0) GOTO 235
      DO 230 I=1,ML
        AA = AA - 1.0D0
        EN(K) = (EMX-AA*EN(K+1))/X
        K = K - 1
  230 CONTINUE
  235 IF (MU.LE.0) RETURN
      AA = FKS
```

```
  240 K = IND
      DO 250 I=1,MU
        EN(K+1) = (EMX-X*EN(K))/AA
        AA = AA + 1.0D0
        K = K + 1
  250 CONTINUE
      RETURN
  340 CONTINUE
      IERR=2
      RETURN
      END
      SUBROUTINE DGEXINT(X, N, KODE, M, TOL, GN, EN, NZ, IERR)
C***BEGIN PROLOGUE  DGEXINT
C***DATE WRITTEN   020304   (YYMMDD)
C***REVISION DATE  020425   (YYMMDD)
C***CATEGORY NO.  B5E
C***KEYWORDS  EXPONENTIAL INTEGRAL
C***AUTHOR  AMOS, DONALD E.
C***PURPOSE  DGEXINT COMPUTES THE DERIVATIVE WRT ORDER OF THE
C***         EXPONENTIAL INTEGRAL AT INTEGER ORDERS AS WELL AS THE
C***         EXPONENTIAL INTEGRAL ITSELF AT INTEGER ORDERS
C***DESCRIPTION
C
C                    * A DOUBLE PRECISION ROUTINE *
C         DGEXINT COMPUTES M MEMBER SEQUENCES OF EXPONENTIAL INTEGRALS
C         E(N+K,X) AND G(N+K,X),  K=0,1,...,M-1 FOR INTEGER ORDERS
C         N.GE.1 AND X.GE.0.0D0. THE EXPONENTIAL INTEGRAL IS DEFINED BY
C
C         E(N,X)=INTEGRAL ON (1,INFINITY) OF EXP(-XT)/T**N
C
C         AND THE G INTEGRAL (=-DERIVATIVE WRT ORDER OF E AT N) IS
C         DEFINED BY
C
C         G(N,X)=INTEGRAL ON (1,INFINITY) OF E(N,XT)/T**N
C
C         WHERE X=0.0 AND N=1 CANNOT OCCUR SIMULTANEOUSLY. FORMULAS
C         AND NOTATION ARE FOUND IN THE NBS HANDBOOK OF MATHEMATICAL
C         FUNCTIONS (REF. 1). THIS IS AN ADAPTATION OF REF. 2 TO INTEGER
C         ORDERS. SERIES FOR E(1,X) OR E(2,X) ALONG WITH G(1,X) AND
C         G(2,X) ARE COMPUTED FOR X.LE.2 AND FORWARD RECURRENCE IS USED
C         FOR HIGHER INTEGER ORDERS. THE MILLER ALGORITHM IS RETAINED
C         FOR THE E FUNCTION AND THE DOCUMENTATION IN FOLDER 18 EXPLAINS
C         THE G COMPUTATION FOR X.GT.2.
C
C         INPUT     X , AND TOL ARE DOUBLE PRECISION
C           X       X.GT.0.0D0 FOR N=1 AND  X.GE.0.0D0 FOR N.GE.2
C           N       ORDER OF THE FIRST MEMBER OF THE SEQUENCE, N.GE.1
C           KODE    A SELECTION PARAMETER FOR SCALED VALUES
C                   KODE=1   RETURNS E(N+K,X), G(N+K,X) K=0,1,...,M-1.
C                       =2   RETURNS EXP(X)*E(N+K,X), EXP(X)*G(N,X),
C                                    K=0,1,...,M-1.
C           M       NUMBER OF INTEGRALS IN THE E AND G SEQUENCES, M.GE.1
C           TOL     RELATIVE ACCURACY WANTED, ETOL.LE.TOL.LE.0.1D0
C                   ETOL = MAXIMUM OF DOUBLE PRECISION UNIT ROUNDOFF
C                   (D1MACH(4)) AND 1.0D-18
C
C         OUTPUT    EN IS A DOUBLE PRECISION VECTOR
C           EN      A VECTOR OF DIMENSION AT LEAST M CONTAINING VALUES
C                   EN(K) = E(N+K-1,X) OR EXP(X)*E(N+K-1,X), K=1,M
C                   DEPENDING ON KODE
C           GN      A VECTOR OF DIMENSION AT LEAST M CONTAINING VALUES
C                   GN(K) = G(N+K-1,X) OR EXP(X)*G(N+K-1,X), K=1,M
C                   DEPENDING ON KODE
C           NZ      UNDERFLOW INDICATOR
C                   NZ=0   A NORMAL RETURN
C                   NZ=M   X EXCEEDS XLIM AND AN UNDERFLOW OCCURS.
C                          EN(K)=GN(K)=0.0D0 , K=1,M RETURNED ON KODE=1
```

```
C           IERR    ERROR FLAG
C                   IERR=0, NORMAL RETURN, COMPUTATION COMPLETED
C                   IERR=1, INPUT ERROR,   NO COMPUTATION
C                   IERR=2, ERROR,         NO COMPUTATION
C                           ALGORITHM TERMINATION CONDITION NOT MET
C
C***REFERENCES  HANDBOOK OF MATHEMATICAL FUNCTIONS BY M. ABRAMOWITZ
C                 AND I. A. STEGUN, NBS AMS SERIES 55, U.S. DEPT. OF
C                 COMMERCE, 1955.
C
C               COMPUTATION OF EXPONENTIAL INTEGRALS BY D. E. AMOS,
C                 ACM TRANS. MATH SOFTWARE, 6, 1980,PP. 365-377, 420-428
C
C               FOLDER 18 EVALUATION OF E(NU,X), G(NU,X)=-DE(NU,X)/DNU
C                 FOR INTEGER AND HALF ODD INTEGER ORDERS WITH
C                 APPLICATION TO I(N,B,T)= INTEGRAL ON (T,INFINITY) OF
C                 EXP(-B^2*X^2)*LN(X)/X**N, N.GE.0
C***ROUTINES CALLED  I1MACH,D1MACH
C***END PROLOGUE  DGEXINT
C       IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
      DOUBLE PRECISION A, AA, AAMS, AH, AK, AT, B, BK, BT, CC, CNORM,
     * CT, EM, EMX, EN, ETOL, FX, FN, FKN, PT, P1, P2, RIX,
     * TOL, TX, X, XCUT, XLIM, XTOL, YT, Y1, Y2, XOVR
      DOUBLE PRECISION D1MACH,  GN, YI1, DICT, DKCT, ANUM, ADEN
      DOUBLE PRECISION G1FUN, G2FUN, REL, PTOL
      INTEGER I, IC, ICASE, ICT, IERR, IND, IX, I1M, K, KK, MF,
     * KODE,  M, ML, MU, N, NZ, I1P, MND, ND
      INTEGER I1MACH
      DIMENSION EN(M), GN(M), A(150), B(150)
C
      DATA XCUT, PTOL /  2.0D0, 1.0D+10   /
C
C***FIRST EXECUTABLE STATEMENT  DGEXINT
      IERR=0
      NZ=0
      ETOL = DMAX1(D1MACH(4),0.5D-18)
      IF (X.LT.0.0D0) IERR=1
      IF (N.LT.1) IERR=1
      IF (KODE.LT.1 .OR. KODE.GT.2) IERR=1
      IF (M.LT.1) IERR=1
      IF (TOL.LT.ETOL .OR. TOL.GT.0.1D0) IERR=1
      IF (X.EQ.0.0D0 .AND. N.EQ.1) IERR=1
      IF (IERR.NE.0) RETURN
      I1M = -I1MACH(15)
      PT = 2.3026D0*DBLE(FLOAT(I1M))*D1MACH(5)
      XLIM = PT - 6.907755D0
      BT = PT + DBLE(FLOAT(N+M-1))
      IF (BT.GT.1000.0D0) XLIM = PT - DLOG(BT)
C-----------------------------------------------------------------------
C     SET OVERFLOW LIMIT ON X. THE UNDERFLOW LIMIT IS 1.0D0/XOVR.
C     TEST EXPONENTS ON SMALLEST AND LARGEST NUMBERS TO BE SURE THE
C     RECIPROCALS EXIST. EXPONENTS ARE I1MACH(15) AND I1MACH(16).
C-----------------------------------------------------------------------
      I1P = I1MACH(16)
      IF(I1M.LT.I1P) THEN
        XOVR = 1.0D-3/D1MACH(1)
      ELSE
        XOVR = 1.0D-3*D1MACH(2)
      ENDIF
      FN=DBLE(FLOAT(N))
      IF (X.GT.XOVR) GO TO 105
      IF (X.LT.1.0D0/XOVR) THEN
        IF (N.GT.1) GOTO 80
        IERR=1
        RETURN
      ENDIF
```

```
C-----------------------------------------------------------------------
C     START SEQUENCE AT E(1,X),G(1,X) OR E(2,X),G(2,X)  FOR X.LE.XCUT
C-----------------------------------------------------------------------
      IF (X.GT.XCUT) GOTO 100
      IF (N.GT.20) GOTO 100
      IF(KODE.EQ.1) THEN
        EMX=DEXP(-X)
      ELSE
        EMX=1.0D0
      ENDIF
      MF=0
      MND=0
      IF (N.GT.2) THEN
        ND=2
        MF=N-2
      ELSE
        ND=N
      ENDIF
      XTOL=TOL/100.0D0
      REL=DMAX1(ETOL,XTOL)
      IF(ND.EQ.1) THEN
C       MF=0 HERE
C       SERIES FOR YI1=G1 AND Y1=E1 FOR N=1 SCALED BY KODE
        YI1=G1FUN(X,KODE,REL,ETOL,Y1)
        EN(1)=Y1
        GN(1)=YI1
        IF(M.EQ.1) THEN
          RETURN
        ENDIF
        ND=2
        MND=1
      ENDIF
      IF(ND.EQ.2) THEN
C       SERIES FOR YI1=G2 AND Y1=E2 FOR N=2 SCALED BY KODE
        YI1=G2FUN(X,KODE,REL,ETOL,Y1)
        IF(MND.EQ.1) THEN
C         MF=0 HERE
          EN(2)=Y1
          GN(2)=YI1
          IF(M.EQ.2) THEN
            RETURN
          ENDIF
          AA=2.0D0
          IND=2
          MU=M-2
        ELSE
          IF(MF.EQ.0) THEN
            EN(1)=Y1
            GN(1)=YI1
            IF(M.EQ.1) THEN
              RETURN
            ENDIF
            IND=1
            MU=M-1
          ENDIF
          AA=2.0D0
        ENDIF
      ENDIF
      IF(MF.NE.0) THEN
        DO 10 K=1,MF
          Y2=(EMX-X*Y1)/AA
          Y1=Y2
          YI1=(Y2-X*YI1)/AA
          AA=AA+1.0D0
 10     CONTINUE
        EN(1)=Y1
        GN(1)=YI1
```

```
          IF(M.EQ.1) RETURN
          IND = 1
          MU = M-1
        ENDIF
        GOTO 240
   80 CONTINUE
        DO 90 I=1,M
          CC = 1.0D0/(FN-1.0D0+DBLE(FLOAT(I-1)))
          EN(I)=CC
          GN(I)=CC*CC
   90 CONTINUE
      RETURN
C-----------------------------------------------------------------------
C     BACKWARD RECURSIVE MILLER ALGORITHM FOR
C              E(N,X)=DEXP(-X)*(X**(N-1))*U(N,N,X)
C     WITH RECURSION AWAY FROM N=INTEGER CLOSEST TO X.
C     U(A,B,X) IS THE SECOND CONFLUENT HYPERGEOMETRIC FUNCTION
C-----------------------------------------------------------------------
  100 CONTINUE
      EMX = 1.0D0
      IF (KODE.EQ.2) GO TO 130
      IF (X.LE.XLIM) GO TO 120
  105 CONTINUE
      IF (N.EQ.1) THEN
        IERR=1
        RETURN
      ENDIF
      NZ = M
      DO 110 I=1,M
        EN(I) = 0.0D0
        GN(I) = 0.0D0
  110 CONTINUE
      RETURN
  120 EMX = DEXP(-X)
  130 CONTINUE
      MF=0
      RIX = ANINT(X+0.5D0)
      FKN = FN+DBLE(FLOAT(M-1))
      IF (FKN.LE.RIX) GO TO 140
      IF (FN.LT.RIX .AND. RIX.LT.FKN) GO TO 170
      IF (FN.GE.RIX) GO TO 160
      GO TO 340
  140 ICASE = 1
C     WHOLE SEQUENCE LIES TO THE LEFT OF RIX. START AT FKN, RECUR DOWN
C     ADJUST INDEX UP TO GET BETTER CONVERGENCE AND MORE ACCURACY.
C     RECUR DOWN MF TIMES (UNINDEXED) TO GET BACK TO ORIGINAL INDEX.
      FKS=FKN
      IF(FKN.LT.2.0D0) THEN
        FKS=2.0D0
        MF=INT(SNGL(2.0D0-FKN))
      ENDIF
      ML = M - 1
      MU = -1
      IND = M
      GOTO 180
  160 ICASE = 2
C     WHOLE SEQUENCE LIES TO THE RIGHT OF RIX. START AT FN, RECUR UP
      IND = 1
      FKS=FN
      MU = M - 1
      ML = -1
      GOTO 180
  170 CONTINUE
C     RIX LIES IN THE RANGE OF THE SEQUENCE. START AT RIX, RECUR UP &
C     DOWN
      IX=INT(SNGL(RIX))
      ICASE = 1
```

```
C      N = INT(FN)
      ML = IX - N
      IND = ML + 1
      FKS = FN+DBLE(FLOAT(IND-1))
      MU = M - IND
  180 CONTINUE
      AH = FKS/2.0D0
C-----------------------------------------------------------------------
C     START COMPUTATION FOR
C              EN(IND) = C*U( A , A ,X)
C-----------------------------------------------------------------------
      IC = 0
      AA = AH + AH
      AAMS = AA - 1.0D0
      AAMS = AAMS*AAMS
      TX = X + X
      FX = TX + TX
      AK = AH
      XTOL = TOL
      IF (TOL.LE.1.0D-3) XTOL = 20.0D0*TOL
      CT = AAMS + FX*AH
C-----------------------------------------------------------------------
C     PREVENT OVERFLOW IN DEN OF EM = (AH+1.0D0)/((X+AA)*XTOL*SQRT(CT))
C-----------------------------------------------------------------------
      P1 = (AH+1.0D0)/((X+AA)*XTOL)
      P2 = SQRT(CT)
      IF( P1 .LT. P2/XOVR ) THEN
        Y1 = 1.0D0/X
        GO TO 205
      ENDIF
      EM = P1/P2
      BK = AA
      CC = AH*AH
C-----------------------------------------------------------------------
C     FORWARD RECURSION FOR P(IC),P(IC+1) AND INDEX IC FOR BACKWARD
C     RECURSION
C-----------------------------------------------------------------------
      P1 = 0.0D0
      P2 = 1.0D0
  190 CONTINUE
      IF (IC.EQ.150) GO TO 340
      IC = IC + 1
      AK = AK + 1.0D0
      AT = BK/(BK+AK+CC+DBLE(FLOAT(IC)))
      BK = BK + AK + AK
      A(IC) = AT
      BT = (AK+AK+X)/(AK+1.0D0)
      B(IC) = BT
      PT = P2
      P2 = BT*P2 - AT*P1
      P1 = PT
      CT = CT + FX
      EM = EM*AT*(1.0D0-TX/CT)
      IF (EM*(AK+1.0D0).GT.P1*P1/PTOL) GOTO 190
      ICT = IC
      KK = IC + 1
      BT = TX/(CT+FX)
      Y2 = (BK/(BK+CC+DBLE(FLOAT(KK))))*(P1/P2)*(1.0D0-BT+0.375D0*BT*BT)
      Y1 = 1.0D0
      DICT=DBLE(FLOAT(KK))
      ANUM=AH+DICT
      ADEN=AA+DICT-1.0D0
      YT=Y2/(DICT)
      YI1=ANUM*YT/ADEN
      DKCT=DICT
C-----------------------------------------------------------------------
C     BACKWARD RECURRENCE FOR
```

```
C              Y1=             C*U( A ,A,X)
C              Y2= C*(A/(1+A/2))*U(A+1,A,X)
C-----------------------------------------------------------------------
      DO 200 K=1,ICT
        KK = KK - 1
        YT = Y1
        Y1 = (B(KK)*Y1-Y2)/A(KK)
        Y2 = YT
        DKCT=DKCT-1.0D0
        ANUM=AH+DKCT
        ADEN=AA+DKCT-1.0D0
        YI1=(ANUM/ADEN)*(Y2/DKCT+YI1)
  200 CONTINUE
C-----------------------------------------------------------------------
C     THE CONTIGUOUS RELATION
C              X*U(B,C+1,X)=(C-B)*U(B,C,X)+U(B-1,C,X)
C     WITH  B=A+1 , C=A IS USED FOR
C              Y(2) = C * U(A+1,A+1,X)
C     X IS INCORPORATED INTO THE NORMALIZING RELATION
C-----------------------------------------------------------------------
C      WRITE(7,203) RIX,DKCT,ICT
C 203  FORMAT(2D15.5,I5)
      PT = Y2/Y1
      ADEN = AA - PT*(AH+1.0D0)+X
      CNORM=1.0D0/ADEN
      YI1=YI1*CNORM/Y1
      Y1 = CNORM
  205 CONTINUE
      IF (MF.NE.0) THEN
        AK=FKS
        DO 206 J=1,MF
          AK=AK-1.0D0
          Y2=(1.0D0-AK*Y1)/X
          YI1=(Y1-AK*YI1)/X
          Y1=Y2
  206   CONTINUE
        FKS=AK
      ENDIF
      EN(IND) = EMX*Y1
      GN(IND) = EMX*YI1
      IF (M.EQ.1) RETURN
      AA = FKS
      GO TO (220, 240), ICASE
C-----------------------------------------------------------------------
C     RECURSION SECTION  FKH*E(FKH+1,X) + X*E(FKH,X)=EMX
C-----------------------------------------------------------------------
  220 K = IND-1
      IF (ML.LE.0) GOTO 235
      DO 230 I=1,ML
        AA = AA - 1.0D0
        EN(K) = (EMX-AA*EN(K+1))/X
        GN(K)= (EN(K+1)-AA*GN(K+1))/X
        K = K - 1
  230 CONTINUE
  235 IF (MU.LE.0) RETURN
      AA = FKS
  240 K = IND
      DO 250 I=1,MU
        EN(K+1) = (EMX-X*EN(K))/AA
        GN(K+1)= (EN(K+1)-X*GN(K))/AA
        AA = AA + 1.0D0
        K = K + 1
  250 CONTINUE
      RETURN
  340 CONTINUE
      IERR=2
      RETURN
```

```
      END
      DOUBLE PRECISION FUNCTION G1FUN(X,KODE,REL,ETOL,E1)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION B(100)
      DATA EULER /0.57721566490153286D0/
      DATA HPI   /1.5707963267948966D0/
      IFLAG=0
      TOL=REL/10.0D0
      IF(X.LT.ETOL) THEN
C       ELIMINATING UNDERFLOW WHEN X IS SMALL
        SUM0=-X
        SUM1=-X
        ASUM=0.0D0
        IFLAG=1
      ELSE
        AK=1.0D0
        TRM=1.0D0
        SUM0=0.0D0
        SUM1=0.0D0
        DO 10 K=1,100
          TRM=-TRM*X/AK
          TRM0=TRM/AK
          SUM0=SUM0+TRM0
          SUM1=SUM1+TRM0/AK
          IF(DABS(TRM).LT.TOL*DABS(SUM0)) GOTO 20
          AK=AK+1.0D0
  10    CONTINUE
        PRINT *,'IN G1FUN, DROP THRU LOOP 1'
        PAUSE
  20  CONTINUE
      ENDIF
      EPLNX=EULER+DLOG(X)
      E1=-EPLNX-SUM0
      S=HPI*HPI/3.0D0+0.5D0*E1*E1
      S=S-EPLNX*SUM0+SUM1
      IF(IFLAG.EQ.1) GOTO 45
      B(1)=1.0D0
      B(2)=1.0D0
      DO 25 J=3,100
        B(J)=0.0D0
  25  CONTINUE
C     HERE B VECTOR CONTAINS THE BINOMIAL COEFFICIENTS FOR (X+Y)**1
      FACK=2.0D0
      TRM=-X
      AK=2.0D0
      ASUM=0.0D0
      DO 30 K=2,99
        TRM=-TRM*X
C       GENERATE ROW K OF PASCAL TRIANGLE( BACKWARD RECURRENCE DOES NOT
C       OVERWRITE K-1 ENTRIES UNTIL THEY ARE NO LONGER NEEDED FOR THE
C       ROW K COMPUTATION)
        M=K
        DO 35 J=1,K
          B(M+1)=B(M)+B(M+1)
          M=M-1
  35    CONTINUE
C       B VECTOR = BINOMIAL COEFFICIENTS FOR (X+Y)**K
        SUM1=0.0D0
        AM=AK-1.0D0
        DO 40 J=2,K
          SUM1=SUM1+B(J)/AM
          AM=AM-1.0D0
  40    CONTINUE
        CS=SUM1/FACK
        TRM1=TRM*CS/AK
        ASUM=ASUM+TRM1
        IF(DABS(TRM1).LT.TOL*DABS(ASUM)) GOTO 45
```

```
        AK=AK+1.0D0
        FACK=FACK*AK
   30 CONTINUE
      PRINT *,'IN G1FUN, DROP THRU LOOP 2'
      PAUSE
   45 CONTINUE
      G1=S-ASUM
      IF(KODE.EQ.1) THEN
       G1FUN=G1
      ELSE
       TRM=DEXP(X)
       G1FUN=G1*TRM
       E1=E1*TRM
      ENDIF
      RETURN
      END
      DOUBLE PRECISION FUNCTION G2FUN(X,KODE,REL,ETOL,E2)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION B(100)
      DATA EULER /0.57721566490153286060/
      DATA HPI /1.5707963267948966D0/
      TOL=REL/10.0D0
      IFLAG=0
      IF(X.EQ.0.0D0) THEN
        G2FUN=1.0D0
        E2=1.0D0
        RETURN
      ENDIF
      IF(X.LT.ETOL) THEN
        SUM0=-0.5D0*X
        SUM2=0.0D0
        ASUM=0.0D0
        DSUM=0.0D0
        IFLAG=1
      ELSE
        AK=2.0D0
        AKM=1.0D0
        TRM=-X
        SUM0=0.5D0*TRM
        SUM2=0.0D0
        DO 10 K=2,100
          TRM=-TRM*X/AK
          CK=AK+1.0D0
          TRM0=TRM/(AK*AK)
          TRM0=TRM0/CK
          SUM0=SUM0+TRM0
          TRM2=TRM/AKM
          SUM2=SUM2+TRM2
          IF(DABS(TRM0).LT.TOL*DABS(SUM0)) GOTO 20
          AKM=AK
          AK=AK+1.0D0
   10   CONTINUE
        PRINT *,'IN G2FUN, DROP THRU SUM0 LOOP'
        PAUSE
   20   CONTINUE
      ENDIF
      EPLNX=EULER+DLOG(X)
      E2=X*(EPLNX-1.0D0)-SUM2+1.0D0
      S=-0.5D0*EPLNX*EPLNX+EPLNX-SUM0-HPI*HPI/3.0D0-1.0D0
      IF(IFLAG.EQ.1) GOTO 145
C     START SUMS WITH A(N) AND D(N) IN THEM
      B(1)=1.0D0
      B(2)=1.0D0
      DO 25 J=3,100
        B(J)=0.0D0
   25 CONTINUE
C     HERE B VECTOR CONTAINS THE BINOMIAL COEFFICIENTS FOR (X+Y)**1
```

```
      FACK=2.0D0
      TRM=X
      AK=2.0D0
      ASUM=0.0D0
      DSUM=0.0D0
      DO 130 K=2,99
        TRM=-TRM*X
C       GENERATE ROW K OF PASCAL TRIANGLE( BACKWARD RECURRENCE DOES NOT
C       OVERWRITE K-1 ENTRIES UNTIL THEY ARE NO LONGER NEEDED FOR THE
C       ROW K COMPUTATION)
        M=K
        DO 135 J=1,K
          B(M+1)=B(M)+B(M+1)
          M=M-1
  135   CONTINUE
C       B VECTOR (K+1 COMPONENTS) = BINOMIAL COEFFICIENTS FOR (X+Y)**K
        SUMA=0.0D0
        SUMD=0.0D0
        AM=AK-1.0D0
        BM=1.0D0
        DO 140 J=2,K
          TRM=B(J)/AM
          SUMA=SUMA+TRM
          SUMD=SUMD+TRM/BM
          AM=AM-1.0D0
          BM=BM+1.0D0
  140   CONTINUE
        CA=SUMA/FACK
        CD=SUMD/FACK
        CK=AK+1.0D0
        TRMA=TRM*CA/(AK*CK)
        TRMD=TRM*CD/CK
        ASUM=ASUM+TRMA
        DSUM=DSUM+TRMD
        IF(DABS(TRMD).LT.TOL*DABS(DSUM)) GOTO 145
        AK=AK+1.0D0
        FACK=FACK*AK
  130 CONTINUE
      PRINT *,'IN G2FUN, DROP THRU A(N),D(N) SUM LOOP'
      PAUSE
  145 CONTINUE
      S=S+ASUM-0.5D0*DSUM
      G2=1.0D0+X*S
      IF(KODE.EQ.1) THEN
        G2FUN=G2
      ELSE
        TRM=DEXP(X)
        G2FUN=G2*TRM
        E2=E2*TRM
      ENDIF
      RETURN
      END
      SUBROUTINE DGHEXINT(X, FNH, KODE, M, TOL, GN, EN, NZ, IERR)
C***BEGIN PROLOGUE  DGHEXINT
C***DATE WRITTEN   020304   (YYMMDD)
C***REVISION DATE  020425   (YYMMDD)
C***CATEGORY NO.  B5E
C***KEYWORDS  EXPONENTIAL INTEGRAL
C***AUTHOR  AMOS, DONALD E.
C***PURPOSE  DGHEXINT COMPUTES THE DERIVATIVE WRT ORDER OF THE
C***         EXPONENTIAL INTEGRAL AT HALF ODD ORDERS AS WELL AS THE
C***         EXPONENTIAL INTEGRAL ITSELF AT HALF ODD INTEGERS
C***DESCRIPTION
C
C                    * A DOUBLE PRECISION ROUTINE *
C         DGHEXINT COMPUTES M MEMBER SEQUENCES OF EXPONENTIAL INTEGRALS
C         E(FNH+K,X) AND G(FNH+K,X),  K=0,1,...,M-1 FOR HALF ODD INTEGER
```

```
C         ORDERS FNH.GE.0.5 AND X.GT.0.0D0. THE EXPONENTIAL INTEGRAL IS
C         DEFINED BY (IN THIS CONTEXT NU=0.5,1.5,2.5,...)
C
C         E(NU,X)=INTEGRAL ON (1,INFINITY) OF EXP(-XT)/T**NU, NU.GT.0
C
C         AND THE G INTEGRAL (=-DERIVATIVE WRT ORDER OF E AT NU) IS
C         DEFINED BY
C
C         G(NU,X)=INTEGRAL ON (1,INFINITY) OF E(N,XT)/T**NU
C
C         WHERE X=0.0 AND FNH=0.5 CANNOT OCCUR SIMULTANEOUSLY. FORMULAS
C         AND NOTATION ARE FOUND IN THE NBS HANDBOOK OF MATHEMATICAL
C         FUNCTIONS (REF. 1). THIS IS AN ADAPTATION OF REF. 2 TO HALF
C         ODD INTEGER ORDERS. SERIES FOR E(0.5,X) OR E(1.5,X) ALONG
C         WITH G(0.5,X) ARE COMPUTED FOR X.LE.2 AND FORWARD RECURRENCE
C         IS USED FOR HIGHER ORDERS. THE MILLER ALGORITHM IS RETAINED
C         FOR THE E FUNCTION AND THE DOCUMENTATION IN FOLDER 18 EXPLAINS
C         THE G COMPUTATION FOR X.GT.2.
C
C         INPUT     X , FNH, AND TOL ARE DOUBLE PRECISION
C           X       X.GT.0.0D0 FOR FNH=0.5 AND X.GE.0.0D0 FOR FNH.GE.1.5
C           FNH     ORDER OF FIRST MEMBER OF THE SEQUENCE, FNH.GE.0.5
C           KODE    A SELECTION PARAMETER FOR SCALED VALUES
C                   KODE=1   RETURNS E(FNH+K,X), G(FNH+K,X) K=0,1,...,M-1.
C                       =2   RETURNS EXP(X)*E(FNH+K,X), EXP(X)*G(FNH,X),
C                                       K=0,1,...,M-1.
C           M       NUMBER OF INTEGRALS IN THE E AND G SEQUENCES, M.GE.1
C           TOL     RELATIVE ACCURACY WANTED, ETOL.LE.TOL.LE.0.1D0
C                   ETOL = MAXIMUM OF DOUBLE PRECISION UNIT ROUNDOFF
C                   (D1MACH(4)) AND 1.0D-18
C
C         OUTPUT    EN IS A DOUBLE PRECISION VECTOR
C           EN      A VECTOR OF DIMENSION AT LEAST M CONTAINING VALUES
C                   EN(K) = E(FNH+K-1,X) OR EXP(X)*E(FNH+K-1,X), K=1,M
C                   DEPENDING ON KODE
C           GN      A VECTOR OF DIMENSION AT LEAST M CONTAINING VALUES
C                   GN(K) = G(FNH+K-1,X) OR EXP(X)*G(FNH+K-1,X), K=1,M
C                   DEPENDING ON KODE
C           NZ      UNDERFLOW INDICATOR
C                   NZ=0   A NORMAL RETURN
C                   NZ=M   X EXCEEDS XLIM AND AN UNDERFLOW OCCURS.
C                          EN(K)=GN(K)=0.0D0 , K=1,M RETURNED ON KODE=1
C           IERR    ERROR FLAG
C                   IERR=0, NORMAL RETURN, COMPUTATION COMPLETED
C                   IERR=1, INPUT ERROR,   NO COMPUTATION
C                   IERR=2, ERROR,         NO COMPUTATION
C                           ALGORITHM TERMINATION CONDITION NOT MET
C
C***REFERENCES  HANDBOOK OF MATHEMATICAL FUNCTIONS BY M. ABRAMOWITZ
C                 AND I. A. STEGUN, NBS AMS SERIES 55, U.S. DEPT. OF
C                 COMMERCE, 1955.
C
C               COMPUTATION OF EXPONENTIAL INTEGRALS BY D. E. AMOS,
C                 ACM TRANS. MATH SOFTWARE, 6, 1980,PP. 365-377, 420-428
C
C               FOLDER 18 BY D. E. AMOS EVALUATION OF E(NU,X), G(NU,X)=
C                 -DE(NU,X)/DNU FOR INTEGER AND HALF ODD INTEGER ORDERS
C                 WITH APPLICATION TO I(N,B,T)= INTEGRAL ON (T,INFINITY)
C                 OF EXP(-B^2*X^2)*LN(X)/X**N, N=1,2,3,...
C
C***ROUTINES CALLED  I1MACH,D1MACH,DRERF
C***END PROLOGUE  DGHEXINT
C      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
      DOUBLE PRECISION A, AA, AAMS, AH, AK, AT, B, BK, BT, CC, CNORM,
     * CT, EM, EMX, EN, ERF2, ETOL, FX, FN, FNH, FKN, PT, P1, P2, RIX,
     * RTZ, RTPI, TOL, TX, X, XCUT, XLIM, XTOL, YT, Y1, Y2, XOVR, PTOL
```

```
      DOUBLE PRECISION D1MACH, DRERF, GN, YI1, DICT, DKCT, ANUM, ADEN
      DOUBLE PRECISION DGERFC
      INTEGER I, IC, ICASE, ICT, IERR, IND, IX, I1M, K, KK, MF,
     * KODE, KODEP, M, ML, MU, N, NH, NZ, I1P
      INTEGER I1MACH
      DIMENSION EN(M), GN(M), A(150), B(150)
C
      DATA XCUT,PTOL  /     2.0D0,    1.0D+10     /
      DATA RTPI                  /1.772453850905516D0/
C
C***FIRST EXECUTABLE STATEMENT  DGHEXINT
      IERR=0
      NZ=0
      ETOL = DMAX1(D1MACH(4),0.5D-18)
      IF (X.LT.0.0D0) IERR=1
      IF (FNH.LT.0.5D0) IERR=1
      FN=FNH-0.5D0
      IF (FN-DBLE(FLOAT(INT(FN))).NE.0.0D0) IERR=1
      IF (KODE.LT.1 .OR. KODE.GT.2) IERR=1
      IF (M.LT.1) IERR=1
      IF (TOL.LT.ETOL .OR. TOL.GT.0.1D0) IERR=1
      IF (X.EQ.0.0D0 .AND. FNH.EQ.0.5D0) IERR=1
      IF (IERR.NE.0) RETURN
      I1M = -I1MACH(15)
      PT = 2.3026D0*DBLE(FLOAT(I1M))*D1MACH(5)
      XLIM = PT - 6.907755D0
      BT = PT + DBLE(FLOAT(N+M-1))
      IF (BT.GT.1000.0D0) XLIM = PT - DLOG(BT)
C-----------------------------------------------------------------------
C     SET OVERFLOW LIMIT ON X. THE UNDERFLOW LIMIT IS 1.0D0/XOVR.
C     TEST EXPONENTS ON SMALLEST AND LARGEST NUMBERS TO BE SURE THE
C     RECIPROCALS EXIST. EXPONENTS ARE I1MACH(15) AND I1MACH(16).
C-----------------------------------------------------------------------
      I1P = I1MACH(16)
      IF(I1M.LT.I1P) THEN
        XOVR = 1.0D-3/D1MACH(1)
      ELSE
        XOVR = 1.0D-3*D1MACH(2)
      ENDIF
      IF (X.GT.XOVR) GO TO 105
      IF (X.LT.1.0D0/XOVR) THEN
        IF (FNH.GT.0.5D0) GOTO 80
        IERR=1
        RETURN
      ENDIF
C-----------------------------------------------------------------------
C     START SEQUENCE AT G(X) (A=0.5D0) FOR X.LE.XCUT
C-----------------------------------------------------------------------
      IF (X.GT.XCUT) GOTO 100
      IF (FNH.GT.20.0D0) GOTO 100
      RTZ=DSQRT(X)
      IF (KODE.EQ.1) THEN
        EMX=DEXP(-X)
      ELSE
        EMX=1.0D0
      ENDIF
C     E(0.5,X)=RTPI*ERFC(RTX)/RTX  , RTX=SQRT(X)
      IF (FNH.EQ.0.5D0) THEN
        NZERO=0
        KODEP=KODE+1
        ERF2=DRERF(RTZ,KODEP,NZERO)
        CC=RTPI/RTZ
        Y1=CC*ERF2
        YT=DGERFC(RTZ,KODE,TOL,IERRG)
        YT=YT+YT
        YI1=YT*CC
        EN(1)=Y1
```

```
        GN(1)=YI1
        IF (M.EQ.1) RETURN
C        FNH=1.5 HERE E(3/2,X)=2*RTPI*IERFC(SQRT(X))
        CALL DIERFC(RTZ,KODE,YT,IERR)
        Y1=(YT+YT)*RTPI
        YI1=Y1-X*YI1
        YI1=YI1+YI1
        EN(2)=Y1
        GN(2)=YI1
        IF(M.EQ.2) RETURN
        IND = 2
        MU = M-2
        AA = 1.5D0
      ELSE
        CALL DIERFC(RTZ,KODE,YT,IERR)
        Y1=(YT+YT)*RTPI
        YT=DGERFC(RTZ,KODE,TOL,IERRG)
        YT=YT+YT
        CC=RTPI/RTZ
        YI1=YT*CC
        YI1=Y1-X*YI1
        YI1=YI1+YI1
        AA=1.5D0
        NH=INT(FN)-1
        IF (NH.LE.0) GOTO 11
        DO 10 K=1,NH
          Y2=(EMX-X*Y1)/AA
          Y1=Y2
          YI1=(Y2-X*YI1)/AA
          AA=AA+1.0D0
   10   CONTINUE
   11   CONTINUE
        EN(1)=Y1
        GN(1)=YI1
        IF(M.EQ.1) RETURN
        IND = 1
        MU = M-1
      ENDIF
      GOTO 240
   80 CONTINUE
      DO 90 I=1,M
        EN(I) = 1.0D0/(FNH-1.0D0+DBLE(FLOAT(I-1)))
        GN(I)=EN(I)*EN(I)
   90 CONTINUE
      RETURN
C-----------------------------------------------------------------------
C     BACKWARD RECURSIVE MILLER ALGORITHM FOR
C              E(N,X)=DEXP(-X)*(X**(N-1))*U(N,N,X)
C     WITH RECURSION AWAY FROM N=INTEGER CLOSEST TO X.
C     U(A,B,X) IS THE SECOND CONFLUENT HYPERGEOMETRIC FUNCTION
C-----------------------------------------------------------------------
  100 CONTINUE
      EMX = 1.0D0
      IF (KODE.EQ.2) GO TO 130
      IF (X.LE.XLIM) GO TO 120
  105 CONTINUE
      IF (FNH.EQ.0.5D0) THEN
        IERR=1
        RETURN
      ENDIF
      NZ = M
      DO 110 I=1,M
        EN(I) = 0.0D0
        GN(I) = 0.0D0
  110 CONTINUE
      RETURN
  120 EMX = DEXP(-X)
```

```
  130 CONTINUE
      MF=0
      RIX = ANINT(X)+0.5D0
      FKN = FNH+DBLE(FLOAT(M-1))
      IF (FKN.LE.RIX) GO TO 140
      IF (FNH.LT.RIX .AND. RIX.LT.FKN) GO TO 170
      IF (FNH.GE.RIX) GO TO 160
      GO TO 340
  140 ICASE = 1
C     WHOLE SEQUENCE LIES TO THE LEFT OF RIX. START AT FKN, RECUR DOWN
C     ADJUST INDEX UP TO GET BETTER CONVERGENCE AND MORE ACCURACY.
C     RECUR DOWN MF TIMES (UNINDEXED) TO GET BACK TO ORIGINAL INDEX.
      FKS=FKN
      IF(FKN.LT.2.5D0) THEN
        FKS=2.5D0
        MF=INT(SNGL(2.5D0-FKN))
      ENDIF
      ML = M - 1
      MU = -1
      IND = M
      GOTO 180
  160 ICASE = 2
C     WHOLE SEQUENCE LIES TO THE RIGHT OF RIX. START AT FNH, RECUR UP
      IND = 1
      FKS=FNH
      MU = M - 1
      ML = -1
      GOTO 180
  170 CONTINUE
C     RIX LIES IN THE RANGE OF THE SEQUENCE. START AT RIX, RECUR UP &
C     DOWN
      IX=INT(SNGL(RIX))
      ICASE = 1
      N = INT(FN)
      ML = IX - N
      IND = ML + 1
      FKS = FNH+DBLE(FLOAT(IND-1))
      MU = M - IND
  180 CONTINUE
      AH = FKS/2.0D0
C-----------------------------------------------------------------------
C     START COMPUTATION FOR
C              EN(IND) = C*U( A , A ,X)
C-----------------------------------------------------------------------
      IC = 0
      AA = AH + AH
      AAMS = AA - 1.0D0
      AAMS = AAMS*AAMS
      TX = X + X
      FX = TX + TX
      AK = AH
      XTOL = TOL
      IF (TOL.LE.1.0D-3) XTOL = 20.0D0*TOL
      CT = AAMS + FX*AH
C-----------------------------------------------------------------------
C     PREVENT OVERFLOW IN DEN OF EM = (AH+1.0D0)/((X+AA)*XTOL*SQRT(CT))
C-----------------------------------------------------------------------
      P1 = (AH+1.0D0)/((X+AA)*XTOL)
      P2 = SQRT(CT)
      IF( P1 .LT. P2/XOVR ) THEN
        Y1 = 1.0D0/X
        GO TO 205
      ENDIF
      EM = P1/P2
      BK = AA
      CC = AH*AH
C-----------------------------------------------------------------------
```

```
C     FORWARD RECURSION FOR P(IC),P(IC+1) AND INDEX IC FOR BACKWARD
C     RECURSION
C------------------------------------------------------------------------
      P1 = 0.0D0
      P2 = 1.0D0
  190 CONTINUE
      IF (IC.EQ.150) GO TO 340
      IC = IC + 1
      AK = AK + 1.0D0
      AT = BK/(BK+AK+CC+DBLE(FLOAT(IC)))
      BK = BK + AK + AK
      A(IC) = AT
      BT = (AK+AK+X)/(AK+1.0D0)
      B(IC) = BT
      PT = P2
      P2 = BT*P2 - AT*P1
      P1 = PT
      CT = CT + FX
      EM = EM*AT*(1.0D0-TX/CT)
      IF (EM*(AK+1.0D0).GT.P1*P1/PTOL) GOTO 190
      ICT = IC
      KK = IC + 1
      BT = TX/(CT+FX)
      Y2 = (BK/(BK+CC+DBLE(FLOAT(KK))))*(P1/P2)*(1.0D0-BT+0.375D0*BT*BT)
      Y1 = 1.0D0
      DICT=DBLE(FLOAT(KK))
      ANUM=AH+DICT
      ADEN=AA+DICT-1.0D0
      YT=Y2/(DICT)
      YI1=ANUM*YT/ADEN
      DKCT=DICT
C------------------------------------------------------------------------
C     BACKWARD RECURRENCE FOR
C              Y1=             C*U( A ,A,X)
C              Y2= C*(A/(1+A/2))*U(A+1,A,X)
C------------------------------------------------------------------------
      DO 200 K=1,ICT
        KK = KK - 1
        YT = Y1
        Y1 = (B(KK)*Y1-Y2)/A(KK)
        Y2 = YT
        DKCT=DKCT-1.0D0
        ANUM=AH+DKCT
        ADEN=AA+DKCT-1.0D0
        YI1=(ANUM/ADEN)*(Y2/DKCT+YI1)
  200 CONTINUE
C------------------------------------------------------------------------
C     THE CONTIGUOUS RELATION
C              X*U(B,C+1,X)=(C-B)*U(B,C,X)+U(B-1,C,X)
C     WITH  B=A+1 , C=A IS USED FOR
C              Y(2) = C * U(A+1,A+1,X)
C     X IS INCORPORATED INTO THE NORMALIZING RELATION
C------------------------------------------------------------------------
C      WRITE(7,203) RIX,DKCT,ICT
C 203  FORMAT(2D15.5,I5)
      PT = Y2/Y1
      ADEN = AA - PT*(AH+1.0D0)+X
      CNORM=1.0D0/ADEN
      YI1=YI1*CNORM/Y1
      Y1 = CNORM
  205 CONTINUE
      IF (MF.NE.0) THEN
        AK=FKS
        DO 206 J=1,MF
          AK=AK-1.0D0
          Y2=(1.0D0-AK*Y1)/X
          YI1=(Y1-AK*YI1)/X
```

```
          Y1=Y2
  206   CONTINUE
        FKS=AK
      ENDIF
      EN(IND) = EMX*Y1
      GN(IND)= EMX*YI1
      IF (M.EQ.1) RETURN
      AA = FKS
      GO TO (220, 240), ICASE
C-----------------------------------------------------------------------
C     RECURSION SECTION  FKH*E(FKH+1,X) + X*E(FKH,X)=EMX
C-----------------------------------------------------------------------
  220 K = IND-1
      IF (ML.LE.0) GOTO 235
      DO 230 I=1,ML
        AA = AA - 1.0D0
        EN(K) = (EMX-AA*EN(K+1))/X
        GN(K)= (EN(K+1)-AA*GN(K+1))/X
        K = K - 1
  230 CONTINUE
  235 IF (MU.LE.0) RETURN
      AA = FKS
  240 K = IND
      DO 250 I=1,MU
        EN(K+1) = (EMX-X*EN(K))/AA
        GN(K+1)= (EN(K+1)-X*GN(K))/AA
        AA = AA + 1.0D0
        K = K + 1
  250 CONTINUE
      RETURN
  340 CONTINUE
      IERR=2
      RETURN
      END
      DOUBLE PRECISION FUNCTION DGAMLN(Z,IERR)
C***BEGIN PROLOGUE  DGAMLN
C***DATE WRITTEN   830501   (YYMMDD)
C***REVISION DATE  830501   (YYMMDD)
C***CATEGORY NO.  B5F
C***KEYWORDS  GAMMA FUNCTION,LOGARITHM OF GAMMA FUNCTION
C***AUTHOR  AMOS, DONALD E., SANDIA NATIONAL LABORATORIES
C***PURPOSE  TO COMPUTE THE LOGARITHM OF THE GAMMA FUNCTION
C***DESCRIPTION
C
C               **** A DOUBLE PRECISION ROUTINE ****
C         DGAMLN COMPUTES THE NATURAL LOG OF THE GAMMA FUNCTION FOR
C         Z.GT.0.  THE ASYMPTOTIC EXPANSION IS USED TO GENERATE VALUES
C         GREATER THAN ZMIN WHICH ARE ADJUSTED BY THE RECURSION
C         G(Z+1)=Z*G(Z) FOR Z.LE.ZMIN.  THE FUNCTION WAS MADE AS
C         PORTABLE AS POSSIBLE BY COMPUTIMG ZMIN FROM THE NUMBER OF BASE
C         10 DIGITS IN A WORD, RLN=AMAX1(-ALOG10(R1MACH(4)),0.5E-18)
C         LIMITED TO 18 DIGITS OF (RELATIVE) ACCURACY.
C
C         SINCE INTEGER ARGUMENTS ARE COMMON, A TABLE LOOK UP ON 100
C         VALUES IS USED FOR SPEED OF EXECUTION.
C
C     DESCRIPTION OF ARGUMENTS
C
C         INPUT      Z IS D0UBLE PRECISION
C           Z      - ARGUMENT, Z.GT.0.0D0
C
C         OUTPUT      DGAMLN IS DOUBLE PRECISION
C           DGAMLN  - NATURAL LOG OF THE GAMMA FUNCTION AT Z.NE.0.0D0
C           IERR    - ERROR FLAG
C                     IERR=0, NORMAL RETURN, COMPUTATION COMPLETED
C                     IERR=1, Z.LE.0.0D0,    NO COMPUTATION
C
```

```
C
C***REFERENCES  COMPUTATION OF BESSEL FUNCTIONS OF COMPLEX ARGUMENT
C                 BY D. E. AMOS, SAND83-0083, MAY, 1983.
C***ROUTINES CALLED  I1MACH,D1MACH
C***END PROLOGUE  DGAMLN
      DOUBLE PRECISION CF, CON, FLN, FZ, GLN, RLN, S, TLG, TRM, TST,
     * T1, WDTOL, Z, ZDMY, ZINC, ZM, ZMIN, ZP, ZSQ, D1MACH
      INTEGER I, IERR, I1M, K, MZ, NZ, I1MACH
      DIMENSION CF(22), GLN(100)
C           LNGAMMA(N), N=1,100
      DATA GLN(1), GLN(2), GLN(3), GLN(4), GLN(5), GLN(6), GLN(7),
     1     GLN(8), GLN(9), GLN(10), GLN(11), GLN(12), GLN(13), GLN(14),
     2     GLN(15), GLN(16), GLN(17), GLN(18), GLN(19), GLN(20),
     3     GLN(21), GLN(22)/
     4     0.00000000000000000D+00,     0.00000000000000000D+00,
     5     6.93147180559945309D-01,     1.79175946922805500D+00,
     6     3.17805383034794562D+00,     4.78749174278204599D+00,
     7     6.57925121201010100D+00,     8.52516136106541430D+00,
     8     1.06046029027452502D+01,     1.28018274800814696D+01,
     9     1.51044125730755153D+01,     1.75023078458738858D+01,
     A     1.99872144956618861D+01,     2.25521638531234229D+01,
     B     2.51912211827386815D+01,     2.78992713838408916D+01,
     C     3.06718601060806728D+01,     3.35050734501368889D+01,
     D     3.63954452080330536D+01,     3.93398841871994940D+01,
     E     4.23356164607534850D+01,     4.53801388984769080D+01/
      DATA GLN(23), GLN(24), GLN(25), GLN(26), GLN(27), GLN(28),
     1     GLN(29), GLN(30), GLN(31), GLN(32), GLN(33), GLN(34),
     2     GLN(35), GLN(36), GLN(37), GLN(38), GLN(39), GLN(40),
     3     GLN(41), GLN(42), GLN(43), GLN(44)/
     4     4.84711813518352239D+01,     5.16066755677643736D+01,
     5     5.47847293981123192D+01,     5.80036052229805199D+01,
     6     6.12617017610020020D+01,     6.45575386270063311D+01,
     7     6.78897431371815350D+01,     7.12570389671680090D+01,
     8     7.46582363488301644D+01,     7.80922235533153106D+01,
     9     8.15579594561150372D+01,     8.50544670175815174D+01,
     A     8.85808275421976788D+01,     9.21361756036870925D+01,
     B     9.57196945421432025D+01,     9.93306124547874269D+01,
     C     1.02968198614513813D+02,     1.06631760260643459D+02,
     D     1.10320639714757395D+02,     1.14034211781461703D+02,
     E     1.17771881399745072D+02,     1.21533081515438634D+02/
      DATA GLN(45), GLN(46), GLN(47), GLN(48), GLN(49), GLN(50),
     1     GLN(51), GLN(52), GLN(53), GLN(54), GLN(55), GLN(56),
     2     GLN(57), GLN(58), GLN(59), GLN(60), GLN(61), GLN(62),
     3     GLN(63), GLN(64), GLN(65), GLN(66)/
     4     1.25317271149356895D+02,     1.29123933639127215D+02,
     5     1.32952575035616310D+02,     1.36802722637326368D+02,
     6     1.40673923648234259D+02,     1.44565743946344886D+02,
     7     1.48477766951773032D+02,     1.52409592584497358D+02,
     8     1.56360836303078785D+02,     1.60331128216630907D+02,
     9     1.64320112263195181D+02,     1.68327445448427652D+02,
     A     1.72352797139162802D+02,     1.76395848406997352D+02,
     B     1.80456291417543771D+02,     1.84533828861449491D+02,
     C     1.88628173423671591D+02,     1.92739047287844902D+02,
     D     1.96866181672889994D+02,     2.01009316399281527D+02,
     E     2.05168199482641199D+02,     2.09342586752536836D+02/
      DATA GLN(67), GLN(68), GLN(69), GLN(70), GLN(71), GLN(72),
     1     GLN(73), GLN(74), GLN(75), GLN(76), GLN(77), GLN(78),
     2     GLN(79), GLN(80), GLN(81), GLN(82), GLN(83), GLN(84),
     3     GLN(85), GLN(86), GLN(87), GLN(88)/
     4     2.13532241494563261D+02,     2.17736934113954227D+02,
     5     2.21956441819130334D+02,     2.26190548323727593D+02,
     6     2.30439043565776952D+02,     2.34701723442818268D+02,
     7     2.38978389561834323D+02,     2.43268849002982714D+02,
     8     2.47572914096186884D+02,     2.51890402209723194D+02,
     9     2.56221135550009525D+02,     2.60564940971863209D+02,
     A     2.64921649798552801D+02,     2.69291097651019823D+02,
     B     2.73673124285693704D+02,     2.78067573440366143D+02,
```

```
     C     2.82474292687630396D+02,     2.86893133295426994D+02,
     D     2.91323950094270308D+02,     2.95766601350760624D+02,
     E     3.00220948647014132D+02,     3.04686856765668715D+02/
      DATA GLN(89), GLN(90), GLN(91), GLN(92), GLN(93), GLN(94),
     1     GLN(95), GLN(96), GLN(97), GLN(98), GLN(99), GLN(100)/
     2     3.09164193580146922D+02,     3.13652829949879062D+02,
     3     3.18152639620209327D+02,     3.22663499126726177D+02,
     4     3.27185287703775217D+02,     3.31717887196928473D+02,
     5     3.36261181979198477D+02,     3.40815058870799018D+02,
     6     3.45379407062266854D+02,     3.49954118040770237D+02,
     7     3.54539085519440809D+02,     3.59134205369575399D+02/
C               COEFFICIENTS OF ASYMPTOTIC EXPANSION
      DATA CF(1), CF(2), CF(3), CF(4), CF(5), CF(6), CF(7), CF(8),
     1     CF(9), CF(10), CF(11), CF(12), CF(13), CF(14), CF(15),
     2     CF(16), CF(17), CF(18), CF(19), CF(20), CF(21), CF(22)/
     3     8.33333333333333333D-02,    -2.77777777777777778D-03,
     4     7.93650793650793651D-04,    -5.95238095238095238D-04,
     5     8.41750841750841751D-04,    -1.91752691752691753D-03,
     6     6.41025641025641026D-03,    -2.95506535947712418D-02,
     7     1.79644372368830573D-01,    -1.39243221690590112D+00,
     8     1.34028640441683920D+01,    -1.56848284626002017D+02,
     9     2.19310333333333333D+03,    -3.61087712537249894D+04,
     A     6.91472268851313067D+05,    -1.52382215394074162D+07,
     B     3.82900751391414141D+08,    -1.08822660357843911D+10,
     C     3.47320283765002252D+11,    -1.23696021422692745D+13,
     D     4.88788064793079335D+14,    -2.13203339609193739D+16/
C
C               LN(2*PI)
      DATA CON                    /     1.83787706640934548D+00/
C
C***FIRST EXECUTABLE STATEMENT  DGAMLN
      IERR=0
      IF (Z.LE.0.0D0) GO TO 70
      IF (Z.GT.101.0D0) GO TO 10
      NZ = INT(SNGL(Z))
      FZ = Z - FLOAT(NZ)
      IF (FZ.GT.0.0D0) GO TO 10
      IF (NZ.GT.100) GO TO 10
      DGAMLN = GLN(NZ)
      RETURN
   10 CONTINUE
      WDTOL = D1MACH(4)
      WDTOL = DMAX1(WDTOL,0.5D-18)
      I1M = I1MACH(14)
      RLN = D1MACH(5)*FLOAT(I1M)
      FLN = DMIN1(RLN,20.0D0)
      FLN = DMAX1(FLN,3.0D0)
      FLN = FLN - 3.0D0
      ZM = 1.8000D0 + 0.3875D0*FLN
      MZ = INT(SNGL(ZM)) + 1
      ZMIN = FLOAT(MZ)
      ZDMY = Z
      ZINC = 0.0D0
      IF (Z.GE.ZMIN) GO TO 20
      ZINC = ZMIN - FLOAT(NZ)
      ZDMY = Z + ZINC
   20 CONTINUE
      ZP = 1.0D0/ZDMY
      T1 = CF(1)*ZP
      S = T1
      IF (ZP.LT.WDTOL) GO TO 40
      ZSQ = ZP*ZP
      TST = T1*WDTOL
      DO 30 K=2,22
        ZP = ZP*ZSQ
        TRM = CF(K)*ZP
        IF (DABS(TRM).LT.TST) GO TO 40
```

```
        S = S + TRM
   30 CONTINUE
   40 CONTINUE
      IF (ZINC.NE.0.0D0) GO TO 50
      TLG = DLOG(Z)
      DGAMLN = Z*(TLG-1.0D0) + 0.5D0*(CON-TLG) + S
      RETURN
   50 CONTINUE
      ZP = 1.0D0
      NZ = INT(SNGL(ZINC))
      DO 60 I=1,NZ
        ZP = ZP*(Z+FLOAT(I-1))
   60 CONTINUE
      TLG = DLOG(ZDMY)
      DGAMLN = ZDMY*(TLG-1.0D0) - DLOG(ZP) + 0.5D0*(CON-TLG) + S
      RETURN
C
C
   70 CONTINUE
      IERR=1
      RETURN
      END
      SUBROUTINE DPSIFN(X, N, KODE, M, ANS, NZ, IERR)
C***BEGIN PROLOGUE  DPSIFN
C***DATE WRITTEN   820601   (YYMMDD)
C***REVISION DATE  820601   (YYMMDD)
C***CATEGORY NO.  B5F
C***KEYWORDS  PSI FUNCTION, GAMMA FUNCTION,
C             DERIVATIVES OF THE GAMMA FUNCTION
C***AUTHOR  AMOS, DONALD E., SANDIA NATIONAL LABORATORIES
C***PURPOSE  DPSIFN COMPUTES DERIVATIVES OF THE PSI FUNCTION
C***DESCRIPTION
C
C                   * A DOUBLE PRECISION ROUTINE *
C         DPSIFN COMPUTES M MEMBER SEQUENCES OF SCALED DERIVATIVES OF
C         THE PSI FUNCTION
C
C              W(K,X)=(-1)**(K+1)*PSI(K,X)/GAMMA(K+1)
C
C         K=N,...,N+M-1 WHERE PSI(K,X) IS THE K-TH DERIVATIVE OF THE PSI
C         FUNCTION.  ON KODE=1, DPSIFN RETURNS THE SCALED DERIVATIVES
C         AS DESCRIBED.  KODE=2 IS OPERATIVE ONLY WHEN K=0 AND DPSIFN
C         RETURNS -PSI(X) + LN(X).  THAT IS, THE LOGARITHMIC BEHAVIOR
C         FOR LARGE X IS REMOVED WHEN KODE=2 AND K=0. WHEN SUMS OR
C         DIFFERENCES OF PSI FUNCTIONS ARE COMPUTED, THE LOGARITHMIC
C         TERMS CAN BE COMBINED ANALYTICALLY AND COMPUTED SEFARATELY
C         TO HELP RETAIN SIGNIFICANT DIGITS. DEFINTIONS AND NOTAION ARE
C         FOUND IN THE NBS HANDBOOK OF MATHEMATICAL FUNCTIONS (REF. 1).
C
C         THE BASIC METHOD OF EVALUATION IS THE ASYMPTOTIC EXPANSION
C         FOR LARGE X.GE.XMIN FOLLOWED BY BACKWARD RECURSION ON A TWO
C         TERM RECURSION RELATION
C
C                  W(X+1) + X**(-N-1) = W(X).
C
C         THIS IS SUPPLEMENTED BY A SERIES
C
C                  SUM( (X+K)**(-N-1) , K=0,1,2,... )
C
C         WHICH CONVERGES RAPIDLY FOR LARGE N. BOTH XMIN AND THE
C         NUMBER OF TERMS OF THE SERIES ARE CALCULATED FROM THE UNIT
C         ROUND OFF OF THE MACHINE ENVIRONMENT.
C
C         THE NOMINAL COMPUTATIONAL ACCURACY IS THE MAXIMUM OF UNIT
C         ROUNDOFF (=D1MACH(4)) AND 1.0D-18 SINCE CRITICAL CONSTANTS
C         ARE GIVEN TO ONLY 18 DIGITS.
C
```

```
C         PSIFN IS THE SINGLE PRECISION VERSION OF DPSIFN.
C
C
C         INPUT      X IS DOUBLE PRECISION
C           X      - ARGUMENT, X.GT.0.0D0
C           N      - FIRST MEMBER OF THE SEQUENCE, 0.LE.N.LE.100
C                    N=0 GIVES ANS(1) = -PSI(X)       ON KODE=1
C                                       -PSI(X)+LN(X) ON KODE=2
C           KODE   - SELECTION PARAMETER
C                    KODE=1 RETURNS SCALED DERIVATIVES OF THE PSI
C                    FUNCTION
C                    KODE=2 RETURNS SCALED DERIVATIVES OF THE PSI
C                    FUNCTION EXCEPT WHEN N=0. IN THIS CASE,
C                    ANS(1) = -PSI(X) + LN(X) IS RETURNED.
C           M      - NUMBER OF MEMBERS OF THE SEQUENCE, M.GE.1
C
C         OUTPUT     ANS IS DOUBLE PRECISION
C           ANS    - A VECTOR OF LENGTH AT LEAST M WHOSE FIRST M
C                    COMPONENTS CONTAIN THE SEQUENCE OF  DERIVATIVES
C                    SCALED ACCORDING TO KODE
C           NZ     - UNDERFLOW FLAG
C                    NZ.EQ.0, A NORMAL RETURN
C                    NZ.NE.0, UNDERFLOW, LAST NZ COMPONENTS OF ANS SET
C                             TO ZERO, ANS(M-K+1)=0.0, K=1,...,NZ
C           IERR   - ERROR FLAG
C                    IERR=0, A NORMAL RETURN, COMPUTATION COMPLETED
C                    IERR=1, INPUT ERROR,     NO COMPUTATION
C                    IERR=2, OVERFLOW,        X TOO SMALL OR N+M-1 TOO
C                            LARGE OR BOTH
C                    IERR=3, ERROR,           N TOO LARGE. DIMENSIONED
C                            ARRAY TRMR(NMAX) IS NOT LARGE ENOUGH FOR N
C
C***REFERENCES  HANDBOOK OF MATHEMATICAL FUNCTIONS, AMS 55, NATIONAL
C                 BUREAU OF STANDARDS BY M. ABRAMOWITZ AND I. A.
C                 STEGUN, 1964, PP.258-260, EQTNS. 6.3.5, 6.3.18, 6.4.6,
C                 6.4.9, 6.4.10
C
C               A PORTABLE FORTRAN SUBROUTINE FOR DERIVATIVES OF THE
C                 PSI FUNCTION BY D. E. AMOS, ACM TRANS. MATH SOFTWARE,
C                 1983
C***ROUTINES CALLED  I1MACH,D1MACH
C***END PROLOGUE  DPSIFN
      INTEGER I, IERR, J, K, KODE, M, MM, MX, N, NMAX, NN, NP, NX, NZ
      INTEGER I1MACH
      DOUBLE PRECISION ANS, ARG, B, DEN, ELIM, EPS, FLN, FN, FNP, FNS,
     * FX, RLN, RXSQ, R1M4, R1M5, S, SLOPE, T, TA, TK, TOL, TOLS, TRM,
     * TRMR, TSS, TST, TT, T1, T2, WDTOL, X, XDMLN, XDMY, XINC, XLN,
     * XM, XMIN, XQ, YINT
      DOUBLE PRECISION D1MACH
      DIMENSION B(22), TRM(22), TRMR(100), ANS(M)
      DATA NMAX /100/
C-----------------------------------------------------------------------
C             BERNOULLI NUMBERS
C-----------------------------------------------------------------------
      DATA B(1), B(2), B(3), B(4), B(5), B(6), B(7), B(8), B(9), B(10),
     * B(11), B(12), B(13), B(14), B(15), B(16), B(17), B(18), B(19),
     * B(20), B(21), B(22) /1.00000000000000000D+00,
     * -5.00000000000000000D-01,1.66666666666666667D-01,
     * -3.33333333333333333D-02,2.38095238095238095D-02,
     * -3.33333333333333333D-02,7.57575757575757576D-02,
     * -2.53113553113553114D-01,1.16666666666666667D+00,
     * -7.09215686274509804D+00,5.49711779448621554D+01,
     * -5.29124242424242424D+02,6.19212318840579710D+03,
     * -8.65802531135531136D+04,1.42551716666666667D+06,
     * -2.72982310678160920D+07,6.01580873900642368D+08,
     * -1.51163157670921569D+10,4.29614643061166667D+11,
     * -1.37116552050883328D+13,4.88332318973593167D+14,
```

```
     * -1.92965793419400681D+16/
C
C
C***FIRST EXECUTABLE STATEMENT  DPSIFN
      IERR = 0
      NZ=0
      IF (X.LE.0.0D0) IERR=1
      IF (N.LT.0) IERR=1
      IF (KODE.LT.1 .OR. KODE.GT.2) IERR=1
      IF (M.LT.1) IERR=1
      IF (IERR.NE.0) RETURN
      MM=M
      NX = MIN0(-I1MACH(15),I1MACH(16))
      R1M5 = D1MACH(5)
      R1M4 = D1MACH(4)*0.5D0
      WDTOL = DMAX1(R1M4,0.5D-18)
C-----------------------------------------------------------------------
C     ELIM = APPROXIMATE EXPONENTIAL OVER AND UNDERFLOW LIMIT
C-----------------------------------------------------------------------
      ELIM = 2.302D0*(DBLE(FLOAT(NX))*R1M5-3.0D0)
      XLN = DLOG(X)
   41 CONTINUE
      NN = N + MM - 1
      FN = DBLE(FLOAT(NN))
      FNP = FN + 1.0D0
      T = FNP*XLN
C-----------------------------------------------------------------------
C     OVERFLOW AND UNDERFLOW TEST FOR SMALL AND LARGE X
C-----------------------------------------------------------------------
      IF (DABS(T).GT.ELIM) GO TO 290
      IF (X.LT.WDTOL) GO TO 260
C-----------------------------------------------------------------------
C     COMPUTE XMIN AND THE NUMBER OF TERMS OF THE SERIES, FLN+1
C-----------------------------------------------------------------------
      RLN = R1M5*DBLE(FLOAT(I1MACH(14)))
      RLN = DMIN1(RLN,18.06D0)
      FLN = DMAX1(RLN,3.0D0) - 3.0D0
      YINT = 3.50D0 + 0.40D0*FLN
      SLOPE = 0.21D0 + FLN*(0.0006038D0*FLN+0.008677D0)
      XM = YINT + SLOPE*FN
      MX = INT(SNGL(XM)) + 1
      XMIN = DBLE(FLOAT(MX))
      IF (N.EQ.0) GO TO 50
      XM = -2.302D0*RLN - DMIN1(0.0D0,XLN)
      FNS = DBLE(FLOAT(N))
      ARG = XM/FNS
      ARG = DMIN1(0.0D0,ARG)
      EPS = DEXP(ARG)
      XM = 1.0D0 - EPS
      IF (DABS(ARG).LT.1.0D-3) XM = -ARG
      FLN = X*XM/EPS
      XM = XMIN - X
      IF (XM.GT.7.0D0 .AND. FLN.LT.15.0D0) GO TO 200
   50 CONTINUE
      XDMY = X
      XDMLN = XLN
      XINC = 0.0D0
      IF (X.GE.XMIN) GO TO 60
      NX = INT(SNGL(X))
      XINC = XMIN - DBLE(FLOAT(NX))
      XDMY = X + XINC
      XDMLN = DLOG(XDMY)
   60 CONTINUE
C-----------------------------------------------------------------------
C     GENERATE W(N+MM-1,X) BY THE ASYMPTOTIC EXPANSION
C-----------------------------------------------------------------------
      T = FN*XDMLN
```

```
      T1 = XDMLN + XDMLN
      T2 = T + XDMLN
      TK = DMAX1(DABS(T),DABS(T1),DABS(T2))
      IF (TK.GT.ELIM) GO TO 380
      TSS = DEXP(-T)
      TT = 0.5D0/XDMY
      T1 = TT
      TST = WDTOL*TT
      IF (NN.NE.0) T1 = TT + 1.0D0/FN
      RXSQ = 1.0D0/(XDMY*XDMY)
      TA = 0.5D0*RXSQ
      T = FNP*TA
      S = T*B(3)
      IF (DABS(S).LT.TST) GO TO 80
      TK = 2.0D0
      DO 70 K=4,22
        T = T*((TK+FN+1.0D0)/(TK+1.0D0))*((TK+FN)/(TK+2.0D0))*RXSQ
        TRM(K) = T*B(K)
        IF (DABS(TRM(K)).LT.TST) GO TO 80
        S = S + TRM(K)
        TK = TK + 2.0D0
   70 CONTINUE
   80 CONTINUE
      S = (S+T1)*TSS
      IF (XINC.EQ.0.0D0) GO TO 100
C-----------------------------------------------------------------------
C     BACKWARD RECUR FROM XDMY TO X
C-----------------------------------------------------------------------
      NX = INT(SNGL(XINC))
      NP = NN + 1
      IF (NX.GT.NMAX) GO TO 390
      IF (NN.EQ.0) GO TO 160
      XM = XINC - 1.0D0
      FX = X + XM
C-----------------------------------------------------------------------
C     THIS LOOP SHOULD NOT BE CHANGED. FX IS ACCURATE WHEN X IS SMALL
C-----------------------------------------------------------------------
      DO 90 I=1,NX
        TRMR(I) = FX**(-NP)
        S = S + TRMR(I)
        XM = XM - 1.0D0
        FX = X + XM
   90 CONTINUE
  100 CONTINUE
      ANS(MM) = S
      IF (FN.EQ.0.0D0) GO TO 180
C-----------------------------------------------------------------------
C     GENERATE LOWER DERIVATIVES, J.LT.N+MM-1
C-----------------------------------------------------------------------
      IF (MM.EQ.1) RETURN
      DO 150 J=2,MM
        FNP = FN
        FN = FN - 1.0D0
        TSS = TSS*XDMY
        T1 = TT
        IF (FN.NE.0.0D0) T1 = TT + 1.0D0/FN
        T = FNP*TA
        S = T*B(3)
        IF (DABS(S).LT.TST) GO TO 120
        TK = 3.0D0 + FNP
        DO 110 K=4,22
          TRM(K) = TRM(K)*FNP/TK
          IF (DABS(TRM(K)).LT.TST) GO TO 120
          S = S + TRM(K)
          TK = TK + 2.0D0
  110   CONTINUE
  120   CONTINUE
```

```
        S = (S+T1)*TSS
        IF (XINC.EQ.0.0D0) GO TO 140
        IF (FN.EQ.0.0D0) GO TO 160
        XM = XINC - 1.0D0
        FX = X + XM
        DO 130 I=1,NX
          TRMR(I) = TRMR(I)*FX
          S = S + TRMR(I)
          XM = XM - 1.0D0
          FX = X + XM
  130   CONTINUE
  140   CONTINUE
        MX = MM - J + 1
        ANS(MX) = S
        IF (FN.EQ.0.0D0) GO TO 180
  150 CONTINUE
      RETURN
C-----------------------------------------------------------------------
C     RECURSION FOR N = 0
C-----------------------------------------------------------------------
  160 CONTINUE
      DO 170 I=1,NX
        S = S + 1.0D0/(X+DBLE(FLOAT(NX-I)))
  170 CONTINUE
  180 CONTINUE
      IF (KODE.EQ.2) GO TO 190
      ANS(1) = S - XDMLN
      RETURN
  190 CONTINUE
      IF (XDMY.EQ.X) RETURN
      XQ = XDMY/X
      ANS(1) = S - DLOG(XQ)
      RETURN
C-----------------------------------------------------------------------
C     COMPUTE BY SERIES (X+K)**(-(N+1)) , K=0,1,2,...
C-----------------------------------------------------------------------
  200 CONTINUE
      NN = INT(SNGL(FLN)) + 1
      NP = N + 1
      T1 = (FNS+1.0D0)*XLN
      T = DEXP(-T1)
      S = T
      DEN = X
      DO 210 I=1,NN
        DEN = DEN + 1.0D0
        TRM(I) = DEN**(-NP)
        S = S + TRM(I)
  210 CONTINUE
      ANS(1) = S
      IF (N.NE.0) GO TO 220
      IF (KODE.EQ.2) ANS(1) = S + XLN
  220 CONTINUE
      IF (MM.EQ.1) RETURN
C-----------------------------------------------------------------------
C     GENERATE HIGHER DERIVATIVES, J.GT.N
C-----------------------------------------------------------------------
      TOL = WDTOL/5.0D0
      DO 250 J=2,MM
        T = T/X
        S = T
        TOLS = T*TOL
        DEN = X
        DO 230 I=1,NN
          DEN = DEN + 1.0D0
          TRM(I) = TRM(I)/DEN
          S = S + TRM(I)
          IF (TRM(I).LT.TOLS) GO TO 240
```

```
  230   CONTINUE
  240   CONTINUE
        ANS(J) = S
  250 CONTINUE
      RETURN
C-----------------------------------------------------------------------
C     SMALL X.LT.UNIT ROUND OFF
C-----------------------------------------------------------------------
  260 CONTINUE
      ANS(1) = X**(-N-1)
      IF (MM.EQ.1) GO TO 280
      K = 1
      DO 270 I=2,MM
        ANS(K+1) = ANS(K)/X
        K = K + 1
  270 CONTINUE
  280 CONTINUE
      IF (N.NE.0) RETURN
      IF (KODE.EQ.2) ANS(1) = ANS(1) + XLN
      RETURN
  290 CONTINUE
      IF (T.GT.0.0D0) GO TO 380
      NZ=0
      IERR=2
      RETURN
  380 CONTINUE
      NZ=NZ+1
      ANS(MM)=0.0D0
      MM=MM-1
      IF (MM.EQ.0) RETURN
      GO TO 41
  390 CONTINUE
      NZ=0
      IERR=3
      RETURN
      END
      SUBROUTINE DIERFC(X,KODE,ANS,IERR)
C
C     DONALD E. AMOS  DECEMBER,2000; APRIL, 2002
C
C     DIERFC COMPUTES THE ITERATED COERROR FUNCTION ANS=IERFC(X) FOR
C     X.GE.0.0D0 FOR KODE=1 AND ANS=DEXP(X^2)*IERFC(X) FOR KODE=2
C-----------------------------------------------------------------------
C     A DOUBLE PRECISION ROUTINE
C-----------------------------------------------------------------------
C
C     ERROR CONDITIONS:
C
C     IERR=0 NORMAL RETURN, COMPUTATION COMPLETED
C     IERR=1 X IS NEGATIVE, OR KODE IS NOT 1 OR 2. ANS=0.0D0 RETURNED
C     IERR=2 UNDERFLOW, ANS=0.0D0 RETURNED
C
C     CALLS ROUTINES: DRERF
C-----------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION COE(22,3),COEF(24)
      DATA RTPI             /1.772453850905516027D0/
C     CHEBYSHEV COEFFIENTS FOR [1,3]
      DATA (COE(I,1),I=1,22)/
     1   .202344905364807570l8D+0  ,   .521042442637979244 58D-1  ,
```

```
     9   .41241005463317949716D-12 ,  -.57267334816262484620D-13 ,
     A   .77252252761786501244D-14 ,  -.10145811447810374360D-14 ,
     B   .12996661027298386093D-15 ,  -.16264168799487232138D-16 /
C     CHEBYSHEV COEFFIENTS FOR [3.0,5.5]
      DATA (COE(I,2), I=1,20)/
     1   .25921033022179837406D+0  ,   .11863530222703435905D-1  ,
     2  -.22456106528565330611D-2  ,   .36452384007591723187D-3  ,
     3  -.53326825393439572246D-4  ,   .71668700726266341359D-5  ,
     4  -.88930993054872490740D-6  ,   .10122521979082776356D-6  ,
     5  -.10319937364889814600D-7  ,   .88033499475257759951D-9  ,
     6  -.47579661094375305761D-10 ,  -.27339349067429062305D-11 ,
     7   .14647408524171149142D-11 ,  -.31955443360429115183D-12 ,
     8   .55559386991032515372D-13 ,  -.86214247220764385389D-14 ,
     9   .12445509652263484721D-14 ,  -.17053165852882875168D-15 ,
     A   .22435960748663718688D-16 ,  -.28550649356569489785D-17 /
C     CHEBYSHEV COEFFIENTS FOR [5.5,8.0]
      DATA (COE(I,3),I=1,16)/
     1   .27286438146173752156D+00 ,   .32425882603098985126D-02 ,
     2  -.42224100912591457365D-03 ,   .48123603005869450314D-04 ,
     3  -.50583642058403311588D-05 ,   .50182354871013814033D-06 ,
     4  -.47550893368040893857D-07 ,   .43318144700139924622D-08 ,
     5  -.38068269625179404945D-09 ,   .32311570263089790410D-10 ,
     6  -.26467861486267136191D-11 ,   .20860606373577721774D-12 ,
     7  -.15719568201565543872D-13 ,   .11189571252221398860D-14 ,
     8  -.73431174026315067217D-16 ,   .41944332960396830269D-17 /
C     COEFFICIENTS FOR ASYMPTOTIC EXPANSION, X > 8.0D0
      DATA (COEF(I),I=1,24)/
     1   .50000000000000000000D+0  ,    -.75000000000000000000D+0  ,
     2   .18750000000000000000D+1  ,    -.65625000000000000000D+1  ,
     3   .29531250000000000000D+2  ,    -.16242187500000000000D+3  ,
     4   .10557421875000000000D+4  ,    -.79180664062500000000D+4  ,
     5   .67303564453125000000D+5  ,    -.63938386230468750000D+6  ,
     6   .67135305541992187500D+7  ,    -.77205601373291015625D+8  ,
     7   .96507001716613769531D+9  ,    -.13028445231742858886D+11 ,
     8   .18891245586027145385D+12 ,    -.29281430658342075347D+13 ,
     9   .48314360586264424324D+14 ,    -.84550131025962742567D+15 ,
     A   .15641774239803107374D+17 ,    -.30501459767616059381D+18 ,
     B   .62527992523612921731D+19 ,    -.13443518392576778172D+21 ,
     C   .30247916383297750887D+22 ,    -.71082603500749714585D+23 /
      IERR=0
      IF(X.LT.0.0D0) THEN
        ANS=0.0D0
        IERR=1
        RETURN
      ENDIF
      IF ((KODE.LT.1).OR.(KODE.GT.2)) THEN
        ANS=0.0D0
        IERR=1
        RETURN
      ENDIF
      X2=X*X
      IF(KODE.EQ.1) THEN
        IF(X2.GT.675.0D0) THEN
          IERR=2
          ANS=0.0D0
          RETURN
        ENDIF
        DX=DEXP(-X2)
      ELSE
        DX=1.0D0
      ENDIF
      IF(X.LE.1.0D0) THEN
C       RECURRENCE FOR IERFC: IERFC(X)=EXP(-X*X)/SQRT(PI)-X*ERFC(X)
        KODEA=KODE+1
        NZ=0
        ERFA=DRERF(X,KODEA,NZ)
        ANS=DX/RTPI-X*ERFA
```

```
          RETURN
        ENDIF
        IF (X.LE.3.0D0) THEN
          APB=4.0D0
          BMA=2.0D0
          N=22
          CALL DCHBYS(APB,BMA,X,N,COE(1,1),SUM)
          ANS=DX*SUM/X2
          RETURN
        ENDIF
        IF (X.LE.5.5D0) THEN
          APB=8.50D0
          BMA=2.5D0
          N=20
          CALL DCHBYS(APB,BMA,X,N,COE(1,2),SUM)
          ANS=DX*SUM/X2
          RETURN
        ENDIF
        IF (X.LE.8.0D0) THEN
          APB=13.50D0
          BMA=2.5D0
          N=16
          CALL DCHBYS(APB,BMA,X,N,COE(1,3),SUM)
          ANS=DX*SUM/X2
          RETURN
        ENDIF
C       ASYMPTOTIC EXPANSION FOR IERFC(X) FOR   X > 8.0D0
        SS=COEF(1)
        P=1.0D0
        REL=0.50D-15
        M=25
        ATRM1=SS
        DO 30 I=2,M
          P=P/X2
          TRM2=COEF(I)*P
          ATRM2=DABS(TRM2)
          IF(ATRM2.LT.REL*DABS(SS)) GOTO 40
          IF(ATRM2.GE.ATRM1)  GOTO 40
          SS=SS+TRM2
          ATRM1=ATRM2
 30     CONTINUE
 40     CONTINUE
        SS=DX*SS/RTPI
        ANS=SS/X2
        RETURN
        END
        SUBROUTINE DCHBYS(APB,BMA,X,N,COEF,SUM)
C-----------------------------------------------------------------------
C       CHBYS SUMS A CHEBYSHEV SERIES WITH COEFFICIENTS COEF(I), I=1,N
C       ON [A,B] AT X. APB=A+B , BMA=B-A
C-----------------------------------------------------------------------
        IMPLICIT DOUBLE PRECISION (A-H,O-Z)
        DIMENSION COEF(N)
        T=(X+X-APB)/BMA
        TT=T+T
        F1=0.0D0
        F2=0.0D0
        NM=N-1
        DO 160 I=1,NM
          J=N-I+1
          TEMP=F1
          F1=TT*F1-F2+COEF(J)
          F2=TEMP
  160   CONTINUE
        SUM=T*F1-F2+COEF(1)
        RETURN
        END
```

```
      SUBROUTINE DINERFC(X,N0,KODE,N,REL,Y,NZ)
C-----------------------------------------------------------------------
C     WRITTEN BY DONALD E. AMOS
C     DATE: JULY, 2001
C
C     REFERENCES
C         AMOS, D.E. AND DANIEL, S.L., CDC 6600 CODES FOR THE ERROR
C         FUNCTION, CUMULATIVE NORMAL AND RELATED FUNCTIONS, SANDIA
C         NATIONAL LABORATORIES REPORT SC-DR-72 0918, DECEMBER, 1972
C
C         AMOS, D.E., BOUNDS ON ITERATED COERROR FUNCTIONS AND THEIR
C         RATIOS, MATH. COMP.,27, 1973, PP.413-427
C
C     ABSTRACT       * A DOUBLE PRECISION ROUTINE*
C         DINERFC COMPUTES N-ITERATED COERROR FUNCTIONS(REPEATED
C         INTEGRALS OF THE COERROR FUNCTION ERFC(X))
C
C           Y(K)= I(SUPER N0+K-1)ERFC(X)  K=1,2,...,N
C
C                     OR  SCALED  FUNCTIONS
C
C           Y(K)= I(SUPER N0+K-1)ERFC(X)*EXP(X**2), K=1,2,...,N
C
C         FOR X.GT.0. FOR X.GT.0, ACCURATE RATIOS ARE COMPUTED
C         WITH THE SEQUENCE NORMALIZED BY Y(-1)=2.*EXP(-X**2)/SQRT(PI)
C         FOR X.LT.0., STARTING VALUES USING 2.*EXP(-X**2)/SQRT(PI)
C         AND ERFC(X) ARE USED WITH FORWARD RECURSION ON
C
C                   2*(K+1)*Y(K+1)=-2*X*Y(K)+Y(K-1).
C
C         ERFC(X) FOR X.LT.0 IS COMPUTED FROM
C
C                      ERFC(X) = 2.0 - ERFC(-X)
C
C         WHERE ERFC(-X) IS COMPUTED FROM THE CODE FOR X.GT.0.
C
C     DESCRIPTION OF ARGUMENTS
C
C         INPUT            * A DOUBLE PRECISION ROUTINE*
C           X      - X, ARGUMENT, UNRESTRICTED
C           N0     - FIRST ITERATED COERROR FUNCTION DESIRED, N0.GE.0
C           KODE   - A SELECTION PARAMETER
C                    KODE=1  RETURNS Y(K)=I(SUPER N0+K-1)ERFC(X)
C                            FOR K=1,2,...N
C                    KODE=2  RETURNS
C                          Y(K)=I(SUPER N0+K-1)ERFC(X)           X.LE.0
C                              =I(SUPER N0+K-1)ERFC(X)*EXP(X**2) X.GT.0
C                          FOR K=1,2,...,N
C           N      - NUMBER OF FUNCTIONS IN THE SEQUENCE, N.GE.1
C           REL    - RELATIVE ERROR PARAMETER, REL=1.D-S FOR S
C                    SIGNIFICANT DIGITS, 2.LE.D.LE.18 AND
C                    REL.GT.D1MACH(4)
C
C         OUTPUT
C           Y      - A VECTOR OF DIMENSION N OR GREATER CONTAINING
C                    THE N VALUES REQUESTED, DEPENDING ON KODE
C           NZ     - NZ.EQ.0 - A NORMAL RETURN, COMPUTATION COMPLETED
C                    NZ.NE.0 - LAST NZ COMPONENTS OF Y UNDERFLOWED
C                              Y(K)=0.0, K=N-NZ+1,...,N RETURNED
C
C     ERROR CONDITIONS
C         IMPROPER INPUT ARGUMENT- A FATAL ERROR
C         UNDERFLOW FOR X.GT.0 AND KODE=1 - A NON-FATAL ERROR,Y(K)=0.0,
C         K=N-NZ+1,...,N RETURNED
C         PROCESS FAILS TO CONVERGE AFTER 25 ITERATIONS
C
C     INERFC USES SUBROUTINES DGAMLN,I1MACH,D1MACH,XERROR
```

```
C-----------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION Y(N),SR1(25),SR2(25)
      DATA TORP, MAXI /1.12837916709551D0, 24/
C
      IF(REL.GT.1.0D-2) GO TO 91
      IF(N.LT.1) GO TO 92
      IF(KODE.LT.1 .OR. KODE.GT.2) GO TO 93
      IF(N0.LT.0) GO TO 94
      NZ=0
      R1M4=D1MACH(4)
      R1M5=D1MACH(5)
      TOL=REL
      IF(REL.LT.R1M4) TOL=R1M4
      KK=MIN(IABS(I1MACH(15)),IABS(I1MACH(16)))
      AA=FLOAT(KK)*R1M5-3.0D0
      XLIM=SQRT(D1MACH(2)*1.0D-3)
      ELIM=2.303D0*AA
      IF(ABS(X).GT.XLIM) GO TO 95
      X2=X*X
      IF(X.LE.0.0D0) GO TO 80
      IFLAG=1
      CX=TORP
      IF(KODE.EQ.1) CX=CX*DEXP(-X2)
      XX=X
      NN=N
      NN0=N0
  100 CONTINUE
      NM1=NN-1
      NF=NN0+NN-1
      NSAVE=NF
      IF(NF.GT.24) GO TO 4
      IF(XX.GE.2.0D0) GO TO 4
      NF=25
    4 CONTINUE
      IF(X2.GT.ELIM .AND. KODE.EQ.1) GO TO 800
      SK=NF
      CK=SK
      SKP1=SK+1.0D0
      TSKP1=SKP1+SKP1
      EARG=DGAMLN(0.5D0*(SK+2.0D0),IERR)-DGAMLN(0.5D0*(SK+3.0D0),IERR)
      RK1=0.5D0*DEXP(EARG)
      FAC=TSKP1*RK1
      RSAVE=1.0D0/(XX+SQRT(X2+FAC*FAC))
      SR1(1)=RSAVE
      K=0
      SKM1=SK-1.0D0
      EARG=DGAMLN(0.5D0*(SKM1+2.0D0),IERR)-DGAMLN(0.5D0*(SKM1+3.0D0),IER
     *R)
      RK2=0.5D0*DEXP(EARG)
C
   10 CONTINUE
      K=K+1
      KP1=K+1
      CK=CK+1.0D0
      TSKP1=TSKP1+2.0D0
      IF(MOD(K,2).EQ.0)GO TO 13
C     K ODD
      RK2=RK2*CK/(CK+1.0D0)
      RK=RK2
      GO TO 15
C     K EVEN
   13 RK1=RK1*CK/(CK+1.0D0)
      RK=RK1
   15 CONTINUE
      FAC=TSKP1*RK
      SR2(1)=1.0D0/(XX+DSQRT(X2+FAC*FAC))
```

```
      IF(K.GT.MAXI) GO TO 90
C
      DO 12 J=1,K
        RA=SR2(J)/SR1(J)
        AD=K-J+1
        AKP1=SK+AD
        TAKP1=AKP1+AKP1
        SR2(J+1)=1.0D0/(XX+DSQRT(X2+TAKP1*RA))
   12 CONTINUE
      RBAR=0.5D0/(XX+AKP1*SR2(K))
      IF(ABS(SR2(KP1)-RBAR) .LE. TOL*RBAR) GO TO 30
      DO 14 I=1,KP1
        SR1(I)=SR2(I)
   14 CONTINUE
      GO TO 10
C
   30 CONTINUE
      AK=SK
      RK=SR2(KP1)
      IF(NF.EQ.NSAVE) GO TO 35
      NI=NF-NSAVE
      DO 31 I=1,NI
        RK=0.5D0/(XX+AK*RK)
        AK=AK-1.0D0
   31 CONTINUE
C-----------------------------------------------------------------------
C     GENERATING RATIOS FOR X .GE. 0 BY BACKWARD RECURRENCE
C-----------------------------------------------------------------------
   35 CONTINUE
      Y(NN)=RK
      IF(NN.EQ.1) GO TO 37
      DO 40 K=1,NM1
        RK=0.5D0/(XX+AK*RK)
        Y(NN-K)=RK
        AK=AK-1.0D0
   40 CONTINUE
C
   37 CONTINUE
C
      PROD=CX
      IF(NN0.EQ.0) GO TO 45
      DO 41 K=1,NN0
        RK=0.5D0/(XX+AK*RK)
        AK=AK-1.0D0
        PROD=PROD*RK
   41 CONTINUE
C
   45 CONTINUE
      DO 43 K=1,NN
        PROD=PROD*Y(K)
        Y(K)=PROD
   43 CONTINUE
      IF(IFLAG.EQ.2) GO TO 83
   44 CONTINUE
      KK=NN
      DO 46 K=1,NN
        IF(Y(KK).NE.0.0D0) GO TO 47
        NZ=NZ+1
        KK=KK-1
   46 CONTINUE
   47 CONTINUE
      RETURN
C-----------------------------------------------------------------------
C     NEGATIVE X VALUES
C-----------------------------------------------------------------------
   80 CONTINUE
      IFLAG=2
```

```
      XX=-X
      NN0=0
      NN=1
      KK=IABS(I1MACH(14))+1
      URLIM=2.303D0*DBLE(FLOAT(KK))*R1M5
      IF(X2.GT.URLIM) GO TO 86
      CX=TORP*DEXP(-X2)
      GO TO 100
C
   86 CONTINUE
      Y(1)=0.0D0
      CX=0.0D0
   83 CONTINUE
      Y2 = 2.0D0 - Y(1)
      Y1=CX
C-----------------------------------------------------------------------
C     FORWARD RECURSION
C-----------------------------------------------------------------------
      TX=X+X
      TWAN = 2.0D0
      IF(N0.EQ.0) GO TO 85
      DO 82 K=1,N0
        YS=Y2
        Y2=(-TX*Y2+Y1)/TWAN
        Y1=YS
        TWAN = TWAN+2.0D0
   82 CONTINUE
   85 CONTINUE
      Y(1)=Y2
      IF(N.EQ.1) RETURN
      DO 84 K=2,N
        YS=Y2
        Y2=(-TX*Y2+Y1)/TWAN
        Y(K)=Y2
        Y1=YS
        TWAN = TWAN+2.0D0
   84 CONTINUE
      RETURN
C
  800 CONTINUE
      NZ=N
      DO 805 I=1,N
        Y(I)=0.0D0
  805 CONTINUE
      RETURN
C
C
   90 CALL XERROR(' IN DINERFC, NO CONVERGENCE AFTER 25 ITERATIONS',47,
     & 1,2)
      RETURN
   91 CALL XERROR(' IN DINERFC, REL.GT.1.0D-2',26,1,2)
      RETURN
   92 CALL XERROR(' IN DINERFC, N IS ZERO OR NEGATIVE',34,1,2)
      RETURN
   93 CALL XERROR(' IN DINERFC, KODE NOT 1 OR 2',28,1,2)
      RETURN
   94 CALL XERROR(' IN DINERFC, N0 IS NEGATIVE',27,1,2)
      RETURN
   95 CONTINUE
      IF(KODE.EQ.1) GO TO 800
      CALL XERROR(' IN DINERFC, X**2 OVERFLOWS',27,1,2)
      RETURN
      END
      SUBROUTINE DQUAD8(DQFUN,INIT,X1,SIG,REL,X2,QANS,IERR)
C
C     A DOUBLE PRECISION ROUTINE
C
```

```
C     INFINITE INTERVALS
C-----------------------
C     DQUAD8 EVALUATES INFINITE INTEGRALS ON (X1,INF) BY SUMMING
C     QUADRATURES (WITH DGAUS8) OF LENGTH SIG STARTING AT X1. TESTS
C     FOR CONVERGENCE ARE MADE IN A RELATIVE ERROR TEST USING REL FOR
C     BOTH DGAUS8 AND DQUAD8. THE TRUNCATION ERROR TEST FOR DQAUD8 IS
C
C     DABS(LAST QUADRATURE OF LENGTH SIG).LE.REL*DABS(ACCUMULATED SUM)
C
C     THE QUADRATURE ON (X1,X2) IS RETURNED IN QANS WITH AN ERROR
C     INDICATOR IERR. IERR=1 IS NORMAL AND IERR=5 MEANS THAT THE
C     CONVERGENCE OF THE INTEGRAL WAS NOT OBTAINED IN 20 SIG STEPS.
C     OTHER VALUES OF IERR ARE DEFINED IN THE PROLOGUE OF DGAUS8. X2
C     CAN BE USED TO COMPUTE A RIGOROUS TRUNCATION ESTIMATE IN THE
C     CALLING PROGRAM IF ONE IS KNOWN. IF THIS TRUNCATION ESTIMATE IS
C     NOT SATISFIED, THEN DQUAD8 CAN BE CALLED AGAIN WITH NO CHNAGE IN
C     THE CALL LIST.
C
C     INIT=0 ON THE FIRST CALL. IF DQUAD8 INDICATES THAT X2 IS TOO
C     SMALL (IERR=5) OR ONE WANTS MORE ASSURANCE THAT THE ANSWER IS
C     ACCURATE, THEN DQUAD8 CAN BE CALLED AGAIN WITH NO CHANGES IN
C     PARAMETERS. AFTER THE INITIAL CALL, INIT = THE NUMBER OF SIG
C     STEPS. ON SUBSEQUENT CALLS, INIT.NE.0 SIGNALS A CONTINUATION OF
C     THE STEPPING PROCESS UNTIL THE TRUNCATION ERROR CRITERION IS MET
C     AGAIN, ETC. SIG CAN BE CHANGED ON SUCCESSIVE CALLS IF DESIRED.
C     MOST LIKELY, DQUAD8'S TRUNCATION TEST WILL BE MET IN ONE
C     QUADRATURE STEP ON EACH CALL AFTER THE FIRST IF IERR=1 WAS
C     RETURNED ON THE PREVIOUS CALL.(I.E. HAVING SEEN CONVERGENCE ONCE,
C     DQUAD8 WILL SEE CONVERGENCE THEREAFTER ON EACH SIG STEP.)
C
C     FINITE INTERVALS
C---------------------
C     FOR MQ QUADRATURES ON A FINITE INTERVAL (A,B) SET THE PARAMETERS
C     AS FOLLOWS:
C                INIT=-MQ
C                X1=A
C                SIG=(B-A)/DBLE(FLOAT(MQ))
C     IN THIS CASE, IERR=5 CANNOT OCCUR SINCE THE NUMBER OF QUADRATURES
C     IS PREDETERMINED. ANY OTHER VALUE OF IERR SIGNALS AN ERROR ON SOME
C     INTERVAL FROM DGAUS8.
C
C     CODING THE INTEGRAND
C-------------------------
C     A DOUBLE PRECISION FUNCTION NAME FOR THE INTEGRAND IS SUPPLIED AS
C     THE FIRST ARGUMENT IN THE CALL TO DQUAD8. (DQFUN IS THE PLACE
C     HOLDER NAME FOR THIS FUNCTION). THIS FUNCTION, WHICH COMPUTES THE
C     INTEGRAND, MUST BE DECLARED EXTERNAL IN THE CALLING PROGRAM.
C
C     PARAMETERS FOR THE INTEGRAND ARE SUPPLIED IN A COMMMON STATEMENT.
C
C-------------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DATA MCALLS/20/
      EXTERNAL DQFUN
      IERR=1
      IFLAG=0
      IF (INIT.EQ.0) THEN
        XA=X1
        S=0.0D0
        SS=0.0D0
        JMAX=MCALLS
      ELSE
        IF(INIT.LT.0) THEN
          XA=X1
          S=0.0D0
          SS=0.0D0
          JMAX=IABS(INIT)
```

```
          IFLAG=1
          INIT=JMAX
        ELSE
          XA=X2
          S=0.0D0
          SS=QANS
          JMAX=MCALLS
        ENDIF
      ENDIF
      XB=XA+SIG
      DO 35 J=1,JMAX
        CREL=REL
        CALL DGAUS8(DQFUN,XA,XB,CREL,ANS,KERR)
        IF(KERR.NE.1) THEN
          IERR=KERR
        ENDIF
        S=S+ANS
        IF(IFLAG.EQ.0) THEN
          INIT=INIT+1
          IF (DABS(ANS).LE.REL*DABS(S+SS)) THEN
            X2=XB
            GOTO 36
          ENDIF
        ENDIF
        XA=XB
        XB=XA+SIG
   35 CONTINUE
      X2=XA
      IF(IFLAG.EQ.0) THEN
        IF(KERR.EQ.1) THEN
          IERR=5
        ENDIF
      ENDIF
   36 CONTINUE
      QANS=S+SS
      RETURN
      END

      DOUBLE PRECISION FUNCTION DFERF(X,REL,IERR)
C
C     DONALD E. AMOS, JANUARY, 2001
C
C-----------------------------------------------------------------------
C     A DOUBLE PRECISION ROUTINE
C-----------------------------------------------------------------------
C
C     REFERENCES: FOLDER 16, FOLDER 11, FOLDER 6
C
C     DFERF COMPUTES
C
C                 DFERF = INT ON (0,X] OF ERF(W)/W
C
C     BY THE POWER SERIES FOR SMALL X.LE.1.0D0 AND
C
C                 DFERF = DGERFC(X)+0.5D0*EULER+DLOG(2*X)
C
C     FOR X.GT.1.0D0 WHERE DGERFC IS THE INTEGRAL OF ERFC(W)/W ON
C     [X,INF) AND EULER IS THE EULER CONSTANT = 0.5772156649015329D0
C
C     REL SHOULD SATISFY 0.5D-15 .LE. REL .LE. 0.5D-6.
C-----------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DATA EULER /0.577215664901532861D0/
      DATA RTPI  /1.77245385090551602 7D0/
      IERR=0
      IF(X.LE.0.0D0)THEN
        IERR=1
```

```
          DFERF=0.0D0
          RETURN
        ENDIF
        IF(REL.GT.0.5D-3) THEN
          IERR=1
          REL=0.5D-6
        ENDIF
        IF (REL.LT.0.5D-15) THEN
          IERR=1
          REL=0.5D-15
        ENDIF
        X2=X*X
        IF (X.LE.1.0D0) THEN
C-----------------------------------------------------------------------
C         POWER SERIES FOR ERF(X)/X ON [0,1]
C-----------------------------------------------------------------------
          S=1.0D0
          AK=1.0D0
          BK=3.0D0
          CK=-X2
          DO 10 K=1,50
            TRM=CK/(BK*BK)
            S=S+TRM
            IF (DABS(TRM).LT.REL*DABS(S)) GOTO 11
            AK=AK+1.0D0
            BK=BK+2.0D0
            CK=-CK*X2/AK
   10     CONTINUE
   11     CONTINUE
          SUM=(S+S)*X/RTPI
          DFERF=SUM
          RETURN
        ENDIF
C-----------------------------------------------------------------------
C       DFERF=DGERFC(X)+0.5D0*EULER+DLOG(2*X)
C-----------------------------------------------------------------------
        KODE=1
        DFERF=DGERFC(X,KODE,REL,IERR)+0.5D0*EULER+DLOG(X+X)
        RETURN
        END
        DOUBLE PRECISION FUNCTION DGERFC(X,KODE,REL,IERR)
C
C       DONALD E. AMOS, JANUARY, 2001
C
C-----------------------------------------------------------------------
C       A DOUBLE PRECISION ROUTINE
C-----------------------------------------------------------------------
C
C       REFERENCES: FOLDER 6, FOLDER 11, FOLDER 16
C
C       ON KODE=1, DGERFC COMPUTES
C
C                DGERFC= INTEGRAL ON  [X,INF) OF ERFC(W)/W.
C
C       ON KODE=2, THIS RESULT IS SCALED BY DEXP(X**2).
C       REL IS THE RELATIVE ERROR DESIRED, 0.5D-14 .LE. REL .LE. 0.5D-3
C
C       IERR=0 IS A NORMAL RETURN
C       IERR=1 IS AN INPUT ERROR
C       IERR=2 IS AN UNDERFLOW ON KODE=1 WITH DGERFC=0.0D0 RETURNED
C
C       THE SERIES FOR
C
C                   DFERF=  INTEGRAL ON [0,1] OF ERF(W)/W
C       ALONG WITH
C                   DGERFC=DFERF-0.5D0*EULER-DLOG(2.0D0*X)
C
```

```
C     ARE USED FOR SMALL X.LE.1.125D0 AND THE ASYMPTOTIC EXPANSION IS USED
C     FOR X.GE.7.0D0. CHEBYSHEV EXPANSIONS FOR (X**3)*EXP(X**2) ARE
C     EVALUATED ELSEWHERE WITH RELATIVE ERRORS 0.5D-12 OR BETTER. EULER
C     IS THE EULER CONSTANT = 0.5772156649015328610D0
C
C     REL SHOULD SATISFY 0.5D-15 .LE. REL .LE. 0.5D-6.
C-----------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION COE(24,3),ENH(24)
C     CHEBYSHEV COEFFICIENTS FOR INTERVAL (1,3]
      DATA (COE(I,1), I=1,24)/
     1      0.183936201622926800D+00,      0.602481282921201200D-01,
     2     -0.119525542040734300D-01,      0.163048183862326200D-02,
     3     -0.916761486984451800D-04,     -0.300245124943356100D-04,
     4      0.136020773297374100D-04,     -0.357151573002627300D-05,
     5      0.743773093967212500D-06,     -0.131331376113140100D-06,
     6      0.196950958343117800D-07,     -0.233463166825802300D-08,
     7      0.141611588786531800D-09,      0.304836446286498300D-10,
     8     -0.158873383783807400D-10,      0.471064076171147000D-11,
     9     -0.117254478451858600D-11,      0.268672719869292500D-12,
     A     -0.589487969074253200D-13,      0.126554771091031300D-13,
     B     -0.269424003935865400D-14,      0.573660397938811300D-15,
     C     -0.122809738646140600D-15,      0.265136030774818600D-16/
C     CHEBYSHEV COEFFICIENTS FOR INTERVAL (3,5]
      DATA (COE(I,2), I=1,18)/
     1      0.249911966143169100D+00,      0.137407615980377800D-01,
     2     -0.211076375427525400D-02,      0.274149770820318000D-03,
     3     -0.315160070674318100D-04,      0.324067726462838300D-05,
     4     -0.293779756592443700D-06,      0.221272952269653000D-07,
     5     -0.108159106183739300D-08,     -0.393712006838136900D-10,
     6      0.211188488938613800D-10,     -0.394325155773269300D-11,
     7      0.574358099846202900D-12,     -0.736872227670002800D-13,
     8      0.868815019030334600D-14,     -0.959589743808029100D-15,
     9      0.100178090811985800D-15,     -0.991516948990920300D-17/
C     CHEBYSHEV COEFFICIENTS FOR INTERVAL (5,7)
      DATA (COE(I,3), I=1,14)/
     1      0.267032022166309300D+00,      0.465764615292050200D-02,
     2     -0.529406889059992700D-03,      0.522543466857855500D-04,
     3     -0.471680853238733800D-05,      0.398010531483489200D-06,
     4     -0.317099625639186200D-07,      0.239340951085099200D-08,
     5     -0.170820332342142600D-09,      0.114336576815315400D-10,
     6     -0.703928286525811000D-12,      0.380529299822365100D-13,
     7     -0.156156454824444000D-14,      0.110122861980293900D-16/
      DATA EULER /0.5772156649015328610D0/
      DATA RTPI  /1.772453850905516027D0/
      IERR=0
      IF(X.LE.0.0D0)THEN
        IERR=1
        DGERFC=0.0D0
        RETURN
      ENDIF
      IF((KODE.NE.1).AND.(KODE.NE.2))THEN
        IERR=1
        DGERFC=0.0D0
        RETURN
      ENDIF
      IF(REL.GT.0.5D-3) THEN
        IERR=1
        REL=0.5D-6
      ENDIF
      IF (REL.LT.0.5D-15) THEN
        IERR=1
        REL=0.5D-15
      ENDIF
      X2=X*X
      IF (X.LE.1.0D0) THEN
C-----------------------------------------------------------------------
```

```
C       DGERFC=POWER SERIES FOR ERF(X)/X-0.5*EULER-LOG(2*X) ON (0,1]
C-----------------------------------------------------------------------
        S=1.0D0
        AK=1.0D0
        BK=3.0D0
        CK=-X2
        DO 10 K=1,50
          TRM=CK/(BK*BK)
          S=S+TRM
          IF (DABS(TRM).LT.REL*DABS(S)) GOTO 11
          AK=AK+1.0D0
          BK=BK+2.0D0
          CK=-CK*X2/AK
  10    CONTINUE
  11    CONTINUE
        SUM=(S+S)*X/RTPI
        DGERFC=SUM-0.5D0*EULER-DLOG(X+X)
        IF(KODE.EQ.2) THEN
          DGERFC=DGERFC*DEXP(X2)
        ENDIF
        RETURN
      ENDIF
C-----------------------------------------------------------------------
C     CHEBYSHEV EXPANSIONS ON INTERVALS OF LENGTH 2 ON (1,7)
C-----------------------------------------------------------------------
      IF (X.LE.3.0D0) THEN
        N=24
        BMA=2.0D0
        APB=4.0D0
        CALL DCHBYS(APB,BMA,X,N,COE(1,1),SUM)
        X3=X2*X
        IF (KODE.EQ.1) THEN
          CC=DEXP(-X2)
        ELSE
          CC=1.0D0
        ENDIF
        DGERFC=SUM*CC/X3
        RETURN
      ENDIF
      IF (X.LE.5.0D0) THEN
        N=18
        BMA=2.0D0
        APB=8.0D0
        CALL DCHBYS(APB,BMA,X,N,COE(1,2),SUM)
        X3=X2*X
        IF (KODE.EQ.1) THEN
          CC=DEXP(-X2)
        ELSE
          CC=1.0D0
        ENDIF
        DGERFC=SUM*CC/X3
        RETURN
      ENDIF
      IF (X.LT.7.0D0) THEN
        N=14
        BMA=2.0D0
        APB=12.0D0
        CALL DCHBYS(APB,BMA,X,N,COE(1,3),SUM)
        X3=X2*X
        IF (KODE.EQ.1) THEN
          CC=DEXP(-X2)
        ELSE
          CC=1.0D0
        ENDIF
        DGERFC=SUM*CC/X3
        RETURN
      ENDIF
```

```
C-----------------------------------------------------------------------
C     ASYMPTOTIC EXPANSION FOR X.GE.7.0D0
C-----------------------------------------------------------------------
C     ENH(*) CONTAINS THE EXPONENTIAL INTEGRALS OF HALF ODD ORDERS.
C     THEY ARE USED IN THE ASYMPTOTIC SERIES WITH REL=0.5D-14 FOR THE
C     HIGHEST POSSIBLE ACCURACY IN DOUBLE PRECISION ARITHMETIC. THE
C     EXPONENTIAL INTEGRALS ENH(K)=E[HK+(K-1),ARG], K=1,MM STARTING
C     AT HALF ODD ORDER HK ARE USED AND ARE COMPUTED BY
C
C          CALL DHEXINT(ARG, FNH, KODE, MM, TOL, ENH, NZ, IERR)
C
C     WHERE  FNH=HK, KODE=1, TOL=REL. THE RETURN VARIABLES ARE ENH, NZ,
C     AND IERR. NZ=NUMBER OF TAIL VALUES IN THE SEQUENCE SET TO ZERO DUE
C     TO UNDERFLOW, ENH(MM-J+1)=0.0D0, J=1,NZ AND IERR= ERROR INDICATOR.
C     IERR=0 IS A NORMAL RETURN. KODE=2 RETURNS ENH(K)*DEXP(ARG), K=1,MM
C
C     IN THIS CONTEXT, ARG=X^2, FNH=HK=0.5D0, KODE=1, MM=20, TOL=REL.
C-----------------------------------------------------------------------
      IF(KODE.EQ.1) THEN
        IF (X2.GT.667.0D0) THEN
          DGERFC=0.0D0
          IERR=2
          RETURN
        ENDIF
      ENDIF
      ARG=X2
      FNH=0.5D0
      MM=24
      TOL=REL
      CALL DHEXINT(ARG, FNH, KODE, MM, TOL, ENH, NZ, IERR)
      IF (KODE.EQ.1) THEN
        IF(NZ.NE.0) THEN
          DGERFC=0.0D0
          IERR=2
          RETURN
        ENDIF
      ENDIF
      S=ENH(2)
      TRM1=S
      DTRM1=DABS(TRM1)
      AK=0.5D0
      CK=-AK/X2
      DO 20 K=1,MM-2
        TRM2=CK*ENH(K+2)
        DTRM2=DABS(TRM2)
        IF (DTRM2.LE.REL*DABS(S)) GOTO 22
        IF (DTRM2.GE.DTRM1) GOTO 21
        S=S+TRM2
        TRM1=TRM2
        AK=AK+1.0D0
        CK=-CK*AK/X2
  20  CONTINUE
C      WRITE(7,301)
C  301 FORMAT(" DROP THROUGH THE LOOP IN THE ASYMPTOTIC SERIES" )
  21  CONTINUE
C      WRITE (7,302)
C  302 FORMAT('REQUESTED ERROR NOT MET IN ASYMPTOTIC SERIES')
  22  CONTINUE
C      WRITE(7,303) K,REL
C  303 FORMAT (I5,D13.4)
      DGERFC=0.5D0*S/(X*RTPI)
      RETURN
      END
      DOUBLE PRECISION FUNCTION DHERFC(X)
C
C     DONALD E. AMOS, OCTOBER, 2002
C-----------------------------------------------------------------------
```

```
C     A DOUBLE PRECISION ROUTINE
C-----------------------------------------------------------------------
C     REFS: FOLDER 23,
C
C           C. W. CLENSHAW, CHEBYSHEV SERIES FOR MATHEMATICAL FUNCTIONS,
C           NATIONAL PHYSICAL LABORATORY MATHEMATICAL TABLES, VOLUME 5,
C           HER MAJESTY'S STATIONERY OFFICE, LONDON, 1963, TABLE 9
C
C
C     DHERFC COMPUTES
C
C              DHERFC=INT ON (0,X) OF EXP(W*W)*ERFC(W)
C
C-----------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION Y(50),AR(18)
      DATA RTPI /1.772453850905516D0/
C
C     CHEBYSHEV COEFFICIENTS FOR EXP(X*X)*ERFC(X)*(X*SQRT(PI))
C     FOR X.GE.4.  TABLE 9 OF THE REFERENCE.
C
      DATA (AR(I), I=1,18)/
     &       1.97070527225754492387D0,  -0.01433974027177497552D0,
     &       0.00029736169220261895D0,  -0.00000980351604336237D0,
     &       0.00000043313342034728D0,  -0.00000002362150026241D0,
     &       0.00000000151549676581D0,  -0.00000000011084939856D0,
     &       0.00000000000904259014D0,  -0.00000000000080947054D0,
     &       0.00000000000007853856D0,  -0.00000000000000817918D0,
     &       0.00000000000000090715D0,  -0.00000000000000010646D0,
     &       0.00000000000000001315D0,  -0.00000000000000000170D0,
     &       0.00000000000000000023D0,  -0.00000000000000000003D0/
C
C           S2=INTEGRAL ON (0,2) OF EXP(W*W)*ERFC(W)
C           S4=INTEGRAL ON (0,4) OF EXP(W*W)*ERFC(W)
C
      DATA S2, S4/ 0.9753620874841564D+00,   0.1344468257503159D+01/
      IF(X.LT.4.0D0) THEN
C       POWER SERIES FOR X.LT.4
        IF(X.LE.2.0D0) THEN
          IF(X.LE.1.0D0) THEN
            X0=0.0D0
            SUM=0.0D0
          ELSE
            X0=2.0D0
            SUM=S2
          ENDIF
        ELSE
          IF(X.LE.3.0D0) THEN
            X0=2.0D0
            SUM=S2
          ELSE
            X0=4.0D0
            SUM=S4
          ENDIF
        ENDIF
        REL=0.5D-14
        N0=0
        KODE=2
        N=40
        TOL=REL
        CALL DINERFC(X0,N0,KODE,N,TOL,Y,NZ)
        TRMK=1.0D0
        AK=2.0D0
        DX=X-X0
        SS=Y(1)
        DO 10 K=1,N-1
          TRMK=-2.0D0*TRMK*DX
```

```
          TRM=TRMK*Y(K+1)/AK
          SS=SS+TRM
          IF(DABS(TRM).LE.TOL*DABS(SS)) GOTO 20
          AK=AK+1.0D0
 10     CONTINUE
 20     CONTINUE
        DHERFC=SUM+SS*DX
        RETURN
      ENDIF
C     CHEBYSHEV EXPANSION FOR X.GE.4
      RX=4.0D0/X
      TX=RX+RX
      CAPA=(1.0D0-RX)*(1.0D0+RX)
      SS=0.0D0
      TK=1.0D0
      TP=RX
      AK=1.0D0
      DO 30 J=1,3
        TM=TK
        TK=TP
        TP=TX*TK-TM
 30   CONTINUE
      DO 40 K=1,17
        SS=SS+AR(K+1)*CAPA
        RK=-0.5D0/(AK*(AK+1.0D0))-0.5D0*(TP/(AK+1.0D0)-TM/AK)
        CAPA=RK-CAPA
        TM=TK
        TK=TP
        TP=TX*TK-TM
        TM=TK
        TK=TP
        TP=TX*TK-TM
        AK=AK+1.0D0
 40   CONTINUE
      FX=X/4.0D0
      DHERFC=S4+(SS+DLOG(FX))/RTPI
      RETURN
      END
```

```
      SUBROUTINE BECKINFO
C-----------------------------------------------------------------------
C     DONALD E. AMOS, MAY 1, 2002, FEBRUARY, 2006
C
C     EACH CALLABLE ROUTINE HAS A REFERENCE IN THE PROLOGUE TO AN
C     APPROPRIATE FOLDER WHICH DESCRIBES THE ANALYTICAL BASIS OF THE
C     CODE.
C
C     USAGE:
C
C     COMPILE THIS FILE AND LINK IT TO ANY PROGRAM NEEDING A SUBROUTINE
C     OR FUNCTION FROM THE FOLLOWING LIST:
C
C     CALLABLE SUBROUTINES:                                FOLDER
C              INTEGI1(A,B,T,KODE,REL,ANSI1,IERR)          1,2,10
C              INTEGI2(A,B,T,KODE,ANSI2,IERR)                9
C              INTEGI9(A,B,T,KODE,ANSI9,IERR)                9
C              INTEGI3(A,B,C,T,ANSI3,IERR,KFORM)             7
C              INTEGI5(A,B,X,ANSI5,IERR)                     5
C              INTEGJ5(A,B,X,ANSJ5,IERR)                     5
C              INTEGV5(A,B,X,ANSV5,IERR)                     5
C              INTEGI6(A,B,T,KODE,ANSI6,IERR)              3,6,15
C              INTEGP(A,B,T,KODE,REL,PANS,IERR)              11
C              INTEGQ(A,B,T,REL,QANS,IERR)                   11
C              INTEGW3(A,B,T,KODE,REL,ANSW3,IERR)            10
C              INTEGI21(A,B,C,T,KODE,ANSI21,I21ERR,KFORM)    21
C              INTEGJ21(A,B,C,T,ANSJ21,J21ERR,KFORM)         21
C              INTEGI22(A,B,C,T,ANSI22,I22ERR,KFORM)         22
C              INTEGJ22(A,B,C,T,ANSJ22,J22ERR,KFORM)         22
C              INTEGI29(A,B,T,N0,NN,YN,IERR)                 29
C              INTEGS1(A,B,C,T,TOL,S1,IERR,KFORM)            21
C              INTEGS2(A,B,C,T,TOL,S2,IERR,KFORM)            22
C              GNSEQ(A,B,CAPT,M,REL,YN)                      21
C
C     CALLABLE DOUBLE PRECISION FUNCTIONS:
C              DVOFT(A,B,T,REL,IERR,KFORM)                   21
C              PHIZ(Z)                                       5
C-----------------------------------------------------------------------
      END
      SUBROUTINE INTEGI1(A,B,T,KODE,REL,ANSI1,IERR)
C
C     DONALD E. AMOS, JANUARY, 2003
C-----------------------------------------------------------------------
C     A DOUBLE PRECISION ROUTINE
C-----------------------------------------------------------------------
C
C     INTEGI1 COMPUTES THE I1 INTEGRAL OF FOLDERS 1, 2, AND 10 USING
C     THE CLOSED FORM DERIVED IN FOLDER 10 ON KODE=1:
C
C         ANSI1=INT ON (T,INF) OF EXP(-A^2*W^2)*ERF(B*W)/W^2
C
C     AND THE COMPLEMENTARY FORM FOR KODE=2:
C
C         ANSI1=INT ON (T,INF) OF EXP(-A^2*W^2)*ERFC(B*W)/W^2
C
C     WHERE A.GE.0.0D0, B.GE.0.0D0, T.GT.0.0D0.
C
C     IERR=0 IS A  NORMAL RETURN
C     IERR=1 MEANS T.LE.0.0D0, ANSI1=0.0D0 RETURNED
C     IERR=2 MEANS ONE OR BOTH OF THE PARAMETERS A OR B IS NEGATIVE
C            OR KODE IS NOT 1 OR 2 ANSI1=0.0D0 RETURNED
C     IERR=4 IS AN UNDERFLOW AND ANSI1=0.0D0 IS RETURNED
C-----------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION EN(1)
      DATA RTPI       /1.772453850905516D0/
      IERR=0
```

```
      IF (A.LT.0.0D0) THEN
        IERR=2
        ANSI1=0.0D0
        RETURN
      ENDIF
      IF (B.LT.0.0D0) THEN
        IERR=2
        ANSI1=0.0D0
        RETURN
      ENDIF
      IF (T.LE.0.0D0) THEN
        IERR=1
        ANSI1=0.0D0
        RETURN
      ENDIF
      IF((KODE.NE.1).AND.(KODE.NE.2))THEN
        IERR=2
        ANSI1=0.0D0
        RETURN
      ENDIF
      IF(B.EQ.0.0D0) THEN
        IF(KODE.EQ.1) THEN
          ANSI1=0.0D0
          RETURN
        ELSE
          IF(A.EQ.0.0D0) THEN
            ANSI1=1.0D0/T
            RETURN
          ENDIF
          AT=A*T
          KKODE=1
          CALL DIERFC(AT,KKODE,ANSIE,IERR)
          IF(IERR.EQ.2) IERR=4
          ANSI1=RTPI*ANSIE/T
          RETURN
        ENDIF
      ENDIF
      IF(A.EQ.0.0D0) THEN
C       B NOT ZERO HERE
        BT=B*T
        IF(BT.GT.670.0D0) THEN
          ANSI1=0.0D0
          IERR=4
          RETURN
        ENDIF
        ERFB=DRERF(BT,KODE,NZ)
        BT2=BT*BT
        N=1
        KKODE=1
        M=1
        TOL=REL
        NZ=0
        IERR=0
        CALL DEXINT(BT2, N, KKODE, M, TOL, EN, NZ, IERR)
        IF(KODE.EQ.1) THEN
          ANSI1=ERFB/T+B*EN(1)/RTPI
        ELSE
          ANSI1=ERFB/T-B*EN(1)/RTPI
        ENDIF
        RETURN
      ENDIF
      A2=A*A
      B2=B*B
      T2=T*T
      AT=A*T
      BT=B*T
      ARG=(A2+B2)*T2
```

```
      N=1
      KKODE=1
      M=1
      TOL=REL
      NZ=0
      KERR=0
      CALL DEXINT(ARG, N, KKODE, M, TOL, EN, NZ, KERR)
      AT2=AT*AT
      DEX=DEXP(-AT2)
      ERFB=DRERF(BT,KODE,NZ)
      IF(KODE.EQ.1) THEN
C       THE MAGNITUDE OF DEX MUST BE AT LEAST 2 WORD LENGTHS GREATER
C       THAN UNDERFLOW TO GUARANTEE PROPER ARITHMETIC FOR KODE=1
        IF(AT2.GT.620.0D0) THEN
          ANSI1=0.0D0
          IERR=4
          RETURN
        ENDIF
        IF(A.GT.B) THEN
C         COMBINING THE LEADING ERF FUNCTIONS FROM J5 (FOLDER5) WITH THE
C         ERF(BT) FUNCTION FROM I1 IN FOLDER 10
          CALL GSERI5(B,A,T,GANS,JERR)
          KKODE=1
          CALL DIERFC(AT,KKODE,ANSIE,IERR)
          TRMA=RTPI*ANSIE*ERFB/T
          TRM=TRMA+B*EN(1)/RTPI-(A+A)*GANS*B
        ELSE
          CALL INTEGJ5(A,B,T,ANSJ5,JERR)
          TRMA=DEX*ERFB/T
          TRM=TRMA+B*EN(1)/RTPI-(A2+A2)*ANSJ5
        ENDIF
      ELSE
C       THE ASYMPTOTICS FOR ALL TERMS ARE DOMINATED BY DEXP(-ARG).
C       HENCE THE UNDER FLOW TEST ON ARG.
        IF(ARG.GT.670.0D0) THEN
          ANSI1=0.0D0
          IERR=4
          RETURN
        ENDIF
        IF((A.GT.B).AND.(ARG.GT.3.0D0)) THEN
C         COMBINING THE LEADING ERFC FUNCTIONS FROM I5 (FOLDER5) WITH
C         THE ERFC(BT) FUNCTION FROM I1C IN FOLDER 24A
          KKODE=1
          CALL DIERFC(AT,KKODE,ANSIE,IERR)
          TRMA=RTPI*ANSIE*ERFB/T
          CALL GSERI5(B,A,T,GANS,JERR)
          TRM=TRMA-B*EN(1)/RTPI+(A+A)*GANS*B
        ELSE
          CALL INTEGI5(A,B,T,ANSI5,KERR)
          TRMA=DEX*ERFB/T
          TRM=TRMA-B*EN(1)/RTPI-(A2+A2)*ANSI5
        ENDIF
      ENDIF
      ANSI1=TRM
      RETURN
      END
      SUBROUTINE INTEGI2(A,B,T,KODE,ANSI2,IERR)
C
C     DONALD E. AMOS   DECEMBER, 2005
C-----------------------------------------------------------------------
C     A DOUBLE PRECISION ROUTINE
C-----------------------------------------------------------------------
C     REF: FOLDER 9 , FOLDER 7
C
C     ON KODE=1, INTEGI2 COMPUTES THE INTEGRAL
C
C                 ANSI2=INT ON (0,T) OF ERF(A*W)*ERF(B*W);
```

```
C
C     ON KODE=2, INTEGI2 COMPUTES  THE INTEGRAL
C
C                ANSI2=INT ON (T,INF) OF ERFC(A*W)*ERFC(B*W)
C
C     WHERE      A.GE.0.0D0, B.GE.0.0D0, AND T.GE.0.0D0
C                (A AND B NOT ZERO AT THE SAME TIME FOR KODE=2)
C
C     ERROR CONDITIONS:
C
C     IERR=0 NORMAL RETURN, COMPUTATION COMPLETED
C     IERR=1 A IS NEGATIVE OR B IS NEGATIVE OR C IS NON-POSITIVE OR KODE
C            IS NOT 1 OR 2, OR T IS NEGATIVE, ANSI2=0.0D0 RETURNED
C     IERR=2 IS AN UNDERFLOW,  ANSI2=0.0D0 RETURNED
C     IERR=3 NO CONVERGENCE IN A SERIES
C     IERR=4 KODE=2, A=0.0D0, AND B=0.0D0; THE INTEGRAL DOES NOT EXIST
C
C     CALLS ROUTINES: DRERF, DINERFC
C-----------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION Y(61), CA(62), CB(62)
      DATA RTPI, PI/ 1.772453850905516D0, 3.1415926535897932D0/
      IERR=0
      IF((A.LT.0.0D0).OR.(B.LT.0.0D0).OR.(T.LT.0.0D0)) THEN
        IERR=1
        ANSI2=0.0D0
        RETURN
      ENDIF
      IF((KODE.NE.1).AND.(KODE.NE.2)) THEN
        IERR=1
        ANSI2=0.0D0
        RETURN
      ENDIF
      IF(KODE.EQ.1) THEN
        IF((A.EQ.0.0D0).OR.(B.EQ.0.0D0).OR.(T.EQ.0.0D0)) THEN
          ANSI2=0.0D0
          RETURN
        ENDIF
      ELSE
        IF((A.EQ.0.0D0).AND.(B.EQ.0.0D0)) THEN
          IERR=4
          ANSI2=0.0D0
          RETURN
        ELSE
          IF(T.EQ.0.0D0) THEN
            DNUM=2.0D0/RTPI
            ANSI2=DNUM/(A+B+DSQRT(A*A+B*B))
            RETURN
          ENDIF
          IF(MIN(A,B).EQ.0.0D0) THEN
            XB=MAX(A,B)
            BT=XB*T
            KKODE=1
            CALL DIERFC(BT,KKODE,ANSB,IERR)
            ANSI2=ANSB/XB
            RETURN
          ENDIF
        ENDIF
      ENDIF
      XA=MIN(A,B)
      XB=MAX(A,B)
      AT=XA*T
      BT=XB*T
      A2=XA*XA
      B2=XB*XB
      T2=T*T
      A2B2=DSQRT(A2+B2)
```

```
      IF(KODE.EQ.1) THEN
        XLIM=1.25D0
        IF((AT.LE.XLIM).AND.(BT.LE.XLIM).AND.(T.LE.1.0D0)) THEN
          SUM=1.0D0/3.0D0
          CA(1)=1.0D0
          CB(1)=1.0D0
          AK=2.0D0
          TKP1=5.0D0
          TKP3=5.0D0
          T2A=AT*AT
          T2B=BT*BT
          CA(2)=-T2A/3.0D0
          CB(2)=-T2B/3.0D0
          TA=-T2A
          TB=-T2B
          REL=0.5D-15
          DO 5 K=1,60
            KP1=K+1
            UK=0.0D0
            DO 2 M=1,KP1
              UK=UK+CA(M)*CB(KP1-M+1)
  2         CONTINUE
            TRM=UK/TKP3
            SUM=SUM+TRM
            IF(DABS(TRM).LT.REL*DABS(SUM)) GOTO 6
            TA=-TA*T2A/AK
            TB=-TB*T2B/AK
            CA(K+2)=TA/TKP1
            CB(K+2)=TB/TKP1
            AK=AK+1.0D0
            TKP3=TKP3+2.0D0
            TKP1=TKP1+2.0D0
  5       CONTINUE
          IERR=3
  6       CONTINUE
          COEF=4.0D0*AT*BT*T/PI
          ANSI2=COEF*SUM
          RETURN
        ENDIF
        ERFAT=DRERF(AT,KODE,NZ)
        ERFBT=DRERF(BT,KODE,NZ)
        IF(AT.LT.2.40D0) THEN
          S=T*ERFAT*ERFBT+DEXP(-B2*T2)*ERFAT/(XB*RTPI)
          ARG=0.5D0*AT*AT
          R1=XA/(XB+A2B2)
          R=R1/(XB*RTPI)
          SS=-2.0D0*(1.0D0/(XA*RTPI)+R)*DEXP(-ARG)*SINH(ARG)
          SS=SS-R*DEXP(-A2*T2)*ERFBT
          S=S+SS
          KKODE=2
          ERFAT=DRERF(AT,KKODE,NZ)
          ERFBT=DRERF(BT,KKODE,NZ)
          COEF=T*R*A2B2*DEXP(-AT*AT)
          N0=1
          KKODE=1
          N=60
          REL=0.5D-14
          CALL DINERFC(BT,N0,KKODE,N,REL,Y,NZ)
          SUM=0.0D0
          RAT=AT*R1
          PWR=-2.0D0
          DO 10 N=1,60
            TRM=PWR*Y(N)
            SUM=SUM+TRM
            IF(DABS(TRM).LE.REL*DABS(SUM)) GOTO 15
            PWR=-2.0D0*PWR*RAT
 10       CONTINUE
```

```
          IERR=3
   15     CONTINUE
          ANSI2=S+COEF*SUM
          RETURN
        ENDIF
        ARG=T*A2B2
        ERFABT=DRERF(ARG,KODE,NZ)
        S=T*ERFAT*ERFBT
        S=S+DEXP(-A2*T2)*ERFBT/(XA*RTPI)
        S=S+DEXP(-B2*T2)*ERFAT/(XB*RTPI)
        ANSI2=S-A2B2*ERFABT/(XA*XB*RTPI)
      ELSE
        EX=T2*(A2+B2)
        IF(EX.GT.670.0D0) THEN
          IERR=2
          ANSI2=0.0D0
          RETURN
        ENDIF
        XLIM=1.25D0
        IF((AT.LE.XLIM).AND.(BT.LE.XLIM).AND.(T.LE.1.0D0)) THEN
          SUMAB=1.0D0/3.0D0
          SUMA=1.0D0/2.0D0
          SUMB=1.0D0/2.0D0
          CA(1)=1.0D0
          CB(1)=1.0D0
          AK=2.0D0
          TKP1=5.0D0
          TKP2=4.0D0
          TKP3=5.0D0
          T2A=AT*AT
          T2B=BT*BT
          CA(2)=-T2A/3.0D0
          CB(2)=-T2B/3.0D0
          TA=-T2A
          TB=-T2B
          REL=0.5D-15
          DO 35 K=1,60
            KP1=K+1
            TRMA=CA(KP1)/TKP2
            TRMB=CB(KP1)/TKP2
            SUMA=SUMA+TRMA
            SUMB=SUMB+TRMB
            UK=0.0D0
            DO 32 M=1,KP1
              UK=UK+CA(M)*CB(KP1-M+1)
   32       CONTINUE
            TRMAB=UK/TKP3
            SUMAB=SUMAB+TRMAB
            IF(DABS(TRMAB).LT.REL*DABS(SUMAB)) THEN
              IF(DABS(TRMA).LT.REL*DABS(SUMA)) THEN
                IF(DABS(TRMB).LT.REL*DABS(SUMB)) GOTO 36
              ENDIF
            ENDIF
            TA=-TA*T2A/AK
            TB=-TB*T2B/AK
            CA(K+2)=TA/TKP1
            CB(K+2)=TB/TKP1
            AK=AK+1.0D0
            TKP3=TKP3+2.0D0
            TKP2=TKP2+2.0D0
            TKP1=TKP1+2.0D0
   35     CONTINUE
          IERR=3
   36     CONTINUE
          COEFA=2.0D0*AT*T/RTPI
          COEFB=2.0D0*BT*T/RTPI
          COEFAB=4.0D0*AT*BT*T/PI
```

```
          ANSA=COEFA*SUMA
          ANSB=COEFB*SUMB
          ANSAB=COEFAB*SUMAB
          DNUM=2.0D0/RTPI
          ANST0=DNUM/(A+B+DSQRT(A*A+B*B))
          ANSI2=ANST0-T+ANSA+ANSB-ANSAB
          RETURN
        ENDIF
        IF(AT.LE.2.40D0) THEN
          ERFAT=DRERF(AT,KODE,NZ)
          ERFBT=DRERF(BT,KODE,NZ)
          N0=1
          KKODE=1
          N=60
          REL=0.5D-14
          CALL DINERFC(BT,N0,KKODE,N,REL,Y,NZ)
          R1=XA/(XB+A2B2)
          R=R1/(XB*RTPI)
C         Y(1)=IERFC(BT)
          S=ERFAT*Y(1)/XB-EXP(-A2*T2)*ERFBT*R
          COEF=T*A2B2*R*DEXP(-AT*AT)
          SUM=0.0D0
          RAT=AT*R1
          PWR=-2.0D0
          DO 20 N=1,60
            TRM=PWR*Y(N)
            SUM=SUM+TRM
            IF(DABS(TRM).LE.REL*DABS(SUM)) GOTO 30
            PWR=-2.0D0*PWR*RAT
   20     CONTINUE
          IERR=3
   30     CONTINUE
          ANSI2=S-COEF*SUM
          RETURN
        ELSE
          ARG=T*A2B2
          KK0DE=1
          CALL DIERFC(ARG,KKODE,ANSAB,KERR)
          CALL DIERFC(AT,KKODE,ANSA,KERR)
          CALL DIERFC(BT,KKODE,ANSB,KERR)
          ANSI2=T*(ANSAB/RTPI-ANSA*ANSB)/(AT*BT)
        ENDIF
      ENDIF
      RETURN
      END
      SUBROUTINE INTEGI9(A,B,T,KODE,ANSI9,IERR)
C
C     DONALD E. AMOS   DECEMBER, 2005; APRIL, 2006
C-----------------------------------------------------------------------
C     A DOUBLE PRECISION ROUTINE
C-----------------------------------------------------------------------
C
C     REF. FOLDER 9
C
C     ON KODE=1, INTEGI9 COMPUTES THE INTEGRAL
C
C                ANSI9 =  INT ON (0,T) OF W*ERF(A*W)*ERF(B*W)
C
C     ON KODE=2, INTEGI9 COMPUTES THE COMPLEMENTARY INTEGRAL
C
C                ANSI9 =  INT ON (T,INF) OF W*ERFC(A*W)*ERFC(B*W)
C
C     FOR A.GE.0, B.GE.0, T.GE.0.
C
C     ERROR CONDITIONS
C     IERR=0 NORMAL RETURN, COMPUTATION COMPLETED
C     IERR=1 A IS NEGATIVE OR B IS NEGATIVE OR T IS NEGATIVE OR KODE
```

```
C             IS NOT 1 OR 2, ANSI9=0.0D0 RETURNED
C      IERR=2 IS AN UNDERFLOW,  ANSI9=0.0D0 RETURNED
C      IERR=3 NO CONVERGENCE IN A SERIES
C      IERR=4 KODE=2, A=0.0D0, AND B=0.0D0; THE INTEGRAL DOES NOT EXIST
C
C      CALLS ROUTINES: DRERF, INERFC, DIERFC, INTEGI5, INTEGV5
C-----------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION CA(62),CB(62),Y(62)
      DATA RTPI, PI  /1.772453850905516D0, 3.1415926535897932D0/
      IERR=0
      IF((A.LT.0.0D0).OR.(B.LT.0.0D0).OR.(T.LT.0.0D0)) THEN
        IERR=1
        ANSI9=0.0D0
        RETURN
      ENDIF
      IF((KODE.NE.1).AND.(KODE.NE.2)) THEN
        IERR=1
        ANSI9=0.0D0
        RETURN
      ENDIF
      AT=A*T
      BT=B*T
      A2=A*A
      B2=B*B
      T2=T*T
      AT2=AT*AT
      BT2=BT*BT
      ARTPI=A*RTPI
      BRTPI=B*RTPI
      IF(KODE.EQ.1) THEN
        XA=MIN(A,B)
        IF((XA.EQ.0.0D0).OR.(T.EQ.0.0D0)) THEN
          ANSI9=0.0D0
          RETURN
        ENDIF
C       I9 BY POWER SERIES
        XLIM=1.25D0
        IF((AT.LE.XLIM).AND.(BT.LE.XLIM).AND.(T.LE.1.0D0)) THEN
          SUM=1.0D0/4.0D0
          CA(1)=1.0D0
          CB(1)=1.0D0
          AK=2.0D0
          TKP1=5.0D0
          TKP4=6.0D0
          T2A=AT*AT
          T2B=BT*BT
          CA(2)=-T2A/3.0D0
          CB(2)=-T2B/3.0D0
          TA=-T2A
          TB=-T2B
          REL=0.5D-15
          DO 5 K=1,60
            KP1=K+1
            UK=0.0D0
            DO 2 M=1,KP1
              UK=UK+CA(M)*CB(KP1-M+1)
  2         CONTINUE
            TRM=UK/TKP4
            SUM=SUM+TRM
            IF(DABS(TRM).LT.REL*DABS(SUM)) GOTO 6
            TA=-TA*T2A/AK
            TB=-TB*T2B/AK
            CA(K+2)=TA/TKP1
            CB(K+2)=TB/TKP1
            AK=AK+1.0D0
            TKP4=TKP4+2.0D0
```

```
            TKP1=TKP1+2.0D0
  5       CONTINUE
          IERR=3
  6       CONTINUE
          COEF=4.0D0*AT*BT*T2/PI
          ANSI9=COEF*SUM
          RETURN
        ENDIF
        XB=MAX(A,B)
C       I9 FOR A.LE.B AND XAT.LE.2.4
        XAT=XA*T
        IF(XAT.LE.2.4D0) THEN
          XBT=XB*T
          KKODE=1
          ERFAT=DRERF(XAT,KKODE,NZ)
          KKODE=1
          ERFBT=DRERF(XBT,KKODE,NZ)
          RTAB=DSQRT(A2+B2)
          XAT2=XAT*XAT
          EXA=DEXP(-XAT2)
          COEF=(XA/XB)*EXA/((XB+RTAB)*RTPI*RTAB)
          N0=1
          KKODE=1
          N=60
          REL=0.5D-14
          TOL=REL
          CALL DINERFC(XBT,N0,KKODE,N,TOL,Y,NZ)
          CALL INTEGV5(XB,XA,T,ANSV5,IERR5)
          XBT2=XBT*XBT
          EXB=DEXP(-XBT2)
          SUM=T2*ERFAT*ERFBT+(T*EXB*ERFAT-ANSV5)/(XB*RTPI)
          IF(XAT.LE.0.50D0) THEN
C           COMPUTE H1=(T*DEXP(-XAT2)-0.5D0*RTPI*ERF(XAT)/XA)/(XA*RTPI)
C           BY POWER SERIES TO RESOLVE THE INDETERMINANCY FOR XA TO ZERO
            S=-XAT/3.0D0
            AK=-XAT
            FAC=1.0D0
            TKP1=5.0D0
            REL=0.5D-15
            DO 60 K=2,50
              AK=-AK*XAT2/FAC
              TRM=AK/TKP1
              S=S+TRM
              IF(DABS(TRM).LE.REL*DABS(S)) GOTO 61
              FAC=FAC+1.0D0
              TKP1=TKP1+2.0D0
 60         CONTINUE
            IERR=3
 61         CONTINUE
            H1=2.0D0*T2*S/RTPI
          ELSE
            H1=(T*EXA-0.5D0*RTPI*ERFAT/XA)/(XA*RTPI)
          ENDIF
C         COMPUTE H2=(DATAN(RAB)-RAB)/(XA*XA*PI) BY POWER SERIES FOR
C         RAB.LE.0.5 TO RESOLVE THE INDETERMINANCY FOR XA TO ZERO
          RAB=XA/XB
          IF(RAB.LE.0.25D0) THEN
            RAB2=RAB*RAB
            S=-RAB/3.0D0
            AK=-RAB
            TKP1=5.0D0
            REL=0.5D-15
            DO 70 K=2,50
              AK=-AK*RAB2
              TRM=AK/TKP1
              S=S+TRM
              IF(DABS(TRM).LE.REL*DABS(S)) GOTO 71
```

```
              TKP1=TKP1+2.0D0
 70         CONTINUE
            IERR=3
 71         CONTINUE
            H2=S/(XB*XB*PI)
          ELSE
            H2=(DATAN(RAB)-RAB)/(XA*XA*PI)
          ENDIF
          SUM=SUM+H1+H2
          S=0.0D0
          RX=XA*XAT/(XB+RTAB)
          PX=-2.0D0
          AN=2.0D0
          REL=0.5D-15
          DO 45 N=2,60
            TRM=Y(N)*PX*AN
            S=S+TRM
            IF(DABS(TRM).LE.REL*DABS(S)) GOTO 46
            PX=-2.0D0*PX*RX
            AN=AN+1.0D0
  45      CONTINUE
C          WRITE(7,307)
          IERR=3
  46      CONTINUE
          WTRM=COEF*(-Y(1)+XBT*S)
C         ANSG9=G1(A,B,T)/A
          CALL GSERI9(XA,XB,T,ANSG9,IERRG)
          ANSI9=0.5D0*(SUM-WTRM-ANSG9)
          RETURN
        ENDIF
C       DEFAULT COMPUTATION FOR I9
        KKODE=1
        ERFA=DRERF(AT,KKODE,NZ)
        KKODE=1
        ERFB=DRERF(BT,KKODE,NZ)
        ARTPI=A*RTPI
        BRTPI=B*RTPI
        SUM=T*ERFA*ERFB
        SUM=SUM+DEXP(-AT2)*ERFB/(ARTPI)+DEXP(-BT2)*ERFA/(BRTPI)
        CALL INTEGV5(A,B,T,ANSAB,IERRB)
        CALL INTEGV5(B,A,T,ANSBA,IERRA)
        SUM=T*SUM-ANSAB/ARTPI
        SUM=SUM-ANSBA/BRTPI
        ARG=0.5D0*(AT2+BT2)
        IF(ARG.GT.18.0D0) THEN
          PHI=1.0D0
        ELSE
          PHI=2 0D0*DEXP(-ARG)*DSINH(ARG)
        ENDIF
        ANSI9=0.5D0*(SUM-PHI/(A*B*PI))
        RETURN
      ELSE
C       COMPUTE FOR I9C
        ARG=T2*(A2+B2)
        IF(ARG.GT.670.0D0) THEN
          ANSI9=0.0D0
          IERR=2
          RETURN
        ENDIF
        IF((A.EQ.0.0D0).AND.(B.EQ.0.0D0)) THEN
          ANSI9=0.0D0
          IERR=4
          RETURN
        ENDIF
        XA=MIN(A,B)
        IF(XA.EQ.0.0D0) THEN
          XB=MAX(A,B)
```

```
          XBT=XB*T
          N0=1
          KKODE=1
          N=2
          REL=0.5D-14
          CALL DINERFC(XBT,N0,KKODE,N,REL,Y,NZ)
          ANSI9=T*Y(1)/XB+Y(2)/(XB*XB)
          IF(NZ.NE.0) THEN
            ANSI9=0.0D0
            IERR=2
          ENDIF
          RETURN
        ENDIF
C       I9C BY POWER SERIES
        XLIM=1.25D0
        IF((AT.LE.XLIM).AND.(BT.LE.XLIM).AND.(T.LE.1.0D0)) THEN
          SUMAB=1.0D0/4.0D0
          SUMA=1.0D0/3.0D0
          SUMB=1.0D0/3.0D0
          CA(1)=1.0D0
          CB(1)=1.0D0
          AK=2.0D0
          TKP1=5.0D0
          TKP3=5.0D0
          TKP4=6.0D0
          T2A=AT*AT
          T2B=BT*BT
          CA(2)=-T2A/3.0D0
          CB(2)=-T2B/3.0D0
          TA=-T2A
          TB=-T2B
          REL=0.5D-15
          DO 35 K=1,60
            KP1=K+1
            TRMA=CA(KP1)/TKP3
            TRMB=CB(KP1)/TKP3
            SUMA=SUMA+TRMA
            SUMB=SUMB+TRMB
            UK=0.0D0
            DO 32 M=1,KP1
              UK=UK+CA(M)*CB(KP1-M+1)
   32       CONTINUE
            TRMAB=UK/TKP4
            SUMAB=SUMAB+TRMAB
            IF(DABS(TRMAB).LT.REL*DABS(SUMAB)) THEN
              IF(DABS(TRMA).LT.REL*DABS(SUMA)) THEN
                IF(DABS(TRMB).LT.REL*DABS(SUMB)) GOTO 36
              ENDIF
            ENDIF
            TA=-TA*T2A/AK
            TB=-TB*T2B/AK
            CA(K+2)=TA/TKP1
            CB(K+2)=TB/TKP1
            AK=AK+1.0D0
            TKP3=TKP3+2.0D0
            TKP4=TKP4+2.0D0
            TKP1=TKP1+2.0D0
   35     CONTINUE
          IERR=3
   36     CONTINUE
          COEFA=2.0D0*AT*T2/RTPI
          COEFB=2.0D0*BT*T2/RTPI
          COEFAB=4.0D0*AT*BT*T2/PI
          ANSA=COEFA*SUMA
          ANSB=COEFB*SUMB
          ANSAB=COEFAB*SUMAB
          XA=MIN(A,B)
```

```
          XB=MAX(A,B)
          X=XA/XB
          XB2=XB*XB
C         COMPUTE I9C(A,B,0) BY SERIES IF A/B.LT.0.125
          IF(X.LT.0.125D0) THEN
            REL=0.5D-15
            SUM=0.0D0
            AK=3.0D0
            X2=X*X
            PX=-XA/XB2
            DO 50 K=1,20
              TRM=PX/AK
              SUM=SUM+TRM
              IF(DABS(TRM).LT.REL*DABS(SUM)) GOTO 51
              AK=AK+2.0D0
              PX=-PX*X2
  50        CONTINUE
            IERR=3
  51        CONTINUE
            TEMP=0.5D0*SUM/(XB*PI)+0.25D0/XB2
            ANST0=TEMP-0.5D0*DATAN(X)/(XB2*PI)
          ELSE
            TEMP=0.5D0*(DATAN(X)/X-1.0D0)/(XA*XB*PI)+0.25D0/XB2
            ANST0=TEMP-0.5D0*DATAN(X)/(XB2*PI)
          ENDIF
          ANSI9=ANST0-0.5D0*T2+ANSA+ANSB-ANSAB
          RETURN
        ENDIF
        XB=MAX(A,B)
C       I9C FOR A.LE.B AND XAT.LE.2.4
        XAT=XA*T
        IF(XAT.LE.2.4D0) THEN
          XBT=XB*T
          KKODE=2
          ERFAT=DRERF(XAT,KKODE,NZ)
          KKODE=2
          ERFBT=DRERF(XBT,KKODE,NZ)
          RTAB=DSQRT(A2+B2)
          XAT2=XAT*XAT
          COEF=(XA/XB)*DEXP(-XAT2)/((XB+RTAB)*RTPI*RTAB)
          N0=1
          KKODE=1
          N=60
          REL=0.5D-14
          TOL=REL
          CALL DINERFC(XBT,N0,KKODE,N,TOL,Y,NZ)
          CALL INTEGI5(XB,XA,T,ANSI5,IERR5)
          SUM=(T*ERFAT*Y(1)+ANSI5/RTPI)/XB
          S=0.0D0
          RX=XA*XAT/(XB+RTAB)
          PX=-2.0D0
          AN=2.0D0
          DO 40 N=2,60
            TRM=Y(N)*PX*AN
            S=S+TRM
            IF(DABS(TRM).LT.REL*DABS(S)) GOTO 41
            PX=-2.0D0*PX*RX
            AN=AN+1.0D0
  40      CONTINUE
C          WRITE(7,307)
C  307     FORMAT("NO CONVERGENCE IN INERFC LOOP")
          IERR=3
  41      CONTINUE
          WTRM=COEF*(-Y(1)+XBT*S)
          CALL GSERI9(XA,XB,T,ANSG9,IERRG)
          ANSI9=0.5D0*(SUM+WTRM+ANSG9)
          RETURN
```

```
        ENDIF
C       DEFAULT COMPUTATION FOR I9C
        KKODE=1
        CALL DIERFC(BT,KKODE,DIERFB,IERRB)
        CALL DIERFC(AT,KKODE,DIERFA,IERRA)
        SUM=-DIERFA*DIERFB/(A*B)
        CALL INTEGI5(A,B,T,ANSAB,KERR)
        CALL INTEGI5(B,A,T,ANSBA,KERR)
        ANSI9=0.5D0*(SUM+ANSAB/ARTPI+ANSBA/BRTPI)
        RETURN
      ENDIF
      END
      SUBROUTINE GSERI9(A,B,X,SUM,IERR)
C
C     GSERI9 COMPUTES THE G SUM OF FOLDER 9 = G SUM OF FOLDER 5 LESS
C     FIRST TERM MULTIPLIED BY 1/(A*RTPI).
C
C     IERR=0 NORMAL RETURN
C     IERR=1 UNDERFLOW, SUM=0.0D0
C     IERR=2 NO CONVERGENCE IN 55 TERMS OF THE SERIES
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION EN(55)
      DATA RTPI, PI  /1.772453850905516D0, 3.1415926535897932D0/
      IERR=0
      ARG=A*A+B*B
      DEN=ARG
      ARG=ARG*X*X
      FNH=2.5D0
      KODE=1
      M=55
      TOL=0.5D-15
      NZ=0
      KERR=0
      CALL DHEXINT(ARG, FNH, KODE, M, TOL, EN, NZ, KERR)
      IF(NZ.NE.0) THEN
        IERR=1
        SUM=0.0D0
        RETURN
      ENDIF
      S=0.0D0
      AK=A/DEN
      DH=0.5D0
      Z=A*A/DEN
      FAC=1.0D0
      DO 10 K=1,M-1
        AK=AK*DH/FAC
        TRM=AK*EN(K)
        S=S+TRM
        IF (DABS(TRM).LE.TOL*DABS(S)) GOTO 15
        DH=DH+1.0D0
        FAC=FAC+1.0D0
        AK=AK*Z
  10  CONTINUE
       IERR=2
C      WRITE (7,300)
C 300  FORMAT('G9-SERIES NOT CONVERGED IN 55 TERMS')
  15  CONTINUE
      SUM=(0.5D0/PI)*(S/DSQRT(DEN))
      RETURN
      END
      SUBROUTINE INTEGI3(A,B,C,T,KODE,ANSI3,IERR,KFORM)
C
C     DONALD E. AMOS   MAY, 2001   DECEMBER, 2005
C-----------------------------------------------------------------------
C     A DOUBLE PRECISION ROUTINE
C-----------------------------------------------------------------------
```

```
C
C     INTEGI3 COMPUTES THE I3 AND THE I3C INTEGRALS OF FOLDER 7,
C
C     ANSI3=INT ON (T,INF)OF EXP(-C*C*W*W)*ERF(A*W)*ERF(B*W)   ON KODE=1
C
C     AND
C
C     ANSI3=INT ON (T,INF)OF EXP(-C*C*W*W)*ERFC(A*W)*ERFC(B*W) ON KODE=2
C
C     FOR C>0, WITH T, A AND B  NON-NEGATIVE.
C
C     ERROR CONDITIONS
C
C     IERR=0  NORMAL RETURN
C     IERR=1  C IS ZERO OR NEGATIVE AND I3 IS UNDEFINED. ANSI3=0.0D0
C             RETURNED
C     IERR=2  A, B OR T IS NEGATIVE OR KODE IS NOT 1 OR 2. I3 IS DEFINED
C             HERE FOR NON-NEGATIVE A AND B, THOUGH I3 EXITS FOR
C             NEGATIVE VALUES OF A AND B. ANSI3=0.0D0 RETURNED
C     IERR=3  UNDERFLOW, ANSI3=0.0D0 RETURNED
C
C             CONVERGENCE OR COMPUTATIONAL PROBLEMS (NOT LIKELY)
C
C     IERR=4  NO CONVERGENCE IN SSERI3 SERIES IN 85 TERMS. ANSI3=0.0D0
C             RETURNED
C     IERR=5  NO CONVERGENCE IN I5 SERIES IN INTEGI5 IN 55 TERMS
C             ANSI3=0.0D0 RETURNED
C
C     CONDITIONS IERR = 4 OR 5 SHOULD ONLY OCCUR IF THE MACHINE FAILS,
C     SO THERE IS GENERALLY NO REASON TO TEST FOR THEM.
C
C     KFORM IS A RETURN VARIABLE TELLING WHICH FORMULA WAS USED IN THE
C     COMPUTATION OF ANSI3 WHEN IERR=0:
C
C     KFORM=0 MEANS EITHER A OR B WAS ZERO. ANSI3=0.0D0 RETURNED
C     KFORM=1 MEANS THE ENHANCEMENT (A)- ERF(*)=1, BOTH AT AND BT.GE.6
C     KFORM=2 MEANS THE ENHANCEMENT (B)- POWER SERIES AT,BT,CT.LE.2.0D0
C     KFORM=3 MEANS THE ENHANCEMENT (C)- LARGE C FORMULA
C     KFORM=4,5 MEANS MAIN FORMULA CASE I  FOR C.LE.XA.LE.XB
C     KFORM=6,7 MEANS MAIN FORMULA CASE I  FOR XA.LE.C.LE.XB
C     KFORM=8   MEANS MAIN FORMULA CASE II FOR XA.LE.XB.LE.C
C               WHERE XA=MIN(A,B) AND XB=MAX(A,B)
C
C     CALLS ROUTINES: PHIZ, PKSER, SSERI3, I3LARGEC, I3PWRSER, FHYPER,
C                     INTEGI5, SSERI5, DHEXINT, DRERF
C-----------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION ENH(85)
      DATA RTPI ,PI /1.772453850905516D0, 3.14159265358979D0/
      IERR=0
      KFORM=0
      IF (C.LE.0.0D0) THEN
C       IF C=0, THEN I3 DOES NOT EXIST.
        IERR=1
        ANSI3=0.0D0
        RETURN
      ENDIF
      IF((A.LT.0.0D0).OR.(B.LT.0.0D0))THEN
C       I3 HERE IS DEFINED ONLY FOR A AND B NON-NEGATIVE.
        IERR=2
        ANSI3=0.0D0
        RETURN
      ENDIF
      IF(T.LT.0.0D0) THEN
C       THE INTEGRAL IS DEFINED ONLY FOR T.GE.0.0D0
        IERR=2
        ANSI3=0.0D0
```

```
          RETURN
        ENDIF
        IF((KODE.NE.1).AND.(KODE.NE.2)) THEN
          IERR=2
          ANSI3=0.0D0
          RETURN
        ENDIF
        IF(KODE.EQ.2) THEN
          IF((A.EQ.0.0D0).OR.(B.EQ.0.0D0)) THEN
            XB=MAX(A,B)
            IF(XB.EQ.0.0D0) THEN
              ARG=C*T
              ERFA=DRERF(ARG,KODE,NZ)
              ANSI3=0.5D0*RTPI*ERFA/C
              RETURN
            ELSE
              CALL INTEGI5(C,XB,T,ANSI3,IERR)
              RETURN
            ENDIF
          ENDIF
          CALL J3SER(A,B,C,T,ANSI3)
          RETURN
        ENDIF
        IF((A.EQ.0.0D0).OR.(B.EQ.0.0D0)) THEN
          ANSI3=0.0D0
          RETURN
        ENDIF
        XB=MAX(A,B)
        XA=MIN(A,B)
        AT=XA*T
        BT=XB*T
        CT=C*T
        IF (AT.GE.6.0D0) THEN
C         TEST FOR AT>6 AND BT>6 WHERE BOTH ERF FUNCTIONS ARE 1
C         XA=MIN(A,B), XB=MAX(A,B), AT=XA*T.LE.BT=XB*T
          KKODE=2
          NZ=0
          ERFCT=DRERF(CT,KKODE,NZ)
          IF (NZ.NE.0) THEN
            IERR=3
            ANSI3=0.0D0
            KFORM=1
            RETURN
          ENDIF
          ANSI3=0.5D0*RTPI*ERFCT/C
          KFORM=1
          RETURN
        ENDIF
C       TEST FOR POWER SERIES
        IF((AT.LE.2.0D0).AND.(BT.LE.2.0D0).AND.(CT.LE.2.0D0)) THEN
          CALL I3PWRSER(A,B,C,T,SUM,IERRP)
          IF (IERRP.EQ.1) THEN
C           ON CONVERGENCE FAILURE, LET OTHER CASES TRY TO COMPUTE
            GOTO 100
          ENDIF
          IF (C.LE.XB) THEN
            TT=0.0D0
            CALL PKSER(XA,XB,C,TT,SS,IERR)
            DD=DSQRT(XB*XB+C*C)
            S=DATAN(XA/C)/(C*RTPI)-SS*XA/(PI*DD)
            ANSI3=S-SUM
            KFORM=2
            RETURN
          ELSE
            TT=0.0D0
            CALL PKSER(XB,C,XA,TT,SS,IERR)
            DD=DSQRT(XA*XA+C*C)
```

```
          SS=SS*XB/(DD*PI)
          S=XA*SS/C
          CALL PKSER(XA,C,XB,TT,SS,IERR)
          DD=DSQRT(XB*XB+C*C)
          SS=SS*XA/(DD*PI)
          ANSI3=S+XB*SS/C-SUM
          KFORM=2
          RETURN
        ENDIF
      ENDIF
 100  CONTINUE
C     TEST  FOR LARGE C FORMULA
      CT2=CT*CT
      A2B2=A*A+B*B
      IF (CT2.GT.5.0D0) THEN
        FG=CT2-4.0D0
      ELSE
        FG=1.0D0
      ENDIF
      IF ((A2B2.LT.0.5D0*C*C/FG)) THEN
C     JERR=0 MEANS COMPUTATION COMPLETED
C     JERR=1 MEANS NO CONVERGENCE IN LARGEC IN 100 TERMS
C     JERR=2 MEANS AN UNDER FLOW
C     JERR=3 MEANS SERIES ABORTED BECAUSE TERMS INCREASE IN VALUE IN
C            LARGEC
        CALL I3LARGEC(A,B,C,T,ANSI3,JERR)
        IF (JERR.EQ.0) THEN
          KFORM=3
          RETURN
        ENDIF
        IF (JERR.EQ.2) THEN
          KFORM=3
          ANSI3=0.0D0
          IERR=3
          RETURN
        ENDIF
C     COMPUTE BY MAIN METHOD FOR CASES I AND II IF JERR=1 OR JERR=3
      ENDIF
      T2=T*T
      X2=XA*XA+XB*XB+C*C
      ARG=T2*X2
      FNH=1.5D0
      KKODE=1
      M=85
      REL=0.50D-15
      KERR=0
      NZ=0
      CALL DHEXINT(ARG, FNH, KKODE, M, REL, ENH, NZ, KERR)
      IF(KERR.EQ.2) THEN
        ANSI3=0.0D0
        IERR=6
        RETURN
      ENDIF
      IF(C.LE.XA) THEN
C       CASE I, C<=A
        ALFA2=XA*XA+C*C
        ALFA=DSQRT(ALFA2)
        ALT2=ALFA2*T2
        KKODE=1
        NZ=0
        ERFAT=DRERF(AT,KKODE,NZ)
        CALL INTEGI5(C,XB,T,ANSI5,IERR5)
        IF ((IERR5.EQ.5).OR.(IERR5.EQ.6)) THEN
          ANSI3=0.0D0
          IERR=5
          RETURN
        ENDIF
```

```
        SS=ERFAT*ANSI5
        CALL SSERI3(XA,XB,C,T,SER,ENH,IERRS)
        IF(IERRS.EQ.2) THEN
          ANSI3=0.0D0
          IERR=4
          RETURN
        ENDIF
        SS=SS+SER
        CRTPI=C*RTPI
        IF(ALT2.LE.3.0D0) THEN
          S=DATAN(XA/C)/CRTPI
          KKODE=1
          NZ=0
          ERFCT=DRERF(CT,KKODE,NZ)
          S=S-0.5D0*RTPI*ERFCT/C+PHIZ(ALT2)*DATAN(C/XA)/CRTPI
          CALL SSERI5(C,XA,T,SS5,IERRS)
          IF(IERRS.EQ.1) THEN
            ANSI3=0.0D0
            IERR=3
            KFORM=4
            RETURN
          ENDIF
          IF(IERRS.EQ.2) THEN
            ANSI3=0.0D0
            IERR=5
            KFORM=4
          ENDIF
          S=S+0.5D0*ALFA*T2*SS5/RTPI
          ANSI3=S-SS
          KFORM=4
          RETURN
        ELSE
          KKODE=2
          NZ=0
          ERFCT=DRERF(CT,KKODE,NZ)
          IF (NZ.NE.0) THEN
             ANSI3=0.0D0
             IERR=3
             KFORM=5
             RETURN
          ENDIF
          S=0.5D0*RTPI*ERFCT/C
          CALL INTEGI5(C,XA,T,ANSI5,IERR5)
          IF ((IERR5.EQ.5).OR.(IERR5.EQ.6)) THEN
            ANSI3=0.0D0
            IERR=5
            KFORM=5
            RETURN
          ENDIF
          SD=S-ANSI5
          IF (DABS(SD).GT.1.0D-275) THEN
            ANSI3=SD-SS
          ELSE
            ANSI3=0.0D0
            IERR=3
          ENDIF
          KFORM=5
          RETURN
        ENDIF
      ENDIF
      IF(C.LE.XB) THEN
C       CASE I, A<C<=B
        BETA2=XB*XB+C*C
        BETA=DSQRT(BETA2)
        BET2=BETA2*T2
        KKODE=1
        NZ=0
```

```
        ERFAT=DRERF(AT,KKODE,NZ)
        CALL INTEGI5(XA,C,T,ANSI5,IERR5)
        IF ((IERR5.EQ.5).OR.(IERR5.EQ.6)) THEN
          ANSI3=0.0D0
          IERR=5
          RETURN
        ENDIF
        SS=XA*ANSI5/C
        CALL SSERI3(XA,XB,C,T,SER,ENH,IERRS)
        IF(IERRS.EQ.2) THEN
          ANSI3=00D0
          IERR=4
          RETURN
        ENDIF
        SS=SS-SER
        CRTPI=C*RTPI
        IF(BET2.LE.3.0D0) THEN
          S=DATAN(XB/C)/CRTPI
          KKODE=1
          NZ=0
          ERFCT=DRERF(CT,KKODE,NZ)
          S=S-0.5D0*RTPI*ERFCT/C+PHIZ(BET2)*DATAN(C/XB)/CRTPI
          CALL SSERI5(C,XB,T,SS5,IERRS)
          IF(IERRS.EQ.1) THEN
            ANSI3=0.0D0
            IERR=3
            KFORM=6
            RETURN
          ENDIF
          IF(IERRS.EQ.2) THEN
            ANSI3=0.0D0
            IERR=5
            KFORM=6
            RETURN
          ENDIF
          S=S+0.5D0*BETA*T2*SS5/RTPI
          ANSI3=S*ERFAT+SS
          KFORM=6
          RETURN
        ELSE
          KKODE=2
          NZ=0
          ERFCT=DRERF(CT,KKODE,NZ)
          IF (NZ.NE.0) THEN
             ANSI3=0.0D0
             IERR=3
             KFORM=7
             RETURN
          ENDIF
          S=0.5D0*RTPI*ERFCT/C
          IF(S.LT.1.0D-275) THEN
            ANSI3=0.0D0
            IERR=3
            KFORM=7
            RETURN
          ENDIF
          CALL INTEGI5(C,XB,T,ANSI5,IERR5)
          IF ((IERR5.EQ.5).OR.(IERR5.EQ.6)) THEN
            ANSI3=0.0D0
            IERR=5
            KFORM=7
            RETURN
          ENDIF
          IF(SS.EQ.0.0D0) THEN
            SD=(S-ANSI5)*ERFAT
            IF (DABS(SD).LT.1.0D-290) THEN
              ANSI3=0.0D0
```

```
              IERR=3
            ELSE
              ANSI3=SD
            ENDIF
          ELSE
            ANSI3=(S-ANSI5)*ERFAT+SS
          ENDIF
          KFORM=7
          RETURN
        ENDIF
      ENDIF
C     CASE II
      KKODE=2
      NZ=0
      ERFCT=DRERF(CT,KKODE,NZ)
      IF (ERFCT.LT.1.0D-275) THEN
        ANSI3=0.0D0
        IERR=3
        KFORM=8
        RETURN
      ENDIF
      KKODE=1
      NZ=0
      ERFAT=DRERF(AT,KKODE,NZ)
      KKODE=1
      NZ=0
      ERFBT=DRERF(BT,KKODE,NZ)
      S=0.5D0*RTPI*ERFBT*ERFCT/C
      IF (S.LT.1.0D-275) THEN
        ANSI3=0.0D0
        IERR=3
        KFORM=8
        RETURN
      ENDIF
      CALL INTEGI5(XB,C,T,ANSI5,IERR5)
      IF ((IERR5.EQ.5).OR.(IERR5.EQ.6)) THEN
        ANSI3=0.0D0
        IERR=5
        KFORM=8
        RETURN
      ENDIF
      S=(S+XB*ANSI5/C)*ERFAT
      IF(S.LT.1.0D-275) THEN
        ANSI3=0.0D0
        IERR=3
        KFORM=8
        RETURN
      ENDIF
      CALL INTEGI5(XA,C,T,ANSI5,IERR5)
      IF ((IERR5.EQ.5).OR.(IERR5.EQ.6)) THEN
        ANSI3=0.0D0
        IERR=5
        KFORM=8
        RETURN
      ENDIF
      S=S+XA*ANSI5*ERFBT/C
      CALL SSERI3(XB,C,XA,T,SS,ENH,IERRS)
      IF(IERRS.EQ.2) THEN
        ANSI3=00D0
        IERR=4
        KFORM=8
        RETURN
      ENDIF
      S=S+XA*SS/C
      CALL SSERI3(XA,C,XB,T,SS,ENH,IERRS)
      IF(IERRS.EQ.2) THEN
        ANSI3=00D0
```

```
          IERR=4
          KFORM=8
          RETURN
        ENDIF
        ANSI3=S+XB*SS/C
        KFORM=8
        RETURN
        END
        SUBROUTINE J3SER(A,B,C,T,FJ3)
C       J3SER EVALUATES THE SERIES FORM OF J3=INT ON (T,INF) OF
C       EXP(-C*C*W*W)*ERFC(A*W)*ERFC(B*W), DEVELOPED IN FOLDER7
C
        IMPLICIT DOUBLE PRECISION (A-H,O-Z)
        DIMENSION ENH(85)
        DATA RTPI /1.772453850905516D0/
        IF (T.EQ.0.0D0) GOTO 55
        T2=T*T
        X2=A*A+B*B+C*C
        ARG=T2*X2
        FNH=1.5D0
        KODE=1
        M=85
        REL=0.50D-14
        CALL DHEXINT(ARG, FNH, KODE, M, REL, ENH, NZ, IERR)
 55     CONTINUE
        XA=MIN(A,B)
        XB=MAX(A,B)
        IF (C.LE.XB) THEN
          CALL SSERI3(XA,XB,C,T,SS,ENH,IERR)
          CALL INTEGI5(C,XB,T,ANSI5,IERR)
          ARG=XA*T
          KODE=2
          ERFA=DRERF(ARG,KODE,NZ)
          FJ3=ERFA*ANSI5-SS
        ELSE
          CALL SSERI3(XB,C,XA,T,SS,ENH,IERR)
          CALL INTEGI5(XA,C,T,ANSI5,IERR)
          ARG=XB*T
          KODE=2
          ERFB=DRERF(ARG,KODE,NZ)
          FJ3A=ERFB*ANSI5-SS
          CALL SSERI3(XA,C,XB,T,SS,ENH,IERR)
          CALL INTEGI5(XB,C,T,ANSI5,IERR)
          ARG=XA*T
          KODE=2
          ERFA=DRERF(ARG,KODE,NZ)
          FJ3B=ERFA*ANSI5-SS
          ARG=C*T
          KODE=2
          ERFC=DRERF(ARG,KODE,NZ)
          FJ3=0.5D0*RTPI*ERFA*ERFB*ERFC/C - FJ3A*(XA/C) - FJ3B*(XB/C)
        ENDIF
        RETURN
        END
        DOUBLE PRECISION FUNCTION PHIZ(Z)
C
C       PHIZ EVALUATES THE PHI FUNCTION = 1-EXP(-Z) TO GET SIGNIFICANT
C       DIGITS FOR SMALL Z. REF: FOLDER 5
C
        IMPLICIT DOUBLE PRECISION (A-H,O-Z)
        HZ=0.5D0*Z
C       EXP(-40) IS APPROX. 10**(-17.2) AND PHIZ=1.0
        IF(HZ.GT.20.0D0) THEN
          PHIZ=1.0D0
        ELSE
          S=DEXP(-HZ)*DSINH(HZ)
          PHIZ=S+S
```

```
      ENDIF
      RETURN
      END
      SUBROUTINE PKSER(A,B,C,T,SUM,IERR)
C
C     EVALUATES THE PK SERIES FOR S(A,B,C,T)
C
C     IERR=0 NORMAL RETURN
C     IERR=1 UNDERFLOW, SUM=0.0D0 RETURNED
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DATA RTPI /1.772453850905516D0/
      IERR=0
      REL=0.50D-16
      A2=A*A
      B2=B*B
      C2=C*C
      D2=C2+B2
      D=DSQRT(D2)
      X2=D2+A2
      X=DSQRT(X2)
      ARG=T*X
      KKODE=2
      NZ=0
      ERFX=DRERF(ARG,KKODE,NZ)
      IF(NZ.NE.0) THEN
        IERR=1
        SUM=0.0D0
        RETURN
      ENDIF
      XPD=X+D
      S=1.0D0
      R=C2/D2
      CK=1.0D0
      AK=0.5D0
      BK=2.0D0
      Y=A/XPD
      Y=Y*Y
      DO 10 K=1,60
        PK=FHYPER(K,Y)
        CK=CK*AK*R/BK
        TRM=CK*PK
        S=S+TRM
        IF(DABS(CK).LE.REL*DABS(S)) GOTO 20
        AK=AK+1.0D0
        BK=BK+1.0D0
   10 CONTINUE
   20 CONTINUE
      S=S*RTPI/XPD
      SUM=S*ERFX
      RETURN
      END
      DOUBLE PRECISION FUNCTION FHYPER(K,Y)
C
C     FHYPER(K,Y) EVALUATES THE HYPERGEOMETRIC POLYNOMIAL F(-K,1,K+2;Y)
C     FOR THE S SUM IN J3 IN FOLDER 7
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      S=1.0D0
      AK=DBLE(FLOAT(K))
      BK=AK+2.0D0
      AK=-AK
      CK=1.0D0
      REL=0.50D-15
      DO 10 J=1,K
        CK=CK*AK*Y/BK
        S=S+CK
```

```
        IF (DABS(CK).LE.REL*DABS(S)) GOTO 20
        AK=AK+1.0D0
        BK=BK+1.0D0
   10 CONTINUE
   20 CONTINUE
      FHYPER=S
      RETURN
      END
      SUBROUTINE I3PWRSER(A,B,C,T,SUM,IERR)
C
C     THIS ROUTINE COMPUTES THE POWER SERIES OF THE INT ON (0,T) OF
C     EXP(-C*C*T*T)*ERF(A*T)*ERF(B*T) (REF: FOLDER 7b)
C
C     IERR=1 MEANS NO CONVERGENCE IN POWER SERIES AFTER 100 TERMS
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION CTRM(100),ATRM(100),BTRM(100),CONVAB(100)
      DATA PI /3.14159265358979D0/
      IERR=0
      REL=0.50D-15
      IF(T.EQ.0.0D0) THEN
        SUM=0.0D0
        RETURN
      ENDIF
      FK=1.0D0
      CT=C*T
      CT2=CT*CT
      CTRM(1)=1.0D0
      AT=A*T
      AT2=AT*AT
      BT=B*T
      BT2=BT*BT
      AK=3.0D0
      SK=5.0D0
      AC=1.0D0
      ATRM(1)=1.0D0
      BC=1.0D0
      BTRM(1)=1.0D0
      CONVAB(1)=1.0D0
      TSUM=1.0D0/3.0D0
      DO 10 K=2,100
        AC=-AC*AT2/FK
        ATRM(K)=AC/AK
        BC=-BC*BT2/FK
        BTRM(K)=BC/AK
        CTRM(K)=-CTRM(K-1)*CT2/FK
C     THE POWERS OF T^2 ARE DISTRIBUTED AMONG THE CONVOLUTIONS TO SCALE
C     A^2, B^2 AND C^2 SINCE A, B OR C COULD BE LARGE IF T IS SMALL
C     CONVOLUTE ERF(AT) AND ERF(BT) COEFFICIENTS FOR PRODUCT COEFFIENTS
C     CONVAB(*)
        S=0.0D0
        DO 15 J=1,K
          S=S+ATRM(J)*BTRM(K-J+1)
   15   CONTINUE
        CONVAB(K)=S
C     CONVOLUTE EXP COEFFICIENTS WITH PRODUCT OF ERF COEFFICIENTS FOR
C     THE COEFFIENTS OF THE PRODUCT OF THE ERF FUNCTIONS AND THE EXP
C     FUNCTION
        SS=0.0D0
        DO 20 J=1,K
          SS=SS+CTRM(J)*CONVAB(K-J+1)
   20   CONTINUE
C     SK IS THE FACTOR PRODUCED BY INTEGRATION OF THE FINAL SERIES IN
C     T^2.  SK=2*K+3=POWER OF T IN THE FINAL
C     SERIES. T^3 AND THE COEFFICIENTS OF THE ERF SERIES ARE FACTORED
C     OUT AND APPLIED TO THE FINAL SUM.
        SS=SS/SK
```

```
          TSUM=TSUM+SS
          IF (DABS(SS).LT.TSUM*REL) GOTO 55
          AK=AK+2.0D0
          SK=SK+2.0D0
          FK=FK+1.0D0
   10   CONTINUE
        IERR=1
C        WRITE (7,303)
C   303 FORMAT('NO CONVERGENCE IN I3PWRSER.FOR IN 100 TERMS')
   55   CONTINUE
C       SUM=((4/PI)*A*B*T**3)*TSUM. TSUM=POWER SERIES IN T**2
        TSUM=TSUM+TSUM
        TSUM=TSUM+TSUM
        SUM=TSUM*AT*BT*T/PI
        RETURN
        END
        SUBROUTINE I3LARGEC(A,B,C,T,SUM,IERR)
C
C       THIS ROUTINE COMPUTES THE LARGE C SERIES OF I3 OF FOLDER 7d.
C
C       IERR=1 MEANS NO CONVERGENCE IN 100 TERMS
C
        IMPLICIT DOUBLE PRECISION (A-H,O-Z)
        DIMENSION ATRM(100),BTRM(100),BCOEF(100)
        DATA RTPI, PI /1.772453850905516D0, 3.14159265358979D0/
        REL=0.50D-15
        IERR=0
C       ZERO OUT THE BC ARRAY FOR THE BINOMIAL COEFFIENTS, SET BC(1)=1.0D0
        BCOEF(1)=1.0D0
        DO 25 J=2,100
          BCOEF(J)=0.0D0
   25   CONTINUE
        FK=1.0D0
        CT=C*T
        CT2=CT*CT
        IF (A.GT.B) THEN
          AT=A/C
          BT=B/C
        ELSE
          AT=B/C
          BT=A/C
        ENDIF
        AT2=AT*AT
        BT2=BT*BT
        AK=3.0D0
        AC=1.0D0
        ATRM(1)=1.0D0
C       FHK=GAM(I+0.5), I=1,2
        AHK=0.5D0
C       FHK=GAM(3/2)
        FHK=AHK*RTPI
        BC=1.0D0
        BTRM(1)=1.0D0
C       INCGAM(1/2,CT2)/GAM(1/2)=ERFC(CT)
        KODE=2
        GHK=DRERF(CT,KODE,NZ)
        IF (NZ.NE.0) THEN
          SUM=0.0D0
          IERR=2
          RETURN
        ENDIF
        EX=DEXP(-CT2)
        GHKTRM=CT*EX/FHK
C       INCGAM(3/2,CT2)/GAM(3/2) FROM INCGAM(1/2,CT2)/GAM(1/2)
        GHK=GHK+GHKTRM
C       FIRST TERM OF SERIES
        TRM=FHK*GHK
```

```
      TSUM=TRM
C     UPDATE INCGAM TERMS FOR INDEX=3/2+I (I=1 FOR 2ND TERM OF SERIES)
      AHK=AHK+1.0D0
C     FHK=GAM(3/2+I)/I!, I=INDEX OF SERIES = K-1
      FHK=FHK*AHK/FK
      GHKTRM=GHKTRM*CT2/AHK
      GHK=GHK+GHKTRM
      DO 10 K=2,100
C     GENERATE THE POWERS DIVIDED BY THE FACTOR (2*I+1), I=1,... FOR
C     ERF(AW) AND ERF(BW)
        AC=-AC*AT2
        ATRM(K)=AC/AK
        BC=-BC*BT2
        BTRM(K)=BC/AK
C     GENERATE BINOMIAL COEFFICIENTS FOR PASCAL TRIANGLE FOR EACH K
C     THEY ARE GENERATED FROM RIGHT TO LEFT SO THAT THE NEW VALUES
C     STORED IN THE BCOEF VECTOR WILL NOT OVERWRITE THE OLD VALUES NEEDED
C     FOR THE REMAINING INDICES.
        DO 35 M=2,K
          J=K-M+2
          BCOEF(J)=BCOEF(J)+BCOEF(J-1)
  35    CONTINUE
C     THE POWERS OF C^2 ARE DISTRIBUTED AMONG THE CONVOLUTIONS TO SCALE
C     A^2, B^2. THAT IS, AT2=(A/C)**2, BT2=(B/C)**2
C     ERF(AW) AND ERF(BW) CONVOLUTION CONTAIN BINOMIAL COEFFICIENTS
C     WHICH ARE GENERATED BY THE PASCAL TRIANGLE. ATERM AND BTERM ARRAYS
C     CONTAIN THE POWERS DIVIDED BY THE FACTOR (2*I+1), I=K-1
        S=0.0D0
        DO 15 J=1,K
          S=S+ATRM(J)*BCOEF(J)*BTRM(K-J+1)
  15    CONTINUE
        TRM1=S*FHK*GHK
        IF (DABS(TRM1).GT.DABS(TRM)) THEN
C          WRITE (7,305) K,TRM,TRM1
C          SS=(AT2+BT2)**(K-1)
C          WRITE (7,306) SS,S,FHK,GHK
C 306      FORMAT(5X,4D14.6)
C 305      FORMAT(I5,2D14.6)
C 307      FORMAT('ABORT IN LARGEC FOR ',4D14.6)
          IF(K.GT.10) THEN
             SUM=0.0D0
             IERR=3
C             WRITE(7,307)A,B,C,T
             RETURN
          ENDIF
        ENDIF
        TRM=TRM1
        TSUM=TSUM+TRM
        IF (DABS(TRM).LE.DABS(TSUM)*REL) GOTO 55
        AK=AK+2.0D0
        FK=FK+1.0D0
        AHK=AHK+1.0D0
        FHK=FHK*AHK/FK
        GHKTRM=GHKTRM*CT2/AHK
        GHK=GHK+GHKTRM
  10  CONTINUE
C      WRITE (7,303)
C 303 FORMAT('NO CONVERGENCE IN SUBROUTINE I3LARGEC IN 100 TERMS')
      IERR=1
  55  CONTINUE
C      WRITE (7,304) K
C 304 FORMAT(I5)
C     SUM=((2.0/PI)*A*B/C**3)*TSUM.
      TSUM=(TSUM+TSUM)/PI
      SUM=TSUM*(A/C)*(B/C)/C
      RETURN
      END
```

```
      SUBROUTINE SSERI3(A,B,C,T,SUM,ENH,IERR)
C
C     SSERI3 EVALUATES THE SERIES FOR S(A,B,C,T) WHERE
C     A,C <=B IN FOLDER 7 FOR I3 AND J3=I3C
C
C     IERR=0 NORMAL RETURN
C     IERR=1 UNDERFLOW, SUM=0.0D0 RETURNED
C     IERR=2 NO CONVERGENCE IN THE SSERI3 SERIES IN 85 TERMS
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION ENH(85),BC(86),PAM(86),PCM(86)
      DATA PI /3.14159265358979D0/
      IERR=0
      REL=0.50D-15
      CALL PKSER(A,B,C,T,PKS,IERR)
      IF (IERR.EQ.1) THEN
        SUM=0.0D0
        RETURN
      ENDIF
      A2=A*A
      B2=B*B
      C2=C*C
      D2=C2+B2
      D=DSQRT(D2)
      IF(T.EQ.0.0D0) THEN
        SUM=(PKS/PI)*(A/D)
        RETURN
      ENDIF
      X2=D2+A2
      X=DSQRT(X2)
      S=1.0D0
C     ZERO OUT THE BC ARRAY FOR THE BINOMIAL COEFFIENTS, SET BC(1)=1.0D0
      BC(1)=1.0D0
      DO 25 J=2,81
        BC(J)=0.0D0
   25 CONTINUE
      SS=ENH(1)
      PA=A2/X2
      PC=C2/X2
      PAM(1)=1.0D0
      PCM(1)=1.0D0
      CK=1.0D0
      AK=1.5D0
      BK=2.0D0
C     M=NUMBER OF THE TERM IN THE FORMULA, M=MINDEX+1
      DO 30 M=2,85
C     GENERATE BINOMIAL COEFFICIENTS FOR PASCAL TRIANGLE FOR EACH M
C     THEY ARE GENERATED FROM RIGHT TO LEFT SO THAT THE NEW VALUES
C     STORED IN THE BC VECTOR WILL NOT OVERWRITE THE OLD VALUES NEEDED
C     FOR THE REMAINING INDICES.
        DO 35 K=2,M
          J=M-K+2
          BC(J)=BC(J)+BC(J-1)
   35   CONTINUE
        SM=0.0D0
        DK=1.0D0
        PAM(M)=PAM(M-1)*PA
        PCM(M)=PCM(M-1)*PC
        MP1=M+1
        DO 40 K=1,M
          SM=SM+BC(K)*PAM(MP1-K)*PCM(K)/DK
          DK=DK+2.0D0
   40   CONTINUE
        CK=CK*AK/BK
        TRM=CK*SM*ENH(M)
        SS=SS+TRM
        IF (DABS(TRM).LE.REL*DABS(SS)) GOTO 55
```

```
        AK=AK+1.0D0
        BK=BK+1.0D0
   30 CONTINUE
      IERR=2
C      WRITE(7,45)
C  45  FORMAT('NO CONVERGENCE IN SSERI3 AFTER 85 TERMS')
   55 CONTINUE
      SS=0.5D0*SS*T*(D/X)
   60 CONTINUE
      S=PKS-SS
      SUM=(S/PI)*(A/D)
      RETURN
      END
      SUBROUTINE INTEGI5(A,B,X,ANSI5,IERR)
C
C     DONALD E. AMOS   DECEMBER, 2000
C-----------------------------------------------------------------------
C     A DOUBLE PRECISION ROUTINE
C-----------------------------------------------------------------------
C
C     INTEGI5 COMPUTES THE I INTEGRAL OF FOLDER 5,
C
C            ANSI5=INT ON (X,INF) OF EXP(-A*A*W*W)*ERFC(B*W),
C
C     WHERE X.GE.0 AND PARAMETERS A AND B CANNOT BE NEGATIVE OR BOTH
C     ZERO AT THE SAME TIME. THIS INTEGRAL IS DENOTED BY I5 IN OTHER
C     FOLDERS.
C
C     ERROR CONDITIONS:
C
C     IERR=0  NORMAL RETURN, COMPUTATION COMPLETED
C     IERR=1  X IS NEGATIVE, ANSI5=0.0D0 RETURNED
C     IERR=2  BOTH A=0 AND B=0 AND I5 DOES NOT EXIST. ANSI5=0.0D0
C             RETURNED
C     IERR=3  EITHER A OR B OR BOTH ARE NEGATIVE. ANSI5=0.0D0
C             RETURNED
C     IERR=4  UNDERFLOW, ANSI5=0.0D0 RETURNED
C
C             CONVERGENCE OR COMPUTATIONAL PROBLEMS (NOT LIKELY)
C
C     IERR=5  G SERIES NOT CONVERGENT IN 55 TERMS. REDUCED ACCURACY
C             PROBABLE.
C     IERR=6  S SERIES NOT CONVERGENT IN 55 TERMS. REDUCED ACCURACY
C             PROBABLE.
C
C     CONDITIONS IERR = 5 OR 6 SHOULD ONLY OCCUR IF THE MACHINE FAILS,
C     SO THERE IS GENERALLY NO REASON TO TEST FOR THEM.
C
C     CALLS ROUTINES: GSERI5, SSERI5, DHEXINT, DRERF, DIERFC
C-----------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DATA RTPI       /1.772453850905516D0/
      IERR=0
      IF(X.LT.0.0D0) THEN
        ANSI5=0.0D0
        IERR=1
        RETURN
      ENDIF
      IF((A.EQ.0.0D0).AND.(B.EQ.0.0D0)) THEN
C       I5 IS NOT DEFINED FOR BOTH A=0 AND B=0
        ANSI5=0.0D0
        IERR=2
        RETURN
      ENDIF
      IF((A.LT.0.0D0).OR.(B.LT.0.0D0)) THEN
C       I5 IS NOT DEFINED FOR NEGATIVE PARAMETERS A AND B
        ANSI5=0.0D0
```

```
          IERR=3
          RETURN
        ENDIF
        IF (A.EQ.0.0D0) THEN
C         FOR A=0, ANSI5=IERFC(B*X)/B
          ARG=B*X
          KODE=1
          CALL DIERFC(ARG,KODE,ANS,KERR)
          IF(KERR.EQ.1) THEN
            ANSI5=0.0D0
            IERR=4
            RETURN
          ENDIF
          ANSI5=ANS/B
        ENDIF
        IF (B.EQ.0.0D0) THEN
C         FOR B=0, ANSI5=RTPI*ERFC(A*X)/(2*A)
          ARG=A*X
          KODE=2
          NZ=0
          ERFA=DRERF(ARG,KODE,NZ)
          IF(NZ.NE.0) THEN
            ANSI5=0.0D0
            IERR=4
            RETURN
          ENDIF
          ANSI5=0.5D0*RTPI*ERFA/A
          RETURN
        ENDIF
        IF (A.LE.B) THEN
          CALL GSERI5(A,B,X,ANSI5,JERR)
          IF(JERR.EQ.1) THEN
            IERR=4
            ANSI5=0.0D0
            RETURN
          ENDIF
          IF(JERR.EQ.2) THEN
            IERR=5
          ENDIF
        ELSE
          ARTPI=A*RTPI
          D2=A*A+B*B
          D=DSQRT(D2)
          X2=X*X
          D2X2=D2*X2
          IF (D2X2.LE.3.0D0) THEN
            CALL SSERI5(B,A,X,ANS,JERR)
            IF(JERR.EQ.1) THEN
              IERR=4
              ANSI5=0.0D0
              RETURN
            ENDIF
            IF(JERR.EQ.2) THEN
              IERR=6
            ENDIF
            ARG=A*X
            KODE=1
            ERFA=DRERF(ARG,KODE,NZ)
            ARG=B*X
            ERFB=DRERF(ARG,KODE,NZ)
            ARG=0.5D0*D2X2
            S=(DATAN(A/B)+2.0D0*DEXP(-ARG)*SINH(ARG)*DATAN(B/A))/ARTPI
            S=S-0.5D0*RTPI*(ERFA+ERFB-ERFA*ERFB)/A
            ANSI5=S+(0.5D0*B*D*X2/ARTPI)*ANS
          ELSE
            CALL GSERI5(B,A,X,ANS,JERR)
            IF(JERR.EQ.1) THEN
```

```
            IERR=4
            ANSI5=0.0D0
            RETURN
          ENDIF
          IF(JERR.EQ.2) THEN
            IERR=5
          ENDIF
          ARG1=A*X
          ARG2=B*X
          XX=ARG1*ARG1+ARG2*ARG2
          IF (XX.GT.670.0D0) THEN
            IERR=4
            ANSI5=0.0D0
            RETURN
          ENDIF
          KODE=2
          ERFCA=DRERF(ARG1,KODE,NZ)
          ERFCB=DRERF(ARG2,KODE,NZ)
          ANSI5=0.5D0*RTPI*ERFCA*ERFCB/A-(B/A)*ANS
        ENDIF
      ENDIF
      RETURN
      END
      SUBROUTINE INTEGJ5(A,B,X,ANSJ5,IERR)
C
C     DONALD E. AMOS    SEPTEMBER, 2001
C-----------------------------------------------------------------------
C     A DOUBLE PRECISION ROUTINE
C-----------------------------------------------------------------------
C
C     INTEGJ5 COMPUTES THE J FUNCTION OF FOLDER5:
C
C           ANSJ5 = INT ON (X,INF) OF EXP(-A^2)*ERF(B*X)
C
C     WHERE A.GT.0.0D0, B.GE.0.0D0, X.GE.0.0D0
C
C     IERR =0  IS A NORMAL RETURN.
C     IERR =1  UNDERFLOW IN GSERI5 OR SSERI5
C     IERR =2  ONE OR MORE OF THE PARAMETERS A, B, OR X IS NEGATIVE
C              OR A IS ZERO.
C-----------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DATA RTPI       /1.772453850905516D0/
      IERR=0
      IF ((A.LE.0.0D0).OR.(B.LT.0.0D0).OR.(X.LT.0.0D0)) THEN
        IERR=2
        ANSJ5=0.0D0
        RETURN
      ENDIF
      IF(B.EQ.0.0D0) THEN
        ANSJ5=0.0D0
        RETURN
      ENDIF
      AX=A*X
      BX=B*X
      ARTPI=A*RTPI
      IF ((AX.LE.1.0D0).AND.(BX.LE.1.0D0)) THEN
        CALL V5PSER(A,B,X,SS)
        ANSJ5=DATAN(B/A)/ARTPI-SS
        RETURN
      ENDIF
      IF (A.LE.B) THEN
        D2=A*A+B*B
        D=DSQRT(D2)
        X2=X*X
        D2X2=D2*X2
        IF (D2X2.LE.3.0D0) THEN
```

```
          KODE=1
          ARG=A*X
          ERFA=DRERF(ARG,KODE,NZ)
          CALL SSERI5(A,B,X,SS,KERR)
          ARG=0.5D0*D2X2
          S=2.0D0*DEXP(-ARG)*DSINH(ARG)*DATAN(A/B)/ARTPI
         ANSJ5=S+DATAN(B/A)/ARTPI-0.5D0*RTPI*ERFA/A+(0.5D0*X2*D/RTPI)*SS
        ELSE
          CALL GSERI5(A,B,X,SS,KERR)
          KODE=2
          ARG=A*X
          ERFCA=DRERF(ARG,KODE,NZ)
          TRM=0.5D0*RTPI*ERFCA/A
          IF(KERR.EQ.1) THEN
            IF (ARG*ARG.LT.630.0D0) THEN
              ANSJ5=TRM
            ELSE
              IERR=1
              ANSJ5=0.0D0
            ENDIF
          ELSE
            ANSJ5=TRM-SS
          ENDIF
        ENDIF
      ELSE
        CALL GSERI5(B,A,X,SS,KERR)
        NZ=0
        ARG=B*X
        KODE=1
        ERFB=DRERF(ARG,KODE,NZ)
        NZ=0
        ARG=A*X
        KODE=2
        ERFCA=DRERF(ARG,KODE,NZ)
        TRM=0.5D0*RTPI*ERFCA*ERFB/A
        IF(KERR.EQ.1) THEN
          IF(NZ.EQ.0) THEN
            TM=DLOG(TRM)
            IF(DABS(TM).LT.630.0D0) THEN
              ANSJ5=TRM
            ELSE
              IERR=1
              ANSJ5=0.0D0
            ENDIF
          ELSE
            IERR=1
            ANSJ5=0.0D0
          ENDIF
        ELSE
          ANSJ5=TRM+(B/A)*SS
        ENDIF
      ENDIF
      RETURN
      END
      SUBROUTINE INTEGV5(A,B,X,ANSV5,IERR)
C
C     DONALD E. AMOS, JANUARY, 2003
C-----------------------------------------------------------------------
C     A DOUBLE PRECISION ROUTINE
C-----------------------------------------------------------------------
C
C     INTEGV5 COMPUTES THE V5 INTEGRAL OF FOLDER 5
C
C         ANSV5=INT ON (0,X) OF EXP(-(A*W)**2)*ERF(B*W)
C
C     WHERE A.GE.0.0D0, B.GE.0.0D0, X.GE.0.0D0.
C
```

```
C     IERR=0 IS A  NORMAL RETURN
C     IERR=1 MEANS ONE OR MORE OF THE PARAMETERS A, B, OR X IS NEGATIVE
C            ANSV5=0.0D0 RETURNED
C     IERR=2 IS A NON-CONVERGENCE ERROR FROM GSERI5 OR SSERI5
C
C     UNDERFLOWS DO NOT CONTRIBUTE TO THE ANSWER AND ARE NOT FLAGGED
C-----------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DATA RTPI       /1.772453850905516D0/
      IERR=0
      IF((A.LT.0.0D0).OR.(B.LT.0.0D0).OR.(X.LT.0.0D0)) THEN
        IERR=1
        ANSV5=0.0D0
        RETURN
      ENDIF
      IF(X.EQ.0.0D0) THEN
         ANSV5=0.0D0
         RETURN
      ENDIF
      IF(B.EQ.0.0D0) THEN
        ANSV5=0.0D0
        RETURN
      ENDIF
      IF(A.EQ.0.0D0) THEN
        BX=B*X
        IF(BX.GT.6.0D0) THEN
          ANSV5=(BX-1.0D0/RTPI)/B
          RETURN
        ENDIF
        KKODE=1
        ERFBX=DRERF(BX,KKODE,NZ)
        ARG=0.5D0*BX*BX
        ANSV5=X*ERFBX-2.0D0*DEXP(-ARG)*DSINH(ARG)/(B*RTPI)
        RETURN
      ENDIF
      AX=A*X
      BX=B*X
      IF ((AX.LE.1.0D0).AND.(BX.LE.1.0D0)) THEN
        CALL V5PSER(A,B,X,ANSV5)
        RETURN
      ENDIF
      ARTPI=A*RTPI
      IF (A.LE.B) THEN
        D2=A*A+B*B
        D=DSQRT(D2)
        X2=X*X
        D2X2=D2*X2
        IF (D2X2.LE.3.0D0) THEN
          KKODE=1
          ARG=A*X
          ERFA=DRERF(ARG,KKODE,NZ)
          CALL SSERI5(A,B,X,SS,IERR)
          IF(IERR.EQ.1) IERR=0
          IF(IERR.EQ.2) THEN
            ANSV5=0.0D0
            RETURN
          ENDIF
          ARG=0.5D0*D2X2
          S=0.5D0*RTPI*ERFA/A-2.0D0*DEXP(-ARG)*DSINH(ARG)*DATAN(A/B)/
     &ARTPI
          ANSV5=S-(0.5D0*X2*D/RTPI)*SS
          IPATH=1
        ELSE
          CALL GSERI5(A,B,X,SS,IERR)
          IF(IERR.EQ.1) IERR=0
          IF(IERR.EQ.2) THEN
            ANSV5=0.0D0
```

```
            RETURN
          ENDIF
          KKODE=2
          ARG=A*X
          ERFCA=DRERF(ARG,KKODE,NZ)
          ANSV5=-0.5D0*RTPI*ERFCA/A+DATAN(B/A)/ARTPI+SS
          IPATH=2
        ENDIF
      ELSE
        D2=A*A+B*B
        D=DSQRT(D2)
        X2=X*X
        D2X2=D2*X2
        IF (D2X2.LE.3.0D0) THEN
          CALL SSERI5(B,A,X,SS,IERR)
          IF(IERR.EQ.1) IERR=0
          IF(IERR.EQ.2) THEN
            ANSV5=0.0D0
            RETURN
          ENDIF
          ARG=0.5D0*D2X2
          S=2.0D0*DEXP(-ARG)*DSINH(ARG)*DATAN(B/A)/ARTPI
          ARG=A*X
          KKODE=2
          ERFCA=DRERF(ARG,KKODE,NZ)
          ARG=B*X
          KKODE=1
          ERFB=DRERF(ARG,KKODE,NZ)
          S=S-0.5D0*RTPI*(ERFCA*ERFB)/A
          ANSV5=S+(0.5D0*B*D*X2/ARTPI)*SS
          IPATH=3
        ELSE
          CALL GSERI5(B,A,X,SS,IERR)
          IF(IERR.EQ.1) IERR=0
          IF(IERR.EQ.2) THEN
            ANSV5=0.0D0
            RETURN
          ENDIF
          ARG=A*X
          KKODE=2
          ERFCA=DRERF(ARG,KKODE,NZ)
          ARG=B*X
          KKODE=1
          ERFB=DRERF(ARG,KKODE,NZ)
          S=-0.5D0*RTPI*ERFCA*ERFB/A-(B/A)*SS
          ANSV5=DATAN(B/A)/ARTPI+S
          IPATH=4
        ENDIF
      ENDIF
C      WRITE(7,307)IPATH
C 307  FORMAT('PATH= ',I2)
      RETURN
      END
      SUBROUTINE GSERI5(A,B,X,SUM,IERR)
C
C     GSERI5 COMPUTES THE G SUM OF FOLDER 5.
C
C     IERR=0 NORMAL RETURN
C     IERR=1 UNDERFLOW, SUM=0.0D0
C     IERR=2 NO CONVERGENCE IN 55 TERMS OF THE SERIES
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION EN(55)
      DATA RTPI                  /1.772453850905516D0/
      IERR=0
      ARG=(A*A+B*B)
      DEN=ARG
```

```
      ARG=ARG*X*X
      FNH=1.5D0
      KODE=1
      M=55
      TOL=0.5D-15
      NZ=0
      KERR=0
      CALL DHEXINT(ARG, FNH, KODE, M, TOL, EN, NZ, KERR)
      IF(NZ.NE.0) THEN
        IERR=1
        SUM=0.0D0
        RETURN
      ENDIF
      AK=1.0D0
      S=EN(1)
      DH=0.5D0
      Z=A*A/DEN
      FAC=1.0D0
      DO 10 K=1,M-1
        AK=AK*DH/FAC
        AK=AK*Z
        TRM=AK*EN(K+1)
        S=S+TRM
        IF (DABS(TRM).LE.TOL*DABS(S)) GOTO 15
        DH=DH+1.0D0
        FAC=FAC+1.0D0
   10 CONTINUE
       IERR=2
C      WRITE (7,300)
C 300  FORMAT('G-SERIES NOT CONVERGED IN 55 TERMS')
   15 CONTINUE
      SUM=(0.5D0/RTPI)*(S/DSQRT(DEN))
      RETURN
      END
      SUBROUTINE SSERI5(A,B,X,SUM,IERR)
C
C     SSERI5 COMPUTES THE S SUM OF FOLDER 5.
C
C     IERR=0 NORMAL RETURN
C     IERR=1 UNDERFLOW, SUM=0.0D0
C     IERR=2 NO CONVERGENCE IN 55 TERMS OF THE SERIES
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION EN(55)
      IERR=0
      ARG=(A*A+B*B)
      DEN=ARG
      ARG=ARG*X*X
      FNH=0.5D0
      KODE=1
      M=55
      TOL=0.5D-15
      NZ=0
      KERR=0
      CALL DHEXINT(ARG, FNH, KODE, M, TOL, EN, NZ, KERR)
      IF(NZ.NE.0) THEN
        SUM=0.0D0
        IERR=1
        RETURN
      ENDIF
      AK=1.0D0
      S=2.0D0*EN(1)
      DH=0.5D0
      Z=A*A/DEN
      FAC=1.0D0
      DO 10 K=1,M-1
        AK=AK*DH/FAC
```

```
        AK=AK*Z
        TRM=AK*EN(K+1)/(DH+1.0D0)
        S=S+TRM
        IF (DABS(TRM).LE.TOL*DABS(S)) GOTO 15
        DH=DH+1.0D0
        FAC=FAC+1.0D0
  10  CONTINUE
       IERR=2
C      WRITE (7,300)
C 300  FORMAT('S-SERIES NOT CONVERGED IN 55 TERMS')
  15  CONTINUE
      SUM=S
      RETURN
      END
      SUBROUTINE INTEGI6(A,B,T,KODE,ANSI6,IERR)
C
C     DONALD E. AMOS    DECEMBER,2001, FEBRUARY, 2006
C-----------------------------------------------------------------------
C     A DOUBLE PRECISION ROUTINE
C-----------------------------------------------------------------------
C
C     ON KODE=1, INTEGI6 COMPUTES THE INTEGRAL
C
C         ANSI6=INT ON (T,INF) OF ERF(A*W)*ERF(B*W)/W**2
C
C     ON KODE=2, INTEGI6 COMPUTES THE INTEGRAL
C
C         ANSI6=INT ON (T,INF) OF ERFC(A*W)*ERFC(B*W)/W**2
C
C     BY SERIES FROM THE FORMULAE IN FOLDERS 3, 6, AND 15 FOR
C     A, B, AND T NON-NEGATIVE. T=0 IS AN ERROR ON KODE=2.
C
C     IERR=0 IS A NORMAL RETURN
C     IERR=1 IS AN INPUT ERROR, SOME PARAMETER IS NEGATIVE
C     IERR=2 IS AN UNDERFLOW, ANSI6=0.0D0 RETURNED
C-----------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DATA RTPI/ 1.772453850905516D0/
      IERR=0
      IF((A.LT.0.0D0) .OR. (B.LT.0.0D0) .OR. (T.LT.0.0D0)) THEN
        ANSI6=0.0D0
        IERR=1
        RETURN
      ENDIF
      AT=A*T
      BT=B*T
      REL=0.50D-14
      IF(KODE.EQ.1) THEN
        IF ((A.EQ.0.0D0).OR.(B.EQ.0.0D0)) THEN
          ANSI6=0.0D0
          RETURN
        ENDIF
        IF (T.EQ.0.0D0) THEN
          QAB=A/B
          QBA=B/A
          T2=QAB+DSQRT(1.0D0+QAB*QAB)
          T1=QBA+DSQRT(1.0D0+QBA*QBA)
          ANSI6=2.0D0*(A*DLOG(T1)+B*DLOG(T2))/RTPI
          RETURN
        ENDIF
        ICHECK=0
        IF(ICHECK.EQ.0) THEN
C-----------------------------------------------------------------------
C       CASES WHERE ONE OR BOTH OF THE ERF FUNCTIONS HAVE AGUMENTS 6 OR
C       GREATER WHERE THE ERF(*)=1. IN THESE CASES FUNCTION QPI6
C       EVALUATES THE INTEGRALS IN CLOSED FORM. SEE FOLDER 3 FOR THE
C       DERIVATION.
```

```
C-----------------------------------------------------------------------
          XAB=MIN(A,B)
          WSM=MIN(6.0D0/A,6.0D0/B)
          WLM=MAX(6.0D0/A,6.0D0/B)
          IF(T.GT.WSM) THEN
            IF (T.LE.WLM)  THEN
              SS=QPI6(XAB,T,WLM,REL)
            ELSE
              SS=1.0D0/T
            ENDIF
            ANSI6=SS
            RETURN
          ENDIF
        ENDIF
        COEF=2.0D0
      ELSE
        IF(T.EQ.0.0D0) THEN
          IERR=1
          ANSI6=0.0D0
          RETURN
        ENDIF
        X=AT*AT+BT*BT
        IF(X.GT.670.0D0) THEN
          IERR=2
          ANSI6=0.0D0
          RETURN
        ENDIF
        COEF=-2.0D0
      ENDIF
      NZ=0
      ERFAT=DRERF(AT,KODE,NZ)
      ERFBT=DRERF(BT,KODE,NZ)
      S=ERFAT*ERFBT/T
      CALL INTEGP(A,B,T,KODE,REL,PANSAB,IERRP)
      CALL INTEGP(B,A,T,KODE,REL,PANSBA,IERRP)
      ANSI6=S+COEF*(A*PANSAB+B*PANSBA)/RTPI
      RETURN
      END
      DOUBLE PRECISION FUNCTION QPI6(XAB,FL,FU,REL)
C-----------------------------------------------------------------------
C     HERE FU=WLM AND THE TERM 1/WLM FROM THE FORMULA IS ADDED IN TO GET
C     THE ERFC(T2)/WLM FUNCTION IN THE SECOND TERM = TRM2 OF THE FORMULA
C-----------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION EN(5),FN(5)
      DATA RTPI/1.77245385090552D0/
      T1=XAB*FL
      ARG1=T1*T1
      T2=XAB*FU
      ARG2=T2*T2
      KODE=1
      NZ=0
      TRM1=DRERF(T1,KODE,NZ)/FL
      NZ=0
      KODE=2
      TRM2=DRERF(T2,KODE,NZ)/FU
      S=TRM1+TRM2
      N=1
      KODE=1
      M=1
      TOL=REL
      NZ=0
      IERR=0
      CALL DEXINT(ARG1, N, KODE, M, TOL, EN, NZ, IERR)
      CALL DEXINT(ARG2, N, KODE, M, TOL, FN, NZ, IERR)
      C=XAB/RTPI
      QPI6=S+C*(EN(1)-FN(1))
```

```
      RETURN
      END
      SUBROUTINE V5PSER(A,B,X,SUM)
C     POWER SERIES FOR J5
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION COEXP(21),COERF(21),PAX(21),PBX(21)
      DATA RTPI            /1.772453850905516D0/
      DATA (COEXP(J),J=1,21)/
     C0.100000000000000D+01,0.100000000000000D+01,0.500000000000000D+00,
     C0.166666666666667D+00,0.416666666666667D-01,0.833333333333333D-02,
     C0.138888888888889D-02,0.198412698412698D-03,0.248015873015873D-04,
     C0.275573192239859D-05,0.275573192239859D-06,0.250521083854417D-07,
     C0.208767569878681D-08,0.160590438368216D-09,0.114707455977297D-10,
     C0.764716373181982D-12,0.477947733238739D-13,0.281145725434552D-14,
     C0.156192069685862D-15,0.822063524662433D-17,0.411031762331216D-18/
      DATA (COERF(J),J=1,21)/
     C0.100000000000000D+01,0.333333333333333D+00,0.100000000000000D+00,
     C0.238095238095238D-01,0.462962962962963D-02,0.757575757575758D-03,
     C0.106837606837607D-03,0.132275132275132D-04,0.145891690009337D-05,
     C0.145038522231505D-06,0.131225329638028D-07,0.108922210371486D-08,
     C0.835070279514724D-10,0.594779401363764D-11,0.395542951645853D-12,
     C0.246682701026446D-13,0.144832646435981D-14,0.803273501241577D-16,
     C0.422140728880709D-17,0.210785519144214D-18,0.100251649349077D-19/
      TOL=0.5D-17
      AX=A*X
      BX=B*X
      BX2=BX*BX
      AX2=AX*AX
      PAX(1)=1.0D0
      PBX(1)=1.0D0
      DO 5 J=1,20
        PAX(J+1)=-PAX(J)*AX2
        PBX(J+1)=-PBX(J)*BX2
        IF ((DABS(PAX(J+1)).LT.TOL).AND.(DABS(PBX(J+1)).LT.TOL)) THEN
          KL=J+1
          GOTO 6
        ENDIF
 5    CONTINUE
      KL=21
 6    CONTINUE
      CC=BX*X/RTPI
      S=1.0D0
      DKP1=2.0D0
      KP1=3
      DO 10 K=2,KL
        CK=0.0D0
        DO 15 M=1,K
          CK=CK+COERF(M)*PBX(M)*COEXP(KP1-M)*PAX(KP1-M)
 15     CONTINUE
        S=S+CK/DKP1
        DKP1=DKP1+1.0D0
        KP1=KP1+1
 10   CONTINUE
      SUM=CC*S
      RETURN
      END
      SUBROUTINE INTEGP(A,B,T,KODE,REL,PANS,IERR)
C
C     DONALD E. AMOS    DECEMBER, 2001;  JANUARY, 2006
C-----------------------------------------------------------------------
C     A DOUBLE PRECISION ROUTINE
C-----------------------------------------------------------------------
C
C     REFERENCES: FOLDER 11, FOLDER 6
C
C     ON KODE=1, INTEGP COMPUTES THE INTEGRAL
C
```

```
C              PANS =  INT ON (T,INF) OF EXP(-(A*W)^2)*ERF(B*W)/W
C
C      ON KODE=2, INTEGP COMPUTES THE COMPLEMENTARY INTEGRAL
C
C              PANS =  INT ON (T,INF) OF EXP(-(A*W)^2)*ERFC(B*W)/W
C
C      BY SERIES FOR A.GT.0, B.GE.0, T.GT.0. A CAN BE ZERO IF KODE=2,
C      AND T CAN BE ZERO IF KODE=1. REL=0.5D-14 FOR HIGHEST POSSIBLE
C      ACCURACY.
C
C      IERR=0 IS A NORMAL RETURN
C      IERR=1 IS AN INPUT ERROR, SOME PARAMETER IS OUT OF ITS RANGE
C             PANS=0.0D0 RETURNED
C      IERR=2 UNDERFLOW, PANS=0.0D0 RETURNED
C      IERR=3 REQUESTED ERROR NOT MET IN ASYMPTOTIC EXPANSION
C-----------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION EN(60),ENH(60)
      DATA RTPI/ 1.772453850905516D0/
      IERR=0
      IF((KODE.NE.1).AND.(KODE.NE.2)) THEN
        PANS=0.0D0
        IERR=1
        RETURN
      ENDIF
      IF((A.LT.0.0D0).OR.(B.LT.0.0D0).OR.(T.LT.0.0D0)) THEN
        IERR=1
        PANS=0.0D0
        RETURN
      ENDIF
      IF((KODE.EQ.1).AND.(A.EQ.0.0D0))THEN
        IERR=1
        PANS=0.0D0
        RETURN
      ENDIF
      IF((KODE.EQ.1).AND.(B.EQ.0.0D0)) THEN
        PANS=0.0D0
        RETURN
      ENDIF
      IF((KODE.EQ.2).AND.(A.EQ.0.0D0)) THEN
        IF(B.EQ.0.0D0) THEN
          IERR=1
          PANS=0.0D0
          RETURN
        ELSE
          X=B*T
          KKODE=1
          TOL=0.5D-14
          PANS=DGERFC(X,KKODE,TOL,IERR)
          RETURN
        ENDIF
      ENDIF
      A2=A*A
      B2=B*B
      D=DSQRT(A2+B2)
      IF(T.EQ.0.0D0) THEN
        IF(KODE.EQ.2) THEN
          PANS=0.0D0
          IERR=1
          RETURN
        ENDIF
        RX=B/A
        IF(RX.LE.0.1D0) THEN
C         SMALL RATIO B/A COMPUTATION FOR LN[(B+D)/A]
          X2=RX*RX
          ARG=RX+X2/(1.0D0+DSQRT(1.0D0+X2))
          TRMLN=DLNOPX(ARG)
```

```
        ELSE
          TRMLN=DLOG((B+D)/A)
        ENDIF
        PANS=TRMLN
        RETURN
      ENDIF
      AT=A*T
      ARGA=AT*AT
      IF(ARGA.GT.670.0D0) THEN
        IERR=2
        PANS=0.0D0
        RETURN
      ENDIF
      N=1
      KKODE=1
      M=1
      TOL=REL
      CALL DEXINT(ARGA, N, KKODE, M, TOL, EN, NZ, MERR)
      IF(NZ.NE.0) THEN
        IERR=2
        PANS=0.0D0
        RETURN
      ENDIF
      EN1=EN(1)
      IF((KODE.EQ.2).AND.(B.EQ.0.0D0)) THEN
        PANS=0.5D0*EN1
        RETURN
      ENDIF
      BT=B*T
      ARGB=BT*BT
      XT=MAX(AT,BT)
      IF (XT.GT.6.0D0) THEN
        IF(XT.EQ.AT) THEN
          IF(KODE.EQ.1) THEN
            IF(BT.LT.6.0D0) GOTO 25
C           LARGE AT EXPANSION FOR P
C           ERF(BT)=1.0 +O[10**(-16)], FOR BT.GE.6.0, MAKING THE
C           RELATIVE ERROR O[10**(-16)]
            ERFBT=DRERF(BT,KODE,NZ)
            PANS=0.5D0*EN1*ERFBT
            RETURN
          ENDIF
          IF(XT.LT.7.0D0) GOTO 25
C         LARGE AT EXPANSION FOR PC: WANT XT AT LEAST 7.0 TO TERMINATE
C         ASYMPTOTIC EXPANSION
          ERFBT=DRERF(BT,KODE,NZ)
          IF(NZ.NE.0) THEN
            IERR=2
            PANS=0.0D0
            RETURN
          ENDIF
          COE=0.5D0
          AKSTRT=1.0D0
          SUM=EN1*ERFBT
          COES=-(B/A)/(XT*RTPI)
        ELSE
          IF(KODE.EQ.1)THEN
C           LARGE BT EXPANSION FOR P
C           ERF(BT)=1.0 +O[10**(-16)], FOR BT.GE.6.0, MAKING THE
C           RELATIVE ERROR O[10**(-16)]
            PANS=0.5D0*EN1
           RETURN
          ENDIF
          IF(XT.LT.7.0D0) GOTO 25
C         LARGE BT EXPANSION FOR PC; WANT XT AT LEAST 7.0 TO TERMINATE
C         THE ASYMPTOTIC EXPANSION
          COE=0.5D0/(XT*RTPI)
```

```
          AKSTRT=0.5D0
          SUM=0.0D0
          COES=1.0D0
        ENDIF
        ARGAB=ARGA+ARGB
        IF(ARGAB.GT.670.0D0) THEN
          PANS=0.0D0
          IERR=2
          RETURN
        ENDIF
        NZ=0
        FNH=1.5D0
        KKODE=1
        MM=22
        TOL=REL
        CALL DHEXINT(ARGAB, FNH, KKODE, MM, TOL, EN, NZ, IERR)
        IF(NZ.NE.0) THEN
          IERR=2
          QANS=0.0D0
          RETURN
        ENDIF
        S=EN(1)
        TRM1=S
        DTRM1=DABS(TRM1)
        AK=AKSTRT
        X2=XT*XT
        CK=-AK/X2
        DO 20 K=1,MM-1
          TRM2=CK*EN(K+1)
          DTRM2=DABS(TRM2)
          IF (DTRM2.LE.REL*DABS(S)) GOTO 22
          IF (DTRM2.GE.DTRM1) GOTO 21
          S=S+TRM2
          DTRM1=DTRM2
          AK=AK+1.0D0
          CK=-CK*AK/X2
  20    CONTINUE
C        WRITE(7,301)
C  301   FORMAT(" DROP THROUGH THE LOOP IN THE ASYMPTOTIC SERIES" )
  21    CONTINUE
C        WRITE(7,302)
C  302   FORMAT('REQUESTED ERROR NOT MET IN ASYMPTOTIC SERIES')
        IERR=3
  22    CONTINUE
        PANS=COE*(SUM+COES*S)
        RETURN
      ENDIF
 25   CONTINUE
      ARGAB=ARGA+ARGB
      RTX=T*D
      KKODE=2
      ERFCX=DRERF(RTX,KKODE,NZ)
      IF(NZ.NE.0) THEN
        IERR=2
        PANS=0.0D0
        RETURN
      ENDIF
      IF (A.LE.B) THEN
        RX=A/B
        X2=RX*RX
        IF(X2.LE.0.1D0) THEN
C         COMPUTE LN[2*D/(B+D)] FOR SMALL RATIOS (A/B)**2 TO RETAIN
C         SIGNIFICANT DIGITS
          TRM=0.5D0*DLNOPX(X2)
          ARG=0.5D0*X2/(1.0D0+DSQRT(1.0D0+X2))
          TRMLN=TRM-DLNOPX(ARG)
        ELSE
```

```
            ARGLN=(D+D)/(B+D)
            TRMLN=DLOG(ARGLN)
          ENDIF
          SUM=-TRMLN*ERFCX
          FNH=0.5D0
          KKODE=1
          MM=50
          TOL=REL
          CALL DHEXINT(ARGAB, FNH, KKODE, MM, TOL, ENH, NZ, MERR)
          IF(NZ.NE.0) THEN
            IERR=2
            PANS=0.0D0
            RETURN
          ENDIF
          KKODE=1
          TRM=DGERFC(RTX,KKODE,REL,MERR)
          SUM=SUM-TRM
          COE=0.5D0*RTX/RTPI
          CALL S1SER(A,B,ENH,MM,REL,S1,MERR)
          SUM=SUM+COE*S1
          IF(KODE.EQ.1) THEN
            PANS=0.5D0*EN1+SUM
          ELSE
            PANS=-SUM
          ENDIF
          RETURN
        ELSE
          RX=B/A
          IF(RX.LE.0.1D0) THEN
C           SMALL RATIO B/A COMPUTATION FOR LN[(B+D)/A]
            X2=RX*RX
            ARG=RX+X2/(1.0D0+DSQRT(1.0D0+X2))
            TRMLN=DLNOPX(ARG)
          ELSE
            TRMLN=DLOG((B+D)/A)
          ENDIF
          SUM=TRMLN*ERFCX
          N=1
          KKODE=1
          MM=50
          TOL=REL
          CALL DEXINT(ARGAB, N, KKODE, MM, TOL, EN, NZ, MERR)
          IF(NZ.NE.0) THEN
            IERR=2
            PANS=0.0D0
            RETURN
          ENDIF
          COE=B*T/RTPI
          CALL S2SER(B,A,EN,MM,REL,S2,IERR)
          SUM=SUM-COE*S2
          ERFBT=DRERF(BT,KODE,NZ)
          IF(KODE.EQ.1) THEN
            PANS=SUM+0.5D0*EN1*ERFBT
          ELSE
            PANS=-SUM+0.5D0*EN1*ERFBT
          ENDIF
        ENDIF
        RETURN
        END
        SUBROUTINE INTEGQ(A,B,T,REL,QANS,IERR)
C
C     DONALD E. AMOS    DECEMBER, 2001;  JANUARY, 2006
C-----------------------------------------------------------------------
C     A DOUBLE PRECISION ROUTINE
C-----------------------------------------------------------------------
C
C     REFERENCES: FOLDER 11, FOLDER 6
```

```
C
C     INTEGQ COMPUTES THE INTEGRAL
C
C       QANS =  INT ON (T,INF) OF EXP(-(A*W)**2)*E(1,(B*W)**2)
C
C     BY SERIES FOR A.GE.0, B.GT.0, T.GE.0. REL=0.5D-14 FOR HIGHEST
C     POSSIBLE ACCURACY.
C
C     IERR=0 IS A NORMAL RETURN
C     IERR=1 IS AN INPUT ERROR, SOME PARAMETER IS OUT OF ITS RANGE
C            QANS=0.0D0 RETURNED
C     IERR=2 UNDERFLOW, QANS=0.0D0 RETURNED
C     IERR=3 REQUESTED ERROR NOT MET IN ASYMPTOTIC EXPANSION
C-----------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION EN(100),ENH(100)
      DATA RTPI/ 1.772453850905516D0/
      IERR=0
      IF((A.LT.0.0D0).OR.(B.LE.0.0D0).OR.(T.LT.0.0D0)) THEN
        IERR=1
        QANS=0.0D0
        RETURN
      ENDIF
      IF (A.EQ.0.0D0) THEN
        BT=B*T
        ARGB=BT*BT
        IF(ARGB.GT.670.0D0)THEN
          IERR=2
          QANS=0.0D0
          RETURN
        ENDIF
        KKODE=2
        ERFB=DRERF(BT,KKODE,NZ)
        IF(NZ.NE.0) THEN
          IERR=2
          QANS=0.0D0
          RETURN
        ENDIF
        TRM=ERFB*RTPI/B
        IF(T.EQ.0.0D0) THEN
          QANS=TRM
          RETURN
        ENDIF
        N=1
        KKODE=1
        M=1
        TOL=REL
        CALL DEXINT(ARGB, N, KKODE, M, TOL, EN, NZ, IERR)
        IF(NZ.NE.0) THEN
          IERR=2
          QANS=0.0D0
          RETURN
        ENDIF
        QANS=TRM-T*EN(1)
        RETURN
      ENDIF
      A2=A*A
      B2=B*B
      D=DSQRT(A2+B2)
      IF(T.EQ.0.0D0) THEN
        RX=A/B
        IF(RX.LE.0.1D0) THEN
C         SMALL RATIO A/B COMPUTATION FOR LN[(A+D)/B]
          X2=RX*RX
          ARG=RX+X2/(1.0D0+DSQRT(1.0D0+X2))
          TRMLN=DLNOPX(ARG)
        ELSE
```

```
          TRMLN=DLOG((A+D)/B)
        ENDIF
        QANS=RTPI*TRMLN/A
        RETURN
      ENDIF
      AT=A*T
      BT=B*T
      ARGA=AT*AT
      ARGB=BT*BT
      ARGAB=ARGA+ARGB
      IF(ARGAB.GT.670.0D0) THEN
        IERR=2
        QANS=0.0D0
        RETURN
      ENDIF
      XT=MAX(AT,BT)
      IF (XT.GE.7.0D0) THEN
C-----------------------------------------------------------------------
C       LARGE AT EXPANSION:
C
C       Q(A,B,T)=RTPI*(P(B,A,T)-0.5*E(1,(BT)**2)*ERF(AT))/A USING THE
C       ASYMPTOTIC EXPANSION FOR P(B,A,T) FOR AT.GE.7.0D0
C
C       LARGE BT EXPANSION:
C       INTEGRATION OF Q USING THE ASYMPTOTIC EXPANSION OF E(1,(BT)**2)
C
C       BOTH HAVE SIMILAR SUMS WITH ONLY MINOR CHANGES IN START VALUES
C       AND COEFFICIENTS
C-----------------------------------------------------------------------
        IF(XT.EQ.AT) THEN
C         LARGE AT EXPANSION
          COE=0.5D0*RTPI/A
          AKSTRT=0.5D0
          COES=-1.0D0/(AT*RTPI)
          KKODE=2
          ERFAT=DRERF(AT,KKODE,NZ)
          IF(NZ.NE.0) THEN
            IERR=2
            QANS=0.0D0
            RETURN
          ENDIF
          NZ=0
          N=1
          KKODE=1
          MM=1
          TOL=REL
          CALL DEXINT(ARGB, N, KKODE, MM, TOL, EN, NZ, IERR)
          IF(NZ.NE.0) THEN
            IERR=2
            QANS=0.0D0
            RETURN
          ENDIF
          SUM=EN(1)*ERFAT
        ELSE
C         LARGE BT EXPANSION
          COE=0.5D0/(BT*B)
          AKSTRT=1.0D0
          SUM=0.0D0
          COES=1.0D0
        ENDIF
        NZ=0
        FNH=1.5D0
        KKODE=1
        MM=22
        TOL=REL
        CALL DHEXINT(ARGAB, FNH, KKODE, MM, TOL, EN, NZ, IERR)
        IF(NZ.NE.0) THEN
```

```
            IERR=2
            QANS=0.0D0
            RETURN
          ENDIF
          S=EN(1)
          TRM1=S
          DTRM1=DABS(TRM1)
          AK=AKSTRT
          X2=XT*XT
          CK=-AK/X2
          DO 20 K=1,MM-1
            TRM2=CK*EN(K+1)
            DTRM2=DABS(TRM2)
            IF (DTRM2.LE.REL*DABS(S)) GOTO 22
            IF (DTRM2.GE.DTRM1) GOTO 21
            S=S+TRM2
            DTRM1=DTRM2
            AK=AK+1.0D0
            CK=-CK*AK/X2
   20     CONTINUE
C          WRITE(7,301)
C  301     FORMAT(" DROP THROUGH THE LOOP IN THE ASYMPTOTIC SERIES" )
   21     CONTINUE
C          WRITE(7,302)
C  302     FORMAT('REQUESTED ERROR NOT MET IN ASYMPTOTIC SERIES')
          IERR=3
   22     CONTINUE
          QANS=COE*(SUM+COES*S)
          RETURN
        ENDIF
C       START OF CONVERGENT SERIES FOR A.LE.B AND A.GT.B
        RTX=T*D
        KKODE=2
        ERFCX=DRERF(RTX,KKODE,NZ)
        IF(NZ.NE.0) THEN
          IERR=2
          QANS=0.0D0
          RETURN
        ENDIF
        IF (A.LE.B) THEN
          RX=A/B
          IF(RX.LE.0.1D0) THEN
C         SMALL RATIO A/B COMPUTATION FOR LN[(A+D)/B]
            X2=RX*RX
            ARG=RX+X2/(1.0D0+DSQRT(1.0D0+X2))
            TRMLN=DLNOPX(ARG)
          ELSE
            TRMLN=DLOG((A+D)/B)
          ENDIF
          SUM=TRMLN*ERFCX*RTPI/A
          N=1
          KKODE=1
          MM=50
          TOL=REL
          CALL DEXINT(ARGAB, N, KKODE, MM, TOL, EN, NZ, IERR)
          IF(NZ.NE.0) THEN
            IERR=2
            QANS=0.0D0
            RETURN
          ENDIF
          CALL S2SER(A,B,EN,MM,REL,S2,IERR)
          QANS=SUM-T*S2
        ELSE
          N=1
          KKODE=1
          M=1
          TOL=REL
```

```
        CALL DEXINT(ARGB, N, KKODE, M, TOL, EN, NZ, IERR)
        IF(NZ.NE.0) THEN
          IERR=2
          QANS=0.0D0
          RETURN
        ENDIF
        SUM=0.5D0*EN(1)
        KKODE=2
        ERFCT=DRERF(AT,KKODE,NZ)
        IF(NZ.NE.0) THEN
          IERR=2
          QANS=0.0D0
          RETURN
        ENDIF
        SUM=SUM*ERFCT
        RX=B/A
        X2=RX*RX
        IF(X2.LE.0.1D0) THEN
C         COMPUTE LN[2*D/(A+D)] FOR SMALL RATIOS (B/A)**2 TO RETAIN
C         SIGNIFICANT DIGITS
          TRM=0.5D0*DLNOPX(X2)
          ARG=0.5D0*X2/(1.0D0+DSQRT(1.0D0+X2))
          TRMLN=TRM-DLNOPX(ARG)
        ELSE
          ARGLN=(D+D)/(A+D)
          TRMLN=DLOG(ARGLN)
        ENDIF
        SUM=SUM-TRMLN*ERFCX
        FNH=0.5D0
        KKODE=1
        MM=50
        TOL=REL
        CALL DHEXINT(ARGAB, FNH, KKODE, MM, TOL, ENH, NZ, IERR)
        IF(NZ.NE.0) THEN
          IERR=2
          QANS=0.0D0
          RETURN
        ENDIF
        KKODE=1
        TRM=DGERFC(RTX,KKODE,REL,IERR)
        SUM=(SUM-TRM)*RTPI
        COE=0.5D0*RTX
        CALL S1SER(B,A,ENH,MM,REL,S1,IERR)
        QANS=(SUM+COE*S1)/A
      ENDIF
      RETURN
      END
      SUBROUTINE S1SER(A,B,ENHX,MM,REL,S1,IERR)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION ENHX(MM)
      A2=A*A
      B2=B*B
      D2=A2+B2
      RAB=A2/D2
      SNH=0.0D0
      AKH=0.5D0
      BKH=1.0D0
      CKH=0.5D0*RAB
      DO 10 K=1,MM-1
        TRM=CKH*ENHX(K+1)/BKH
        SNH=SNH+TRM
        IF(DABS(TRM).LE.REL*DABS(SNH)) GOTO 11
        BKH=BKH+1.0D0
        AKH=AKH+1.0D0
        CKH=CKH*(AKH/BKH)*RAB
   10 CONTINUE
      IERR=5
```

```
C       WRITE(7,200)
C 200  FORMAT('NO CONVERGENCE IN S1SER IN MM TERMS')
  11  CONTINUE
      S1=SNH
      RETURN
      END
      SUBROUTINE S2SER(A,B,ENX,MM,REL,S2,IERR)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION ENX(MM)
      A2=A*A
      B2=B*B
      D2=A2+B2
      RAB=A2/D2
      SN=ENX(1)
      AK=3.0D0
      CK=RAB
      DO 10 K=1,MM-1
        TRM=CK*ENX(K+1)/AK
        SN=SN+TRM
        IF(DABS(TRM).LE.REL*DABS(SN)) GOTO 11
        CK=CK*RAB
        AK=AK+2.0D0
  10  CONTINUE
      IERR=5
C       WRITE(7,200)
C 200  FORMAT('NO CONVERGENCE IN S2SER IN MM TERMS')
  11  CONTINUE
      S2=SN
      RETURN
      END
      DOUBLE PRECISION FUNCTION DLNOPX(X)
C
C     DLNOPX COMPUTES THE LN(1+X) FOR SMALL X.LE.0.1D0 TO RETAIN
C     SIGNIFICANT DIGITS WHEN X IS KNOWN AND SMALL
C-----------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      REL=0.5D-16
      S=0.0D0
      AK=1.0D0
      PX=1.0D0
      DO 10 K=1,50
        TRM=PX/AK
        S=S+TRM
        IF (DABS(TRM).LT.REL*DABS(S)) GOTO 11
        AK=AK+1.0D0
        PX=-PX*X
 10   CONTINUE
 11   CONTINUE
      DLNOPX=X*S
      RETURN
      END
      SUBROUTINE INTEGW3(A,B,T,KODE,REL,ANSW3,IERR)
C
C     DONALD E. AMOS, JANUARY, 2003
C-----------------------------------------------------------------------
C     A DOUBLE PRECISION ROUTINE
C-----------------------------------------------------------------------
C
C     INTEGW3 COMPUTES THE W3 INTEGRAL OF FOLDER 10 USING THE CLOSED
C     FORM DERIVED IN FOLDER 10A ON KODE=1:
C
C         ANSW3=INT ON (T,INF) OF ERF(A*W)*ERF(B*W)/(W**3)
C
C     ON KODE=2, THE COMPLEMENTARY FUNCTION W3C IS COMPUTED,
C
C         ANSW3=INT ON (T,INF) OF ERFC(A*W)*ERFC(B*W)/(W**3)
C
```

```
C     WHERE A.GE.0.0D0, B.GE.0.0D0, T.GT.0.0D0.
C
C     IERR=0 IS A  NORMAL RETURN
C     IERR=1 MEANS T.LE.0.0D0, ANSI1=0.0D0 RETURNED FROM INTEGI1
C     IERR=2 MEANS ONE OR BOTH OF THE PARAMETERS A OR B IS NEGATIVE
C            OR KODE IS NOT 1 OR 2 ANSW3=0.0D0 RETURND
C     IERR=4 IS AN UNDERFLOW AND ANSI1=0.0D0 IS RETURNED FROM INTEGI1
C-----------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DATA RTPI       /1.772453850905516D0/
      IF((KODE.NE.1).AND.(KODE.NE.2)) THEN
        IERR=2
        ANSW3=0.0D0
        RETURN
      ENDIF
      IF(KODE.EQ.1) THEN
        COEF=1.0D0
      ELSE
        COEF=-1.0D0
      ENDIF
      ARG=A*T
      ERFAT=DRERF(ARG,KODE,NZ)
      ARG=B*T
      ERFBT=DRERF(ARG,KODE,NZ)
      CALL INTEGI1(A,B,T,KODE,REL,ANSAB,IERR)
      CALL INTEGI1(B,A,T,KODE,REL,ANSBA,IERR)
      ANSW3=0.5D0*(ERFAT/T)*(ERFBT/T)+COEF*(A*ANSAB+B*ANSBA)/RTPI
      RETURN
      END
      SUBROUTINE INTEGI21(A,B,C,T,KODE,ANSI21,I21ERR,KFORM)
C
C     DONALD E. AMOS, OCTOBER, 2002
C-----------------------------------------------------------------------
C     A DOUBLE PRECISION ROUTINE
C-----------------------------------------------------------------------
C
C     REF: FOLDER 21
C
C     ON KODE=1, INTEGI21 COMPUTES
C
C        ANSI21=INT ON (0,T) OF U(A,B,W)*ERF(C/RW);
C
C     ON KODE=2, INTEGI21 COMPUTES THE COMPLEMENTARY FUNCTION
C
C        ANSI21=INT ON (0,T) OF U(A,B,W)*ERFC(C/RW)
C
C     WHERE U(A,B,W)=EXP(A*A*W+2*A*B)*ERFC(A*RW+B/RW), RW=SQRT(W)
C     AND  A , B AND C ARE NON-NEGATIVE.
C
C     ERROR FLAG:
C          I21ERR= 0 NORMAL RETURN
C                =-1 CONVERGENCE NOT SEEN IS SOME SERIES
C                = 4 INPUT VARIABLE OUT OF RANGE
C                = 5 EXPONENTIAL UNDERFLOW, (B^2+C^2)/T TOO LARGE
C                =   ANY OTHER VALUE IS AN ERROR FROM DGAUS8
C
C     LEGENDS FOR KFORM:
C
C     KFORM DIGIT=0 AND I21ERR=0 MEANS A SPECIAL FORMULA WAS USED
C     KFORM DIGIT=1 MEANS T.LE.(C*C/36) WHERE ERF(C/DSQRT(T))=1.0D0
C     KFORM DIGIT=2 MEANS THE CLOSED FORM IS USED FOR DVOFT
C     KFORM DIGIT=3 MEANS THE LARGE L.GE.2 SERIES IS USED FOR W3
C     KFORM DIGIT=4 MEANS THE A*SQRT(T).LT.1 SERIES IS USED FOR I21
C     KFORM DIGIT=5 MEANS THE THE QUADRATURE IS USED FOR W3
C
C     EXAMPLE:
C     KFORM=12 MEANS A COMBINATION OF 1 AND 2 WAS USED
```

```
C-----------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION EN(1),YN(100)
      DATA RTPI /1.772453850905516D0/
      I21ERR=0
      KFORM=0
      IF(A.LT.0.0D0)THEN
         I21ERR=4
         ANSI21=0.0D0
         RETURN
      ENDIF
      IF(B.LT.0.0D0)THEN
         I21ERR=4
         ANSI21=0.0D0
         RETURN
      ENDIF
      IF(C.LT.0.0D0)THEN
         I21ERR=4
         ANSI21=0.0D0
         RETURN
      ENDIF
      IF(T.LT.0.0D0)THEN
         I21ERR=4
         ANSI21=0.0D0
         RETURN
      ENDIF
      IF((KODE.NE.1).AND.(KODE.NE.2))THEN
         I21ERR=4
         ANSI21=0.0D0
         RETURN
      ENDIF
      IF(T.EQ.0.0D0)THEN
         ANSI21=0.0D0
         RETURN
      ENDIF
      IF(KODE.EQ.1) THEN
        IF(C.EQ.0.0D0)THEN
          ANSI21=0.0D0
          RETURN
        ENDIF
      ENDIF
      REL=0.5D-14
      RT=DSQRT(T)
      CAPT=1.0D0/RT
      IF((A.EQ.0.0D0).AND.(B.EQ.0.0D0)) THEN
        X=C*CAPT
        NZ=0
        ERFC=DRERF(X,KODE,NZ)
        KKODE=1
        CALL DIERFC(X,KKODE,ANS,IERR)
        IF(KODE.EQ.1) THEN
          ANSI21=T*ERFC+(C+C)*RT*ANS
        ELSE
          IF((NZ.NE.0).OR.(IERR.EQ.2)) THEN
            ANSI21=0.0D0
            I21ERR=5
          ELSE
            ANSI21=T*ERFC-(C+C)*RT*ANS
          ENDIF
        ENDIF
        RETURN
      ENDIF
      CS=C*C
      BS=B*B
      CAPX=(BS+CS)/T
      IF(CAPX.GT.667.0D0) THEN
        ANSI21=0.0D0
```

```
          I21ERR=5
          RETURN
        ENDIF
        VT=DVOFT(A,B,T,REL,IVERR,IFORM)
        IF(KODE.EQ.1) THEN
          TLIM=CS/36.0D0
          IF(T.LE.TLIM) THEN
C           ERF(C/RT)=1 + O(10**(-16)) HERE
            ANSI21=VT
            KFORM=10+IFORM
            RETURN
          ENDIF
        ELSE
          IF(C.EQ.0.0D0) THEN
            ANSI21=VT
            KFORM=IFORM
            RETURN
          ENDIF
        ENDIF
        IF(IFORM.EQ.4) THEN
C         SERIES FOR A*RT.LE.1 COMPUTED IN DVOFT WHEN IFORM=4
C         RESOLUTION OF INDETERMINANT FORM FOR SMALL A AND SERIES FOR
C         A*RT.LE.1
          KFORM=4
          CRT=A*RT
          CT=CRT+CRT
          IF(A.EQ.0.0D0) THEN
            M=4
          ELSE
            M=40
          ENDIF
          TOL=REL
          CALL GNSEQ(C,B,CAPT,M,TOL,YN)
          S=0.0D0
          PW=1.0D0
          DO 20 K=3,M
            TRM=PW*YN(K)
            S=S+TRM
            IF(DABS(TRM).LE.DABS(S)*REL) GOTO 65
            PW=-PW*CT
 20       CONTINUE
          I21ERR=-1
C          PRINT*,'END OF G LOOP'
 65       CONTINUE
          TERM2=4.0D0*RT*C*S/RTPI
          ARG=C/RT
          ERFA=DRERF(ARG,KODE,NZ)
          IF(KODE.EQ.1) THEN
            ANSI21=VT*ERFA+TERM2*DEXP(-CAPX)
          ELSE
            ANSI21=VT*ERFA-TERM2*DEXP(-CAPX)
          ENDIF
          RETURN
        ENDIF
        ARG=C/RT
        NZ=0
        ERFA=DRERF(ARG,KODE,NZ)
        TRM0=VT*ERFA
        COE=2.0D0/(A*RTPI)
        N=1
        KKODE=1
        M=1
        TOL=REL
        NZ=0
        IERR=0
        CALL DEXINT(CAPX, N, KKODE, M, TOL, EN, NZ, IERR)
        CW1=COE*EN(1)
```

```
      CALL INTEGS1(A,B,C,T,TOL,CW3,IERR,JFORM)
      AS=A*A
      COE=(RT+RT)/(AS*RTPI)
      CW3=CW3*COE
      KFORM=10*IFORM+JFORM
      IF (IERR.EQ.1) THEN
        I21ERR=0
      ELSE
        I21ERR=IERR
      ENDIF
      CALL INTEGI5(C,B,CAPT,ANSI5,IERR)
      CW2=2.0D0*((B+B)/A+1.0D0/AS)*ANSI5
      IF(KODE.EQ.1) THEN
        ANSI21=TRM0+C*(CW1-CW2+CW3)/RTPI
      ELSE
        ANSI21=TRM0-C*(CW1-CW2+CW3)/RTPI
      ENDIF
      RETURN
      END
      SUBROUTINE INTEGJ21(A,B,C,T,ANSJ21,J21ERR,KFORM)
C
C     DONALD E. AMOS, OCTOBER, 2002
C-----------------------------------------------------------------------
C     A DOUBLE PRECISION ROUTINE
C-----------------------------------------------------------------------
C
C     REF: FOLDER 21
C
C     INTEGJ21 COMPUTES
C
C     ANSJ21= INT ON (0,T) OF U(A,B,W)*DEXP(-C^2/W)/(W*RW)
C
C     WHERE U(A,B,W)=EXP(A*A*W+2*A*B)*ERFC(A*RW+B/RW), RW=SQRT(W)
C
C     AND C IS POSITIVE, A NON-NEGATIVE, B NON-NEGATIVE   OR
C         C IS ZERO,     A NON-NEGATIVE, B POSITIVE
C
C     ERROR FLAG:
C          J21ERR= 0 NORMAL RETURN
C                =-1 CONVERGENCE NOT SEEN IS SOME SERIES
C                = 4 INPUT VARIABLE OUT OF RANGE
C                = 5 EXPONENTIAL UNDERFLOW, (B^2+C^2)/T TOO LARGE
C                =   ANY OTHER VALUE IS AN ERROR FROM DGAUS8 OR INTEGI5
C
C     LEGENDS FOR KFORM:
C     KFORM DIGIT=0 AND J21ERR=0 MEANS A SPECIAL CASE FORMULA IS USED
C     KFORM DIGIT=3 MEANS THE LARGE L.GE.2 SERIES IS USED FOR S1
C     KFORM DIGIT=4 MEANS THE SERIES FOR SMALL A*DSQRT(T).LE.1 WAS USED
C     KFORM DIGIT=5 MEANS THE THE QUADRATURE IS USED FOR S1
C-----------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION YN(60)
      DATA RTPI /1.772453850905516D0/
      J21ERR=0
      KFORM=0
      IF(A.LT.0.0D0)THEN
         J21ERR=4
         ANSJ21=0.0D0
         RETURN
      ENDIF
      IF(B.LT.0.0D0)THEN
         J21ERR=4
         ANSJ21=0.0D0
         RETURN
      ENDIF
      IF(C.LT.0.0D0)THEN
         J21ERR=4
```

```
            ANSJ21=0.0D0
            RETURN
         ENDIF
         IF(T.LT.0.0D0)THEN
            J21ERR=4
            ANSJ21=0.0D0
            RETURN
         ENDIF
C        SPECIAL FORMULAE FOR ONE OR MORE PARAMETERS EQUAL TO ZERO
C        IF THE INTEGRAL EXISTS, THEN THE CASE T=0 IS TESTED
         IF(C.EQ.0.0D0) THEN
           IF(B.EQ.0.0D0) THEN
             J21ERR=4
             ANSJ21=0.0D0
             RETURN
           ELSE
             IF(A.EQ.0.0D0) THEN
               IF(T.EQ.0.0D0) THEN
                 ANSJ21=0.0D0
                 RETURN
               ENDIF
               X=B/DSQRT(T)
               KODE=1
               CALL DIERFC(X,KODE,ANS,IERR)
               IF(IERR.EQ.2) THEN
                 J21ERR=5
                 ANSJ21=0.0D0
                 RETURN
               ENDIF
               ANSJ21=2.0D0*ANS/B
               RETURN
             ENDIF
           ENDIF
         ELSE
           IF((A.EQ.0.0D0).AND.(B.EQ.0.0D0)) THEN
             IF(T.EQ.0.0D0) THEN
               ANSJ21=0.0D0
               RETURN
             ENDIF
             ARG=C/DSQRT(T)
             KODE=2
             NZ=0
             ERFC=DRERF(ARG,KODE,NZ)
             IF(NZ.NE.0) THEN
                ANSJ21=0.0D0
                J21ERR=5
                RETURN
             ENDIF
             ANSJ21=RTPI*ERFC/C
             RETURN
           ENDIF
           IF(A.EQ.0.0D0) THEN
             IF(T.EQ.0.0D0) THEN
               ANSJ21=0.0D0
               RETURN
             ENDIF
             CAPT=1.0D0/DSQRT(T)
             CALL INTEGI5(C,B,CAPT,ANSI5,IERR)
             IF(IERR.EQ.4) THEN
               ANSJ21=0.0D0
               J21ERR=5
               RETURN
             ENDIF
             ANSJ21=ANSI5+ANSI5
             RETURN
           ENDIF
         ENDIF
```

```
      IF(T.EQ.0.0D0)THEN
         ANSJ21=0.0D0
         RETURN
      ENDIF
      REL=0.5D-14
      RT=DSQRT(T)
      ART=A*RT
      IF(ART.LE.1.0D0) THEN
        KFORM=4
        CS=C*C
        BS=B*B
        CAPX=(CS+BS)/T
        IF(CAPX.GT.667.0D0) THEN
          ANSJ21=0.0D0
          J21ERR=5
          RETURN
        ENDIF
        AT=ART+ART
        CAPT=1.0D0/RT
        M=40
        TOL=REL
        CALL GNSEQ(C,B,CAPT,M,TOL,YN)
        S=YN(1)
        PW=-AT
        DO 10 K=1,M-1
          TRM=PW*YN(K+1)
          S=S+TRM
          IF(DABS(TRM).LT.DABS(S)*TOL) GOTO 20
          PW=-PW*AT
 10     CONTINUE
 20     CONTINUE
        ANSJ21=S*DEXP(-CAPX)/RT
        RETURN
      ENDIF
      BS=B*B
      CS=C*C
      CAPX=(CS+BS)/T
      IF(CAPX.GT.667.0D0) THEN
        ANSJ21=0.0D0
        J21ERR=5
        RETURN
      ENDIF
      TOL=REL
      CALL INTEGS1(A,B,C,T,TOL,CS1,IERR,KFORM)
      IF (IERR.EQ.1) THEN
        J21ERR=0
      ELSE
        J21ERR=IERR
      ENDIF
      ANSJ21=2.0D0*RT*CS1/RTPI
      RETURN
      END
      SUBROUTINE INTEGI22(A,B,C,T,ANSI22,I22ERR,KFORM)
C
C     DONALD E. AMOS, OCTOBER, 2002
C-----------------------------------------------------------------------
C     A DOUBLE PRECISION ROUTINE
C-----------------------------------------------------------------------
C     REF: FOLDER 22
C
C     INTEGI22 COMPUTES
C
C     ANSI22=INT ON (0,T) OF U(A,B,W)*EXP(-C*C/W)/RW
C
C     WHERE U(A,B,W)=EXP(A*A*W+2*A*B)*ERFC(A*RW+B/RW), RW=SQRT(W)
C
C     AND  A , B  AND C ARE NON-NEGATIVE.
```

```
C
C     ERROR FLAG:
C          I22ERR= 0 NORMAL RETURN
C                =-1 CONVERGENCE NOT SEEN IS SOME SERIES
C                = 4 INPUT VARIABLE OUT OF RANGE
C                = 5 EXPONENTIAL UNDERFLOW, (B*B+C*C)/T TOO LARGE
C                =   ANY OTHER VALUE IS AN ERROR FROM DGAUS8
C
C     LEGENDS FOR KFORM:
C     KFORM DIGIT=0 MEANS A SPECIAL FORMULA WAS USED
C     KFORM DIGIT=1 MEANS A=0
C     KFORM DIGIT=2 MEANS (A=0 AND B=0) OR (B=0 AND C=0)
C     KFORM DIGIT=3 MEANS THE LARGE L.GE.2 SERIES IS USED FOR S2
C     KFORM DIGIT=4 MEANS THE SERIES FOR A*DSQRT(T).LE.1 IS USED
C     KFORM DIGIT=5 MEANS THE THE QUADRATURE IS USED FOR S2
C
C-----------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION EN(1),YN(60)
      DATA RTPI /1.772453850905516D0/
      I22ERR=0
      KFORM=0
      IF(A.LT.0.0D0)THEN
         I22ERR=4
         ANSI22=0.0D0
         RETURN
      ENDIF
      IF(B.LT.0.0D0)THEN
         I22ERR=4
         ANSI22=0.0D0
         RETURN
      ENDIF
      IF(C.LT.0.0D0)THEN
         I22ERR=4
         ANSI22=0.0D0
         RETURN
      ENDIF
      IF(T.LT.0.0D0)THEN
         I22ERR=4
         ANSI22=0.0D0
         RETURN
      ENDIF
      IF(T.EQ.0.0D0)THEN
         ANSI22=0.0D0
         RETURN
      ENDIF
      RT=DSQRT(T)
      IF((A.EQ.0.0D0).AND.(B.EQ.0.0D0).AND.(C.EQ.0.0D0)) THEN
        ANSI22=RT+RT
        RETURN
      ENDIF
      IF((A.EQ.0.0D0).AND.(B.EQ.0.0D0)) THEN
        X=C/RT
        KODE=1
        CALL DIERFC(X,KODE,DI1ERFC,IERR)
        IF(IERR.EQ.2) THEN
          I22ERR=5
          ANSI22=0.0D0
          RETURN
        ENDIF
        ANSI22=2.0D0*RTPI*RT*DI1ERFC
        KFORM=2
        RETURN
      ENDIF
      REL=0.5D-14
      IF((B.EQ.0.0D0).AND.(C.EQ.0.0D0)) THEN
        KFORM=2
```

```
          X=A*RT
          ANSI22=2.0D0*DHERFC(X)/A
          RETURN
        ENDIF
        ART=A*RT
        IF(ART.LE.1.0D0) THEN
C         SMALL A*DSQRT(T).LE.1.0D0 SERIES FOR INTEGI22
          KFORM=4
          CAPT=1.0D0/RT
          ARG=(B*B+C*C)/T
          IF(ARG.GT.667.0D0) THEN
            ANSI22=0.0D0
            I22ERR=5
            RETURN
          ENDIF
          IF(A.EQ.0.0D0) THEN
            M=4
          ELSE
            M=50
          ENDIF
          REL=0.5D-14
          CALL GNSEQ(C,B,CAPT,M,REL,YN)
          S1=YN(2)
          S2=2.0D0*YN(3)
          FK=ART+ART
          PK=-FK
          AK=3.0D0
          IFLAG1=0
          IFLAG2=0
          JFLAG=0
          TOL=REL/10.0D0
          DO 10 K=1,M-3
            IF(IFLAG1.EQ.0) THEN
              TRM1=PK*YN(K+2)
              S1=S1+TRM1
              IF(DABS(TRM1).LT.TOL*DABS(S1)) THEN
                IFLAG1=1
                JFLAG=JFLAG+1
              ENDIF
            ENDIF
            IF(IFLAG2.EQ.0) THEN
              TRM2=PK*YN(K+3)*AK
              S2=S2+TRM2
              IF(DABS(TRM2).LT.TOL*DABS(S2)) THEN
                IFLAG2=1
                JFLAG=JFLAG+1
              ENDIF
            ENDIF
            IF(JFLAG.EQ.2) GOTO 20
            PK=-PK*FK
            AK=AK+1.0D0
 10       CONTINUE
          I22ERR=-1
 20       CONTINUE
          COEF=2.0D0*DEXP(-ARG)
          ANSI22=COEF*(B*S1+RT*S2)
          RETURN
        ENDIF
        BS=B*B
        CS=C*C
        ARG=(BS+CS)/T
        N=1
        KODE=1
        M=1
        NZ=0
        TOL=REL
        CALL DEXINT(ARG, N, KODE, M, TOL, EN, NZ, IERR)
```

```
      IF(NZ.NE.0) THEN
        ANSI22=0.0D0
        I22ERR=5
        KFORM=35
        RETURN
      ENDIF
      TOL=REL
      CALL INTEGS2(A,B,C,T,TOL,S2,IERR,KFORM)
      I22ERR=IERR
      ANSI22=(EN(1)-(RT+RT)*S2)/(A*RTPI)
      RETURN
      END
      SUBROUTINE INTEGJ22(A,B,C,T,ANSJ22,J22ERR,KFORM)
C
C     DONALD E. AMOS, OCTOBER, 2002
C-----------------------------------------------------------------------
C     A DOUBLE PRECISION ROUTINE
C-----------------------------------------------------------------------
C     REF: FOLDER 22
C
C     INTEGJ22 COMPUTES
C
C     ANSJ22=INT ON (0,T) OF U(A,B,W)*EXP(-C*C/W)*RW
C
C     WHERE U(A,B,W)=EXP(A*A*W+2*A*B)*ERFC(A*RW+B/RW), RW=SQRT(W)
C
C     AND  A , B  and C ARE NON-NEGATIVE.
C
C     ERROR FLAG:
C          J22ERR= 0 NORMAL RETURN
C                =-1 CONVERGENCE NOT SEEN IS SOME SERIES
C                = 4 INPUT VARIABLE OUT OF RANGE
C                = 5 EXPONENTIAL UNDERFLOW, (B^2+C^2)/T TOO LARGE
C                =   ANY OTHER VALUE IS AN ERROR FROM DGAUS8
C
C     LEGENDS FOR KFORM:
C
C     KFORM DIGIT=0 MEANS A SPECIAL FORMULA WAS USED
C     KFORM DIGIT=1 MEANS A=0
C     KFORM DIGIT=2 MEANS (A=0 & B=0) OR (A=0 & C=0) OR (B=0 & C=0)
C     KFORM DIGIT=3 MEANS THE LARGE L.GE.2 SERIES IS USED FOR S1 OR S2
C     KFORM DIGIT=4 MEANS THE SERIES FOR A*DSQRT(T).LE.1 IS USED
C     KFORM DIGIT=5 MEANS THE THE QUADRATURE IS USED FOR S1 OR S2
C-----------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION EN(2),YN(100)
      DATA RTPI /1.772453850905516D0/
      J22ERR=0
      KFORM=0
      IF(A.LT.0.0D0)THEN
         J22ERR=4
         ANSJ22=0.0D0
         RETURN
      ENDIF
      IF(B.LT.0.0D0)THEN
         J22ERR=4
         ANSJ22=0.0D0
         RETURN
      ENDIF
      IF(C.LT.0.0D0)THEN
         J22ERR=4
         ANSJ22=0.0D0
         RETURN
      ENDIF
      IF(T.LT.0.0D0)THEN
         J22ERR=4
         ANSJ22=0.0D0
```

```
            RETURN
         ENDIF
         IF(T.EQ.0.0D0)THEN
            ANSJ22=0.0D0
            RETURN
         ENDIF
         RT=DSQRT(T)
         CAPT=1.0D0/RT
         IF((A.EQ.0.0D0).AND.(B.EQ.0.0D0).AND.(C.EQ.0.0D0)) THEN
           ANSJ22=2.0D0*RT*T/3.0D0
           RETURN
         ENDIF
         AS=A*A
         BS=B*B
         CS=C*C
         REL=0.5D-14
         IF((A.EQ.0.0D0).AND.(B.EQ.0.0D0)) THEN
           KFORM=2
           X=CS/T
           FNH=2.5D0
           KODE=1
           M=1
           TOL=REL
           NZ=0
           CALL DHEXINT(X, FNH, KODE, M, TOL, EN, NZ, IERR)
           IF(NZ.NE.0) THEN
             J22ERR=5
             ANSJ22=0.0D0
             RETURN
           ENDIF
           ANSJ22=T*RT*EN(1)
           RETURN
         ENDIF
         IF((A.EQ.0.0D0).AND.(C.EQ.0.0D0)) THEN
           KFORM=2
           X2=B*CAPT
           KODE=2
           NZ=0
           ERFX=DRERF(X2,KODE,NZ)
           ARG=BS/T
           N=2
           KODE=1
           M=1
           NZ=0
           TOL=REL
           CALL DEXINT(ARG, N, KODE, M, TOL, EN, NZ, IERR)
           IF(NZ.NE.0) THEN
             ANSJ22=0.0D0
             J22ERR=5
             RETURN
           ENDIF
           ANSJ22=(T+T)*(RT*ERFX-B*EN(1)/RTPI)/3.0D0
           RETURN
         ENDIF
         IF((B.EQ.0.0D0).AND.(C.EQ.0.0D0)) THEN
C          CASE FOR J23 OF FOLDER 23
           KFORM=2
           X=A*RT
           TOL=REL
           IF (X.LE.1.0D0) THEN
C            PE=RECURRENCE ON EVEN TERMS; PO=RECURRENCE ON ODD TERMS
             PO=2.0D0*X/RTPI
             S=1.0D0/3.0D0-PO/4.0D0
             IF(DABS(PO).LE.TOL*DABS(S)) THEN
               ANSJ22=(S+S)*T*RT
               RETURN
             ENDIF
```

```
          PE=1.0D0
          AK=2.0D0
          BK=1.0D0
          XS=X*X
          DO 15 K=2,41,2
            PE=PE*XS/BK
            S=S+PE/(AK+3.0D0)
            AK=AK+1.0D0
            PO=2.0D0*XS*PO/AK
            TRMO=-PO/(AK+3.0D0)
            S=S+TRMO
            IF(DABS(TRMO).LE.TOL*DABS(S)) GOTO 20
            AK=AK+1.0D0
            BK=BK+1.0D0
   15     CONTINUE
   20     CONTINUE
          ANSJ22=(S+S)*T*RT
          RETURN
        ELSE
          KODE=3
          NZ=0
          ERFA=DRERF(X,KODE,NZ)
          S=X*ERFA+X*X/RTPI-DHERFC(X)
          ANSJ22=S/(A*AS)
          RETURN
        ENDIF
      ENDIF
      ART=A*RT
      IF(ART.LE.1.0D0) THEN
C       SMALL A*DSQRT(T).LE.1.0D0 SERIES FOR INTEGJ22
        KFORM=4
        ARG=(BS+CS)/T
        IF(ARG.GT.667.0D0) THEN
          ANSJ22=0.0D0
          J22ERR=5
          RETURN
        ENDIF
        IF(A.EQ.0.0D0) THEN
          M=6
        ELSE
          M=60
        ENDIF
        TOL=REL
        CALL GNSEQ(C,B,CAPT,M,TOL,YN)
        S1=YN(3)
        S2=5.0D0*YN(4)
        S3=8.0D0*YN(5)
        FK=ART+ART
        PK=-FK
        AK=7.0D0
        BK=15.0D0
        CK=7.0D0
        IFLAG1=0
        IFLAG2=0
        IFLAG3=0
        JFLAG=0
        TOL=REL/10.0D0
        DO 10 K=1,M-5
          IF(IFLAG1.EQ.0) THEN
            TRM1=PK*YN(K+3)
            S1=S1+TRM1
            IF(DABS(TRM1).LT.TOL*DABS(S1)) THEN
              IFLAG1=1
              JFLAG=JFLAG+1
            ENDIF
          ENDIF
          IF(IFLAG2.EQ.0) THEN
```

```
            TRM2=PK*YN(K+4)*AK
            S2=S2+TRM2
            IF(DABS(TRM2).LT.TOL*DABS(S2)) THEN
              IFLAG2=1
              JFLAG=JFLAG+1
            ENDIF
          ENDIF
          IF(IFLAG3.EQ.0) THEN
            TRM3=PK*YN(K+5)*BK
            S3=S3+TRM3
            IF(DABS(TRM3).LT.TOL*DABS(S3)) THEN
              IFLAG3=1
              JFLAG=JFLAG+1
            ENDIF
          ENDIF
          IF(JFLAG.EQ.3) GOTO 30
          PK=-PK*FK
          AK=AK+2.0D0
          CK=CK+2.0D0
C         BK=(K+2)*(K+4), B(K+1)=B(K)+2K+7=B(K)+CK
          BK=BK+CK
 10     CONTINUE
        J22ERR=-1
 30     CONTINUE
        COEF=4.0D0*DEXP(-ARG)
        ANSJ22=COEF*(BS*S1+B*RT*S2+T*S3)*RT
        RETURN
      ENDIF
      ARG=(BS+CS)/T
      N=1
      KODE=1
      M=2
      TOL=REL
      NZ=0
      CALL DEXINT(ARG, N, KODE, M, TOL, EN, NZ, IERR)
      IF(NZ.NE.0) THEN
        J22ERR=5
        ANSJ22=0.0D0
        KFORM=35
        RETURN
      ENDIF
      CAPB=A*RT+B/RT
      KODE=3
      NZ=0
      ERFB=DRERF(CAPB,KODE,NZ)
      COE=(B+0.5D0/A)/A
      S=(T*EN(2)-COE*EN(1))/(A*RTPI)+(RT/AS)*ERFB*DEXP(-ARG)
      CALL INTEGS1(A,B,C,T,TOL,S1,IERR,KFORM1)
      CALL INTEGS2(A,B,C,T,TOL,S2,IERR,KFORM2)
      KFORM=10*KFORM1+KFORM2
      ANSJ22=S+(-2.0D0*CS*RT*S1+RT*S2/A)/(AS*RTPI)
      RETURN
      END
      SUBROUTINE INTEGI29(A,B,T,N0,NN,YN,IERR)
C
C     DONALD E. AMOS   AUGUST, 2006
C-----------------------------------------------------------------------
C     A DOUBLE PRECISION ROUTINE
C-----------------------------------------------------------------------
C     REF: FOLDER 29, FOLDER 28
C
C     INTEGI29 GENERATES NN MEMBER SEQUENCES FOR THE INTEGRALS
C
C     INT ON (T,INF) OF EXP(-A^2*W^2)*I(N0+2*K-2)ERFC(B*W), K=1,NN
C
C     FOR ORDERS N0 TO N0+2*NN-2 IN INCREMENTS OF 2 WHICH ARE STORED IN
C     YN(*) IN LOCATIONS 1 THRU NN. IF N0 IS EVEN, WE GET A SEQUENCE OF
```

```
C     EVEN ORDERS FOR YN(K), K=1,NN. IF N0 IS ODD,  WE GET A SEQUENCE OF
C     ODD  ORDERS FOR YN(K), K=1,NN WITH N0.GE.0, NN.GE.1, A.GE.0.0D0,
C     B.GE.0.0D0, T.GE.0.0D0 BUT A AND B NOT BOTH ZERO AT THE SAME TIME.
C     I(M)ERFC(X) IS THE ITERATED COERROR FUNCTION OF ORDER M.
C
C     ERROR CONDITIONS
C      IERR=0 NORMAL RETURN, COMPUTATION COMPLETED
C      IERR=1 INPUT PARAMETER OUT OF RANGE, YN(K)=0.0D0, K=1,NN RETURNED
C      IERR=2 UNDERFLOW DURING COMPUTATION, COMPUTATION NOT COMPLETED,
C             YN(K)=0.0D0, K=1,NN  RETURNED
C      IERR=3 A LEGITIMATE UNDERFLOW. YN(K)=0.0D0, K=1,NN IS THE LOGICAL
C             RETURN FOR AN ANSWER WHICH IS OFF SCALE
C
C     INTEGI29 CALLS DINERFC
C-----------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION YA(300),YB(300),YN(NN)
      DATA RTPI /1.772453850905516D0/
      DATA REL  /0.5D-14/
      IERR=0
      IF ((A.LT.0.0D0).OR.(B.LT.0.0D0).OR.(T.LT.0.0D0)) THEN
        IERR=1
        GOTO 10
      ENDIF
      IF((A.EQ.0.0D0).AND.(B.EQ.0.D0)) THEN
        IERR=1
        GOTO 10
      ENDIF
      IF((N0.LT.0).OR.(NN.LT.1)) THEN
        IERR=1
        GOTO 10
      ENDIF
 10   CONTINUE
      IF(IERR.NE.0) THEN
        DO 15 K=1,NN
          YN(K)=0.0D0
 15     CONTINUE
        RETURN
      ENDIF
      IF(A.LE.B) THEN
        AA=A
        BB=B
        ICASE=1
      ELSE
        AA=B
        BB=A
        ICASE=2
      ENDIF
      A2=AA*AA
      B2=BB*BB
      D2=A2+B2
      T2=T*T
      ELIM=T2*D2
      IF(ELIM.GT.680.0D0) THEN
        IERR=3
        GOTO 10
      ENDIF
      EX=DEXP(-A2*T2)
C     GENERATE LAST MEMBER OF THE SEQUENCE
      NL=N0+2*NN-2
      NP1=NL+1
      IF(ICASE.EQ.1) THEN
        NI=NP1
        ND=0
      ELSE
        NI=0
        ND=NP1
```

```
      ENDIF
      X=BB*T
      KODE=1
      KLAST=75
      KN=2*KLAST+1+ND
      TOL=REL
      CALL DINERFC(X,NI,KODE,KN,TOL,YB,NZ)
      NGOOD=KN-NZ
      IF(ICASE.EQ.1) THEN
        SS=YB(1)
        MM=3
      ELSE
        SS=YB(NL+2)
        MM=NL+4
      ENDIF
      AN=0.5D0*DBLE(FLOAT(NL))+1.0D0
      FACN=AN
      C=4.0D0*A2/D2
      P=C
      DO 20 K=2,KLAST
        IF(YB(MM).EQ.0.0D0) THEN
          IERR=2
          GOTO 10
        ENDIF
        TRM=P*FACN*YB(MM)
        SS=SS+TRM
        IF(DABS(TRM).LT.DABS(SS)*REL) GOTO 21
        P=P*C
        AN=AN+1.0D0
        FACN=FACN*AN
        MM=MM+2
 20   CONTINUE
 21   CONTINUE
      COE=BB/D2
      TANS=SS*COE*EX
      IF(ICASE.EQ.1) THEN
        YN(NN)=TANS
      ELSE
        X=AA*T
        NI=0
        KODE=1
        KN=NP1
        TOL=REL
        CALL DINERFC(X,NI,KODE,KN,TOL,YA,NZ)
        IF(NZ.NE.0) THEN
          IERR=2
          GOTO 10
        ENDIF
        IF(NGOOD.LT.NP1) THEN
          IERR=2
          GOTO 10
        ENDIF
        SS=0.0D0
        RAB=AA/BB
        P=1.0D0
        IK=NP1
        DO 30 K=1,NP1
          SS=SS+P*YB(K)*YA(IK)
          P=-P*RAB
          IK=IK-1
  30    CONTINUE
        YN(NN)=0.5D0*RTPI*SS/BB+P*TANS
      ENDIF
      IF(NN.EQ.1) RETURN
C     RECUR BACKWARD FROM INDEX NL=N0+2*NN-2 TO INDEX N0 WITH THE
C     FORMULA Y(N-2)=RAB*(2*N*Y(N)+CC*I(N)ERFC(BT)), RAB=A^2/(A^2+B^2),
C     CC=B*EXP(-A^2*T^2)/A^2, WITH Y VALUES STORED CONSECUTIVELY IN YN.
```

```
      IF(ICASE.EQ.1) THEN
        X=B*T
        NI=N0
        KODE=1
        KN=2*NN-2
C       FIRST INDEX IS N0 AND THE LAST IS N0+2*NN-3=NL-1 IN LOCATIONS
C       1 THRU KN
        TOL=REL
        CALL DINERFC(X,NI,KODE,KN,TOL,YB,NZ)
        IF(NZ.NE.0) THEN
          IERR=2
          GOTO 10
        ENDIF
        LNB=KN
      ELSE
        DO 45 K=1,NL
          YB(K)=YA(K)
 45     CONTINUE
        EX=DEXP(-A*A*T2)
        LNB=NL
      ENDIF
      DN=DBLE(FLOAT(NL))
      TDN=DN+DN
      LN=NN
      CC=B*EX/(A*A)
      RAB=(A*A)/D2
      KLAST=NN-1
      DO 40 K=1,KLAST
        YN(LN-1)=RAB*(TDN*YN(LN)+CC*YB(LNB))
        TDN=TDN-4.0D0
        LN=LN-1
        LNB=LNB-2
 40   CONTINUE
      RETURN
      END
      DOUBLE PRECISION FUNCTION DVOFT(A,B,T,REL,IERR,KFORM)
C-----------------------------------------------------------------------
C     REF: FOLDER 21
C
C     DVOFT COMPUTES THE INTEGRAL
C
C     DVOFT=INT ON (0,T) OF EXP(A*A*W+2*A*B)*ERFC(A*RW+B/RW), RW=SQRT(W)
C
C     WHERE A AND B ARE NON-NEGATIVE AND REL IS THE PRECISION DESIRED.
C     IERR IS AN ERROR FLAG:
C
C     IERR= 0 IS A NORMAL RETURN
C         = 4 AN INPUT VARIABLE IS OUT OF RANGE. DVOFT=0.0D0 RETURNED
C         = 5 AN EXPONENTIAL UNDERFLOW OCCURRED. DVOFT=0.0D0 RETURNED
C         =-1 CONVERGENCE CRITERION NOT MET IN THE SERIES
C
C     KFORM INDICATES WHICH FORMULA OF FOLDER 21 WAS USED:
C
C     KFORM =0  AND IERR=0 MEANS A SPECIAL CASE: T=0 OR (A=0 AND B=0)
C           =2  MEANS THE CLOSED FORM WAS USED
C           =4  MEANS THE SERIES FOR SMALL A*SQRT(T).LE.1.0D0 WAS USED
C-----------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION SY(60)
      DATA RTPI /1.772453850905516D0/
      IERR=0
      KFORM=0
      IF(A.LT.0.0D0) THEN
        IERR=4
        DVOFT=0.0D0
        RETURN
      ENDIF
```

```
      IF(B.LT.0.0D0) THEN
        IERR=4
        DVOFT=0.0D0
        RETURN
      ENDIF
      IF(T.LT.0.0D0) THEN
        IERR=4
        DVOFT=0.0D0
        RETURN
      ENDIF
      IF((A.EQ.0.0D0).AND.(B.EQ.0.0D0)) THEN
        DVOFT=T
        RETURN
      ENDIF
      IF(T.EQ.0.0D0) THEN
        DVOFT=0.0D0
        RETURN
      ENDIF
      RT=DSQRT(T)
      CRT=A*RT
      IF (CRT.LE.1.00D0) THEN
C       SERIES FOR SMALL A*RT.LE.1.0D0
        KFORM=4
        CT=CRT+CRT
        ARG=B/RT
        N0=2
        KODE=2
        IF(A.EQ.0.0D0) THEN
          N=2
        ELSE
          N=40
        ENDIF
        CALL DINERFC(ARG,N0,KODE,N,REL,SY,NZ)
        S=0.0D0
        PW=1.0D0
        DO 10 K=1,N
          TRM=PW*SY(K)
          S=S+TRM
          IF(DABS(TRM).LE.DABS(S)*REL) GOTO 55
          PW=-PW*CT
 10     CONTINUE
        IERR=-1
C       PRINT *, 'END OF INERFC LOOP'
 55     CONTINUE
        ARG=B*B/T
        IF(ARG.GT.670.0D0) THEN
          IERR=5
          DVOFT=0.0D0
          RETURN
        ENDIF
        DVOFT=4.0D0*T*S*DEXP(-ARG)
        RETURN
      ENDIF
      KFORM=2
      BS=B*B
      AS=A*A
      ARG=A*RT+B/RT
      KODE=3
      NZ=0
      ERFX=DRERF(ARG,KODE,NZ)
      ARGB=BS/T
      IF(ARGB.GT.670.0D0) THEN
        IERR=5
        DVOFT=0.0D0
        RETURN
      ENDIF
      EX=DEXP(-ARGB)
```

```
      ARG=B/RT
      KODE=3
      NZ=0
      ERFB=DRERF(ARG,KODE,NZ)
      COE=2.0D0*RT/A
      KODE=2
      CALL DIERFC(ARG,KODE,ANS,IERR)
      DVOFT=(COE*ANS+(ERFX-ERFB)/AS)*EX
      RETURN
      END
      SUBROUTINE INTEGS1(A,B,C,T,TOL,S1,IERR,KFORM)
C
C     DONALD E. AMOS, OCTOBER, 2002
C-----------------------------------------------------------------------
C     A DOUBLE PRECISION ROUTINE
C-----------------------------------------------------------------------
C     INTEGS1 COMPUTES THE INTEGRAL
C
C     S1=INT ON (CAPB,INF) OF EXP(A*A*T+2.0D0*A*B-C*C/T)*EXP(-W*W)/
C                               (C*C+T*(W-A*RT)**2)
C
C     WHERE  CAPB=A*RT+B/RT, RT=DSQRT(T)
C            A IS NON-NEGATIVE, B IS NON-NEGATIVE AND C IS POSITIVE   OR
C            C IS ZERO WITH B POSITIVE AND A NON-NEGATIVE
C
C            TOL IS THE ERROR TOLERANCE FOR THE COMPUTATION
C
C            KFORM=0 A SPECIAL FORMULA IS EVALUATED FOR A SPECIAL CASE
C                 =3 FORMULA FOR CAPL.GE.2.0D0 IS EVALUATED
C                 =5 QUADRATURE BY DQUAD8 IS USED
C
C            IERR= 0 IS A NORMAL RETURN
C            IERR= 5 IS AN UNDERFLOW: (B*B+C*C)/T IS TOO LARGE
C            IERR=-1 A CONVERGENCE CONDITION NOT MET
C                = ANY OTHER VALUE IS AN ERROR FROM DGAUS8
C-----------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION EN(55),ENH(55)
      COMMON /CQFS1/CCS,CRT,CB,CTCAPB
      EXTERNAL QFS1
      DATA HPI /1.5707963267948966D0/
      DATA RTPI /1.772453850905516D0/
      IERR=0
      KFORM=0
      IF(T.EQ.0.0D0) THEN
        S1=0.0D0
        RETURN
      ENDIF
      RT=DSQRT(T)
      AS=A*A
      BS=B*B
      CS=C*C
      IF((A.EQ.0.0D0).AND.(B.EQ.0.0D0)) THEN
C       C IS POSITIVE HERE IF INPUT IS OBSERVED
        X=C/RT
        KODE=2
        ERFX=DRERF(X,KODE,NZ)
        IF(NZ.NE.0) THEN
           IERR=5
           S1=0.0D0
           RETURN
        ENDIF
        S1=HPI*ERFX/(C*RT)
        RETURN
      ENDIF
      IF(A.EQ.0.0D0) THEN
        CAPT=1.0D0/RT
```

```
        CALL INTEGI5(C,B,CAPT,ANSI5,JERR)
        IF(JERR.EQ.4) THEN
          S1=0.0D0
          IERR=5
          RETURN
        ENDIF
        S1=RTPI*ANSI5/RT
        RETURN
      ENDIF
      CAPB=A*RT+B/RT
      CAPX=(BS+CS)/T
      IF(CAPX.GT.667.0D0) THEN
        S1=0.0D0
        IERR=5
        RETURN
      ENDIF
      ARG=CS/T+AS*T
      DEN=DSQRT(ARG)
      CAPL=CAPB/DEN
      IF(CAPL.LT.2.0D0) THEN
C       S1 BY QUADRATURE
        KFORM=5
        CB=B
        CRT=RT
        CCS=CS
        TCAPB=CAPB+CAPB
        CTCAPB=TCAPB
        X1=0.0D0
        SIG1=0.5D0/CAPB
        SIG2=0.707106781181865475D0
        SIG=4.0D0*MIN(SIG1,SIG2)
        INIT=0
        DO 5 JR=1,5
          ETOL=TOL
          CALL DQUAD8(QFS1,INIT,X1,SIG,ETOL,X2,QANS,IERR)
C         TRUNCATION ERROR ESTIMATE; CONTINUE INTEGRATION IF NOT MET
          BR=DEXP(-(TCAPB+X2)*X2)/(CS+(B+X2*RT)**2)
          BR=0.5D0*BR/X2
          IF(DABS(BR).LE.DABS(TOL*QANS)) GO TO 6
 5      CONTINUE
        IERR=-1
 6      CONTINUE
        IF(IERR.EQ.1) THEN
          IERR=0
        ENDIF
        S1=QANS*DEXP(-CAPX)
        RETURN
      ELSE
C       S1 BY SERIES IN 1/CAPL**K
        KFORM=3
        ARG=CAPB*CAPB
        N=2
        KODE=2
        M=40
        NZ=0
        IERR=0
        CALL DEXINT(ARG, N, KODE, M, TOL, EN, NZ, IERR)
        KODE=2
        FNH=1.5D0
        MP=M+1
        CALL DHEXINT(ARG,FNH,KODE,MP,TOL,ENH,NZ,IERR)
        U1=1.0D0
        X=A*RT/DEN
        T1=X+X
        U2=T1
        U3=T1*U2-U1
        S=ENH(1)
```

```
        TRMS=1.0D0/CAPL
        REL=TOL/10.0D0
        AK=1.0D0
        DO 10 K=1,59,2
          I=(K+1)/2
          S=S+U2*EN(I)*TRMS
          U1=U2
          U2=U3
          U3=T1*U2-U1
          TRMS=TRMS/CAPL
          S=S+U2*ENH(I+1)*TRMS
          AK=AK+2.0D0
          IF(DABS(TRMS*AK).LT.REL) GO TO 55
          U1=U2
          U2=U3
          U3=T1*U2-U1
          TRMS=TRMS/CAPL
 10     CONTINUE
        IERR=-1
 55     CONTINUE
        S=S/CAPB
        S1=0.5D0*S*DEXP(-CAPX)/T
        RETURN
      ENDIF
      END
      DOUBLE PRECISION FUNCTION QFS1(W)
C     INTEGRAND FOR DQUAD8 IN INTEGS1
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON /CQFS1/CCS,CRT,CB,CTCAPB
      DEN=CCS+(CB+CRT*W)**2
      QFS1=DEXP(-(CTCAPB+W)*W)/DEN
      RETURN
      END
      SUBROUTINE INTEGS2(A,B,C,T,TOL,S2,IERR,KFORM)
C
C     DONALD E. AMOS, OCTOBER, 2002
C------------------------------------------------------------------------
C     A DOUBLE PRECISION ROUTINE
C------------------------------------------------------------------------
C     INTEGS2 COMPUTES THE INTEGRAL
C
C     S2=INT ON (CAPB,INF) OF RT*EXP(A*A*T+2.0D0*A*B-C*C/T)*EXP(-W*W)*
C                              (W-A*RT)/(C*C+T*(W-A*RT)**2)
C
C     WHERE  CAPB=A*RT+B/RT, RT=DSQRT(T)
C            A IS NON-NEGATIVE, B IS NON-NEGATIVE AND C IS POSITIVE   OR
C            C IS ZERO WITH B POSITIVE AND A NON-NEGATIVE
C
C            TOL IS THE ERROR TOLERANCE FOR THE COMPUTATION
C            KFORM=0 A SPECIAL FORMULA IS EVALUATED FOR A SPECIAL CASE
C                 =3 FORMULA FOR CAPL.GE.2.0D0 IS EVALUATED
C                 =5 QUADRATURE BY DQUAD8 IS USED
C
C            IERR= 0 IS A NORMAL RETURN
C                = 5 IS AN UNDERFLOW: (B*B+C*C)/T IS TOO LARGE
C                =-1 A CONVERGENCE CONDITION NOT MET
C                = ANY OTHER VALUE IS AN ERROR FROM DGAUS8
C------------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION EN(55),ENH(55)
      COMMON /CQFS2/CCS,CRT,CB,CTCAPB
      EXTERNAL QFS2
      IERR=0
      KFORM=0
      IF(T.EQ.0.0D0) THEN
        S2=0.0D0
        RETURN
```

```
      ENDIF
      RT=DSQRT(T)
      IF(A.EQ.0.0D0) THEN
        ARG=(CS+BS)/T
        N=1
        KODE=1
        M=1
        NZ=0
        CALL DEXINT(ARG, N, KODE, M, TOL, EN, NZ, J22ERR)
        IF(NZ.NE.0) THEN
          S2=0.0D0
          IERR=5
          RETURN
        ENDIF
        S2=0.5D0*EN(1)/RT
        RETURN
      ENDIF
C     A IS POSITIVE HERE
      AS=A*A
      BS=B*B
      CS=C*C
      ARG=CS/T+AS*T
      DEN=DSQRT(ARG)
      CAPB=A*RT+B/RT
      CAPX=(BS+CS)/T
      IF(CAPX.GT.667.0D0) THEN
        S2=0.0D0
        IERR=5
        RETURN
      ENDIF
      CAPL=CAPB/DEN
      IF(CAPL.LT.2.0D0) THEN
C       S2 BY QUADRATURE
        KFORM=5
        CB=B
        CRT=RT
        CCS=CS
        TCAPB=CAPB+CAPB
        CTCAPB=TCAPB
        X1=0.0D0
        SIG=3.0D0
        INIT=0
        DO 5 JR=1,5
          REL=TOL
          CALL DQUAD8(QFS2,INIT,X1,SIG,REL,X2,QANS,IERR)
C         TRUNCATION ERROR ESTIMATE; CONTINUE INTEGRATION IF NOT MET
          BR=DEXP(-(TCAPB+X2)*X2)/DSQRT(CS+(B+X2*RT)**2)
          BR=0.5D0*BR/X2
          IF(DABS(BR).LE.DABS(TOL*QANS)) GO TO 6
 5      CONTINUE
        IERR=-1
 6      CONTINUE
        IF(IERR.EQ.1) THEN
          IERR=0
        ENDIF
        S2=QANS*DEXP(-CAPX)
        RETURN
      ELSE
C       S2 BY SERIES IN 1/CAPL**K
        KFORM=3
        ARGB=CAPB*CAPB
        N=1
        KODE=2
        M=40
        NZ=0
        IERR=0
        CALL DEXINT(ARGB, N, KODE, M, TOL, EN, NZ, IERR)
```

```
          KODE=2
          FNH=1.5D0
          MP=M+1
          CALL DHEXINT(ARGB,FNH,KODE,MP,TOL,ENH,NZ,IERR)
          X=A*RT/DEN
          TX=X+X
          TM=1.0D0
          TK=X
          TP=TX*TK-TM
          S=EN(1)
          TRMS=1.0D0/CAPL
          REL=TOL/10.0D0
          AK=1.0D0
          DO 10 K=1,59,2
            I=(K+1)/2
            S=S+TK*ENH(I)*TRMS
            TM=TK
            TK=TP
            TP=TX*TK-TM
            TRMS=TRMS/CAPL
            S=S+TK*EN(I+1)*TRMS
            AK=AK+2.0D0
            IF(DABS(TRMS*AK).LT.REL) GO TO 55
            TM=TK
            TK=TP
            TP=TX*TK-TM
            TRMS=TRMS/CAPL
 10       CONTINUE
          IERR=-1
 55       CONTINUE
          S2=0.5D0*S*DEXP(-CAPX)/RT
          RETURN
        ENDIF
        END
        DOUBLE PRECISION FUNCTION QFS2(W)
C       INTEGRAND FOR DQUAD8 IN INTEGS2
        IMPLICIT DOUBLE PRECISION (A-H,O-Z)
        COMMON /CQFS2/CCS,CRT,CB,CTCAPB
        DIFF=CB+CRT*W
        DEN=CCS+DIFF*DIFF
        QFS2=DIFF*DEXP(-(CTCAPB+W)*W)/DEN
        RETURN
        END
        SUBROUTINE GNSEQ(A,B,CAPT,M,REL,YN)
C
C       YN(N+1) CONTAINS 2*G(A,B,N)*(CAPT**(N-1))*DEXP(CAPX), N=0,...,M-1
C       WHERE CAPX=(A*A+B*B)*CAPT*CAPT, CAPT=1/DSQRT(T)
C
C       DIMENSION YN BY AT LEAST M
C
        IMPLICIT DOUBLE PRECISION (A-H,O-Z)
        DIMENSION SY(100),EN(1),YN(M)
        DATA RTPI       /1.772453850905516D0/
        COMMON/CGSEQ/ KN,CREL,CCAPX,CBT
        EXTERNAL DGNFUN
        A2=A*A
        B2=B*B
        CAPX=(A2+B2)*CAPT*CAPT
        BT=B*CAPT
C       NFB IS THE INDEX AT WHICH FORWARD AND BACKWARD RECURRENCE STARTS
C       IF NFB.LT.3 ALL INDICES ARE COMPUTED BY FORWARD RECURRENCE
        ARG=CAPX+CAPX
        NFB=INT(SNGL(ARG))
        IF (NFB.LT.3) THEN
          CALL INTEGI5(A,B,CAPT,ANSI5,IERR)
          TRM=DEXP(CAPX)*ANSI5
          S=TRM+TRM
```

```
          YN(1)=S/CAPT
          IF(M.EQ.1) RETURN
          N=1
          KODE=2
          MM=1
          NZ=0
          IERR=0
          CALL DEXINT(CAPX, N, KODE, MM, REL, EN, NZ, IERR)
          YN(2)=-B*S+EN(1)/RTPI
          IF(M.EQ.2) RETURN
          N0=0
          KODE=2
          MM=M
          CALL DINERFC(BT,N0,KODE,MM,REL,SY,NZ)
          NN=1
          N=M-2
          CALL GFORW(CAPX,BT,NN,N,M,SY,YN)
          RETURN
        ELSE
C         NFB.GE.3
C         SET COMMON PARAMETERS FOR DGNFUN
          KS=MIN(NFB,M-1)
          KN=KS
          CREL=REL
          CCAPX=CAPX
          CBT=BT
          INIT=0
          X1=0.0D0
          SIG1=1.0D0/DSQRT(CAPX)
          SIG2=0.5/CAPX
          DSIG=3.0D0*MIN(SIG1,SIG2)
          DREL=REL
          CALL DQUAD8(DGNFUN,INIT,X1,DSIG,DREL,X2,DQANS,IERR)
          YN(KS+1)=DQANS
          IF(M.EQ.1) RETURN
          KN=KS-1
          CREL=REL
          INIT=0
          X1=0.0D0
          DREL=REL
          CALL DQUAD8(DGNFUN,INIT,X1,DSIG,DREL,X2,DQANS,IERR)
          YN(KS)=DQANS
          IF(M.EQ.2) RETURN
          N=M-KS-1
          IF(N.GT.0)THEN
            MM=M
            N0=0
            KODE=2
            CALL DINERFC(BT,N0,KODE,MM,REL,SY,NZ)
            CALL GFORW(CAPX,BT,KS,N,M,SY,YN)
            CALL GBACK(CAPX,BT,KS,M,SY,YN)
            RETURN
          ELSE
            MM=KS
            N0=0
            KODE=2
            CALL DINERFC(BT,N0,KODE,MM,REL,SY,NZ)
            CALL GBACK(CAPX,BT,KS,M,SY,YN)
            RETURN
          ENDIF
        ENDIF
        END
        SUBROUTINE GFORW(CAPX,BT,NSTART,NF,M,SY,YN)
C
C       YN FOR INDICES NSTART AND NSTART+1 ARE KNOWN. THE FIRST
C       RECURRENCE GENERATES NSTART+2. NF IS THE NUMBER OF FORWARD
C       RECURRENCES FOR INDICES NSTART THROUGH NSTART+NF+1. M= NUMBER
```

```
C     OF MEMBERS IN THE SEQUENCE.
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION SY(100),YN(M)
      J1=NSTART+2
      AJ=DBLE(FLOAT(J1-2))
      TAJ=AJ+AJ
      AJP=AJ+1.0D0
      PROD=AJ*AJP
      BJ=2.0D0*AJP
      DO 10 J=1,NF
        YN(J1)=(SY(J1-2)-TAJ*BT*YN(J1-1)-CAPX*YN(J1-2))/PROD
        J1=J1+1
        TAJ=TAJ+2.0D0
        PROD=PROD+BJ
        BJ=BJ+2.0D0
   10 CONTINUE
      RETURN
      END
      SUBROUTINE GBACK(CAPX,BT,NSTART,M,SY,YN)
C
C     YN MEMBERS FOR INDICES NSTART AND NSTART+1 ARE KNOWN. YN MEMBERS
C     FOR INDICES NSTART-1 DOWN TO 1 ARE GENERATED. M=NUMBER OF MEMBERS
C     IN THE SEQUENCE.
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION SY(100),YN(M)
      IF(NSTART.EQ.1) RETURN
      MM=NSTART-1
      AJ=DBLE(FLOAT(MM))
      TAJ=AJ+AJ
      PROD=AJ*(AJ+1.0D0)
      BJ=TAJ
      KJ=MM
      DO 10 J=1,MM
        YN(KJ)=(SY(KJ)-TAJ*BT*YN(KJ+1)-PROD*YN(KJ+2))/CAPX
        KJ=KJ-1
        TAJ=TAJ-2.0D0
        PROD=PROD-BJ
        BJ=BJ-2.0D0
   10 CONTINUE
      RETURN
      END
      DOUBLE PRECISION FUNCTION DGNFUN(W)
C
C     DGNFUN COMPUTES THE INTEGRAND OF G FOR THE QUADRATURES
C              YN(KN+1)=2.0D0*G(A,B,KN)*(RT**(KN-1))*DEXP(CCAPX)
C              YN(KN)=2.0D0*G(A,B,NK-1)*(RT**(KN-2))*DEXP(CCAPX)
C     TO START THE RECURRENCE FOR THE YN SEQUENCE. HERE KN=MIN(NFB,M-1),
C     NFB=INT(SNGL(2.0D0*CCAPX)), CCAPX=(A*A+B*B)/T, RT=DSQRT(T),
C     M=NUMBER IN THE SEQUENCE. CBT=B*CAPT=B/RT
C
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      DIMENSION Y(1)
      COMMON/CGSEQ/ KN,CREL,CCAPX,CBT
      ARG=-CCAPX*(2.0D0+W)*W
      TM=DEXP(ARG)
      U=1.0D0+W
      ARG=CBT*U
      KODE=2
      N=1
      NZ=0
      CALL DINERFC(ARG,KN,KODE,N,CREL,Y,NZ)
      S=TM*Y(1)/U**KN
      DGNFUN=S+S
      RETURN
      END
```

```
      PROGRAM BPRGINFO
C-----------------------------------------------------------------------
C     DONALD E. AMOS, MAY 1, 2002,   FEBRUARY, 2006
C
C     THIS FILE CONTAINS THE DRIVERS WHICH WILL EXERCIZE SUBROUTINES IN
C     FILE BECKSUBS.FOR. THEY SERVE TO DEMOMSTRATE CALL PROCEDURES, A
C     NOMINAL USE OF ERROR FLAGS AND TYPICAL RELATIVE ERRORS. FILE
C     AMOSSUBS.FOR MUST ALSO BE ATTACHED TO EXECUTE THESE DRIVERS.
C
C     USAGE:
C
C     EXTRACT ONE OF THE PROGRAMS BELOW ALONG WITH ANY SUCCEEDING
C     DOUBLE PRECISION FUNCTION(S) OR SUBROUTINES, COMPILE, LINK
C     COMPILED FILES BECKSUBS.FOR AND AMOSSUBS.FOR, AND THEN EXECUTE THE
C     LINKED FILES. OUTPUT IS WRITTEN TO A TEXT FILE (*.TXT) WITH THE
C     NAME OF THE PROGRAM BEING EXECUTED.
C
C     PROGRAMS:                      OUTPUT:                 FOLDER
C              PROGRAM I1COMP               I1COMP.TXT       1,2,10
C              PROGRAM I2COMP               I2COMP.TXT         9
C              PROGRAM I9COMP               I9COMP.TXT         9
C              PROGRAM I3COMP               I3COMP.TXT         7
C              PROGRAM I5COMP               I5COMP.TXT         5
C              PROGRAM J5COMP               J5COMP.TXT         5
C              PROGRAM V5COMP               V5COMP.TXT         5
C              PROGRAM I6COMP               I6COMP.TXT       3,6,15
C              PROGRAM W3COMP               W3COMP.TXT         10
C              PROGRAM PCOMP                PCOMP.TXT          11
C              PROGRAM QCOMP                QCOMP.TXT          11
C              PROGRAM I21COMP              I21COMP.TXT        21
C              PROGRAM J21COMP              J21COMP.TXT        21
C              PROGRAM I22COMP              I22COMP.TXT        22
C              PROGRAM J22COMP              J22COMP.TXT        22
C              PROGRAM I29COMP              I29COMP.TXT        29
C              PROGRAM GNCOMP               GNCOMP.TXT         21
C              PROGRAM VTCOMP               VTCOMP.TXT         21
C-----------------------------------------------------------------------
      END
      PROGRAM I1COMP
C
C     DONALD E. AMOS   JANUARY, 2003; APRIL, 2006
C
C     REF: FOLDER 1, FOLDER 2, FOLDER 10
C
C     I1COMP COMPARES THE CLOSED FORM OF I1 OF FOLDERS 1, 2 AND 10
C     WITH A QUADRATURE FOR I1 FOR KODE=1:
C
C          I1=INT ON (T,INF) OF EXP(-A^2*W^2)*ERF(B*W)/W^2
C
C     AND THE COMPLEMETARY FORM FOR KODE=2:
C
C         I1C=INT ON (T,INF) OF EXP(-A^2*W^2)*ERFC(B*W)/W^2
C
C     WHERE A.GE.0.0D0, B.GE.0.0D0, AND T.GT.0.0D0. THE CLOSED FORM IS
C     IMPLEMENTED IN DOUBLE PRECISION SUBROUTINE INTEGI1.
C
C     OUTPUT IS WRITTEN TO FILE I1COMP.TXT
C----------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      OPEN(UNIT=7,FILE="I1COMP.TXT")
      DO 5 KODE=2,2
      WRITE(7,115) KODE
 115  FORMAT(/,'                           KODE = ',I3/)
      WRITE (7,110)
 110  FORMAT('       A          B           T          QUAD          ERR
     *     IERR IERR1')
      DO 10 IT=1,101,10
```

```
        T=DBLE(FLOAT(IT))/40.0D0
        DO 20 IA=1,101,10
          A=DBLE(FLOAT(IA-1))/40.0D0
          DO 30 IB=1,101,10
            B=DBLE(FLOAT(IB-1))/40.0D0
            REL=0.50D-14
            CALL INTEGI1(A,B,T,KODE,REL,ANS1,IERR1)
            CALL I1QUAD(A,B,T,KODE,QANS,IERR)
            ERR=DABS(QANS-ANS1)
            IF(QANS.NE.0.0D0) THEN
              ERR=ERR/DABS(QANS)
            ENDIF
            WRITE(7,100) A,B,T,QANS,ERR,IERR,IERR1
 100        FORMAT(3D12.4,D13.5,D12.4,2I5)
 101        FORMAT(3I5)
   30     CONTINUE
   20   CONTINUE
   10 CONTINUE
   5  CONTINUE
      END
      SUBROUTINE I1QUAD(A,B,T,KODE,QANS,IERR)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION EN(1)
      COMMON/ CQFI1/ CA2,CB,CB2,KKODE
      DATA RTPI       /1.772453850905516D0/
      DATA REL /0.50D-14/
      EXTERNAL DQI1
      IERR=0
      IF(B.EQ.0.0D0) THEN
        IF(KODE.EQ.1) THEN
          QANS=0.0D0
          GOTO 40
        ELSE
          IF(A.EQ.0.0D0) THEN
            QANS=1.0D0/T
            GOTO40
          ENDIF
        ENDIF
      ENDIF
      IF (A.EQ.0.0D0) THEN
        IF(KODE.EQ.1) THEN
          BT=B*T
          IF(BT.GE.6.0D0) THEN
            QANS=1.0D0/T
          ELSE
            ERFB=DRERF(BT,KODE,NZ)
            BT2=BT*BT
            N=1
            IKODE=1
            M=1
            TOL=REL
            NZ=0
            IERR=0
            CALL DEXINT(BT2, N, IKODE, M, TOL, EN, NZ, IERR)
            QANS=ERFB/T+B*EN(1)/RTPI
          ENDIF
          GOTO 40
        ELSE
          XXA=B
        ENDIF
      ELSE
        XXA=DSQRT(A*A+B*B)
      ENDIF
C     SET COMMON VARIABLES FOR QUADRATURE
      CA2=A*A
      CB=B
      CB2=CB*CB
```

```
      KKODE=KODE
      SIG=3.0D0/XXA
      X1=T
      TOL=REL
      INIT=0
      CALL DQUAD8(DQI1,INIT,X1,SIG,TOL,X2,QANS,IERR)
C     GET BETTER ESTIMATE OF THE TRUNCATION ERROR
      TOL=REL
      CALL DQUAD8(DQI1,INIT,X1,SIG,TOL,X2,QANS,IERR)
 40   CONTINUE
      RETURN
      END
      DOUBLE PRECISION FUNCTION DQI1(T)
C
C     DQI1 COMPUTES THE INTEGRAND FOR THE QUADRATURE OF I1 AND I1C
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/ CQFI1/ CA2,CB,CB2,KKODE
      T2=T*T
      ARG=CB*T
      IF(KKODE.EQ.1) THEN
        ERFX=DRERF(ARG,KKODE,NZ)
        DQI1=ERFX*DEXP(-CA2*T2)/T2
      ELSE
        IKODE=3
        ERFX=DRERF(ARG,IKODE,NZ)
        ARG=(CA2+CB2)*T2
        DQI1=ERFX*DEXP(-ARG)/T2
      ENDIF
      RETURN
      END
      PROGRAM I2COMP
C
C     DONALD E. AMOS    DECEMBER, 2005; APRIL, 2006
C
C     REF: FOLDER 9, FOLDER 7
C
C     ON KODE=1, I2COMP COMPUTES THE INTEGRAL
C
C                I2=INT ON (0,T) OF ERF(A*W)*ERF(B*W);
C
C     ON KODE=2, I2COMP COMPUTES  THE INTEGRAL
C
C                I2C=INT ON (T,INF) OF ERFC(A*W)*ERFC(B*W)
C
C     AND EACH INTEGRAL IS COMPARED WITH ITS RESPECTIVE QUADRATURE,
C     WHERE     A.GT.0.0D0, B.GE.0.0D0, AND T.GT.0.0D0.
C
C     OUTPUT IS WRITTEN TO FILE I2COMP.TXT
C-----------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DATA REL/0.5D-14/
      OPEN(UNIT=7,FILE="I2COMP.TXT")
      DO 5 KODE=1,2
        KKODE=KODE
        WRITE(7,109) KODE
 109    FORMAT(/,27X,"KODE = ",I1)
        WRITE (7,110)
 110    FORMAT('      A          B           T           QUAD       REL
     &ERR     IERR  IERR2')
        DO 10 IT=1,21,2
          T=DBLE(FLOAT(IT))/5.0D0
          DO 20 IA=1,21,2
            A=DBLE(FLOAT(IA-1))/1.0D0
            DO 30 IB=3,21,2
              B=DBLE(FLOAT(IB-1))/1.0D0+REL
              CALL INTEGI2(A,B,T,KODE,ANSI2,IERR2)
```

```
            CALL I2QUAD(A,B,T,KODE,QANS,IERR)
            ERR=DABS(QANS-ANSI2)
            IF(QANS.NE.0.0D0) THEN
              ERR=ERR/DABS(QANS)
            ENDIF
            WRITE(7,100) A,B,T,QANS,ERR,IERR,IERR2
 100        FORMAT(3D12.4,D13.5,D12.4,2I6)
   30     CONTINUE
   20   CONTINUE
   10 CONTINUE
   5  CONTINUE
      END
      SUBROUTINE I2QUAD(A,B,T,KODE,QANS,IERR)
C     I2 BY QUADRATURE
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/ CFI2/ CA,CB,KKODE
      DATA TOL/0.5D-14/
      EXTERNAL DQI2
      CA=A
      CB=B
      KKODE=KODE
      IF(KODE.EQ.1) THEN
        XA=MIN(A,B)
        IF((XA.EQ.0.0D0).OR.(T.EQ.0.0D0)) THEN
          QANS=0.0D0
          IERR=1
          RETURN
        ENDIF
        WSM=MIN(6.0D0/A,6.0D0/B)
        WBM=MAX(6.0D0/A,6.0D0/B)
        IF(T.LE.WSM) THEN
C         INTERVAL 0.LE.T.LE.WSM
          MMQ=INT(SNGL(T))+2
          MQ=MIN(MMQ,20)
          SIG=T/DBLE(FLOAT(MQ))
          INIT=-MQ
          X1=0.0D0
          REL=TOL
          CALL DQUAD8(DQI2,INIT,X1,SIG,REL,X2,QANS,IERR)
        ELSE
          IF(T.LE.WBM) THEN
C           INTERVAL (0,WSM]
C           ADJUST THE NUMBER OF SIG INTERVALS ACCORDING TO THE LENGTH
C           OF THE QUADRATURE INTERVAL
            MMQ=INT(SNGL(WSM))+2
            MQ=MIN(MMQ,20)
            SIG=WSM/DBLE(FLOAT(MQ))
            INIT=-MQ
            X1=0.0D0
            REL=TOL
            CALL DQUAD8(DQI2,INIT,X1,SIG,REL,X2,ANS,IERR)
            QANS=ANS
C           INTERVAL (WSM,WBM]
            XL=T-WSM
            MMQ=INT(SNGL(XL))+2
            MQ=MIN(MMQ,20)
            SIG=XL/DBLE(FLOAT(MQ))
C           TEST TO DETERMINE IF T IS TOO CLOSE TO WSM TO GET AN
C           ACCURATE QUADRATURE (SUPPRESS AN ERROR MESSAGE FROM DGAUS8)
            XMID=0.5D0*(T+WSM)
            FI2=DQI2(XMID)
            IF(FI2*SIG.LE.2.0D0*REL*QANS) GOTO 51
            INIT=-MQ
            X1=WSM
            REL=TOL
            CALL DQUAD8(DQI2,INIT,X1,SIG,REL,X2,ANS,IERR)
            QANS=QANS+ANS
```

```
 51         CONTINUE
          ELSE
C           INTERVAL (0,WSM]
            MMQ=INT(SNGL(WSM))+2
            MQ=MIN(MMQ,20)
            SIG=WSM/DBLE(FLOAT(MQ))
            INIT=-MQ
            X1=0.0D0
            REL=TOL
            CALL DQUAD8(DQI2,INIT,X1,SIG,REL,X2,ANS,IERR)
            QANS=ANS
C           INTERVAL (WSM,WBM]
            XL=WBM-WSM
            MMQ=INT(SNGL(XL))+2
            MQ=MIN(MMQ,20)
            SIG=XL/DBLE(FLOAT(MQ))
C           SEE IF WSM AND WBM ARE TOO CLOSE TOGETHER TO GET AN ACCURATE
C           ANSWER FROM DGAUS8
            XMID=0.5D0*(WSM+WBM)
            FI2=DQI2(XMID)
            IF(FI2*SIG.LE.2.0D0*REL*QANS) GOTO 52
            INIT=-MQ
            X1=WSM
            REL=TOL
            CALL DQUAD8(DQI2,INIT,X1,SIG,REL,X2,ANS,IERR)
            QANS=QANS+ANS
 52         CONTINUE
C           INTERVAL (WBM,T]
            XL=T-WBM
            MMQ=INT(SNGL(XL))+2
            MQ=MIN(MMQ,20)
            SIG=XL/DBLE(FLOAT(MQ))
C           SEE IF T AND WBM ARE TOO CLOSE TOGETHER TO GET AN ACCURATE
C           ANSWER FROM DGAUS8
            XMID=0.5D0*(T+WBM)
            FI2=DQI2(XMID)
            IF(FI2*SIG.LE.2.0D0*REL*QANS) GOTO 53
            INIT=-MQ
            X1=WBM
            REL=TOL
            CALL DQUAD8(DQI2,INIT,X1,SIG,REL,X2,ANS,IERR)
            QANS=QANS+ANS
 53         CONTINUE
          ENDIF
        ENDIF
      ELSE
        INIT=0
        X1=T
        A2=A*A
        B2=B*B
        SIG=6.0D0/DSQRT(A2+B2)
        REL=TOL
        CALL DQUAD8(DQI2,INIT,X1,SIG,REL,X2,QANS,IERR)
      ENDIF
      RETURN
      END
      DOUBLE PRECISION FUNCTION DQI2(T)
C
C     DQI2 COMPUTES THE INTEGRAND FOR FUNCTIONS I2 AND I2C
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/ CFI2/ CA,CB,KKODE
      ARG=CA*T
      ERFA=DRERF(ARG,KKODE,NZ)
      ARG=CB*T
      ERFB=DRERF(ARG,KKODE,NZ)
      DQI2=ERFA*ERFB
```

```
      RETURN
      END
      PROGRAM I9COMP
C
C     DONALD E. AMOS    DECEMBER, 2005; APRIL, 2006
C
C     REF: FOLDER 9, FOLDER 7
C
C     ON KODE=1, I9COMP COMPUTES THE INTEGRAL
C
C                I9=INT ON (0,T) OF W*ERF(A*W)*ERF(B*W);
C
C     ON KODE=2, I9COMP COMPUTES THE COMPLEMENTARY INTEGRAL
C
C                I9C=INT ON (T,INF) OF W*ERFC(A*W)*ERFC(B*W)
C
C     AND EACH INTEGRAL IS COMPARED WITH ITS RESPECTIVE QUADRATURE,
C     WHERE     A.GT.0.0D0, B.GE.0.0D0, AND T.GT.0.0D0.
C
C     OUTPUT IS WRITTEN TO FILE I9COMP.TXT
C-----------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DATA REL/0.50D-14/
      OPEN(UNIT=7,FILE="I9COMP.TXT")
      DO 5 KODE=1,2
        WRITE(7,109) KODE
 109    FORMAT(/,27X,"KODE = ",I1)
        WRITE (7,110)
 110    FORMAT('       A          B           T           QUAD        REL
     &ERR     IERR  IERR9')
        DO 10 IT=1,21,2
          T=DBLE(FLOAT(IT))/5.0D0
          DO 20 IA=1,21,2
            A=DBLE(FLOAT(IA-1))/1.0D0
            DO 30 IB=3,21,2
              B=DBLE(FLOAT(IB-1))/1.0D0+REL
              CALL INTEGI9(A,B,T,KODE,ANSI9,IERR9)
              CALL I9QUAD(A,B,T,KODE,QANS,IERR)
              ERR=DABS(QANS-ANSI9)
              IF(QANS.NE.0.0D0) THEN
                ERR=ERR/DABS(QANS)
              ENDIF
              WRITE(7,100) A,B,T,QANS,ERR,IERR,IERR9
 100          FORMAT(3D12.4,D13.5,D12.4,2I6)
  30        CONTINUE
  20      CONTINUE
  10    CONTINUE
   5  CONTINUE
      END
      SUBROUTINE I9QUAD(A,B,T,KODE,QANS,IERR)
C     I9 BY QUADRATURE
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/ CFI9/ CA,CB,KKODE
      DATA TOL/0.50D-14/
      EXTERNAL DQI9
      CA=A
      CB=B
      KKODE=KODE
      IF(KODE.EQ.1) THEN
        XA=MIN(A,B)
        IF((XA.EQ.0.0D0).OR.(T.EQ.0.0D0)) THEN
          QANS=0.0D0
          IERR=1
          RETURN
        ENDIF
        WSM=MIN(6.0D0/A,6.0D0/B)
        WBM=MAX(6.0D0/A,6.0D0/B)
```

```
        IF(T.LE.WSM) THEN
C         INTERVAL (0,T]
          MMQ=INT(SNGL(T))+2
          MQ=MIN(MMQ,20)
          SIG=T/DBLE(FLOAT(MQ))
          INIT=-MQ
          X1=0.0D0
          REL=TOL
          CALL DQUAD8(DQI9,INIT,X1,SIG,REL,X2,QANS,IERR)
        ELSE
          IF(T.LE.WBM) THEN
C           INTERVAL (0,WSM]
C           MAKE NUMBER OF INTERVALS OF LENGTH SIG INCREASE WITH TOTAL
C           QUADRATURE LENGTH
            MMQ=INT(SNGL(WSM))+2
            MQ=MIN(MMQ,20)
            SIG=WSM/DBLE(FLOAT(MQ))
            INIT=-MQ
            X1=0.0D0
            REL=TOL
            CALL DQUAD8(DQI9,INIT,X1,SIG,REL,X2,ANS,IERR)
            QANS=ANS
C           INTERVAL (WSM,WBM]
            XL=T-WSM
            MMQ=INT(SNGL(XL))+2
            MQ=MIN(MMQ,20)
            SIG=XL/DBLE(FLOAT(MQ))
C           SKIP QUADRATURE IF INTERVAL IS TOO SMALL FOR AN ACCURATE
C           QUADRATURE. AVOIDS AN ERROR MESSAGE FROM DGAUS8
            XMID=0.5D0*(T+WSM)
            FI2=DQI9(XMID)
            IF(FI2*SIG.LE.2.0D0*REL*QANS) GOTO 51
            INIT=-MQ
            X1=WSM
            REL=TOL
            CALL DQUAD8(DQI9,INIT,X1,SIG,REL,X2,ANS,IERR)
            QANS=QANS+ANS
 51         CONTINUE
          ELSE
C           INTERVAL (0,WSM]
            MMQ=INT(SNGL(WSM))+2
            MQ=MIN(MMQ,20)
            SIG=WSM/DBLE(FLOAT(MQ))
            INIT=-MQ
            X1=0.0D0
            REL=TOL
            CALL DQUAD8(DQI9,INIT,X1,SIG,REL,X2,ANS,IERR)
            QANS=ANS
C           INTERVAL (WSM,WBM]
            XL=WBM-WSM
            MMQ=INT(SNGL(XL))+2
            MQ=MIN(MMQ,20)
            SIG=XL/DBLE(FLOAT(MQ))
            XMID=0.5D0*(WSM+WBM)
            FI2=DQI9(XMID)
            IF(FI2*SIG.LE.2.0D0*REL*QANS) GOTO 52
            INIT=-MQ
            X1=WSM
            REL=TOL
            CALL DQUAD8(DQI9,INIT,X1,SIG,REL,X2,ANS,IERR)
            QANS=QANS+ANS
 52         CONTINUE
C           INTERVAL (WBM,T]
            XL=T-WBM
            MMQ=INT(SNGL(XL))+2
            MQ=MIN(MMQ,20)
            SIG=XL/DBLE(FLOAT(MQ))
```

```
            XMID=0.5D0*(T+WBM)
            FI2=DQI9(XMID)
            IF(FI2*SIG.LE.2.0D0*REL*QANS) GOTO 53
            INIT=-MQ
            X1=WBM
            REL=TOL
            CALL DQUAD8(DQI9,INIT,X1,SIG,REL,X2,ANS,IERR)
            QANS=QANS+ANS
 53         CONTINUE
          ENDIF
        ENDIF
      ELSE
        INIT=0
        X1=T
        A2=A*A
        B2=B*B
        SIG=6.0D0/DSQRT(A2+B2)
        REL=TOL
        CALL DQUAD8(DQI9,INIT,X1,SIG,REL,X2,QANS,IERR)
      ENDIF
      RETURN
      END
      DOUBLE PRECISION FUNCTION DQI9(T)
C
C     DQI9 COMPUTES THE INTEGRAND FOR THE QUADRATURE COMPARISON
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/ CFI9/ CA,CB,KKODE
      ARG=CA*T
      ERFA=DRERF(ARG,KKODE,NZ)
      ARG=CB*T
      ERFB=DRERF(ARG,KKODE,NZ)
      DQI9=T*ERFA*ERFB
      RETURN
      END
      PROGRAM I3COMP
C
C     DONALD E. AMOS    AUGUST, 2000;   MARCH, 2006
C-----------------------------------------------------------------------
C     A DOUBLE PRECISION ROUTINE
C-----------------------------------------------------------------------
C
C     REF: FOLDER 7
C
C     ON KODE=1, I3COMP COMPARES SERIES FROM INTEGI3 AND DGAUS8
C     QUADRATURE EVALUATIONS OF
C
C     I3=INT ON (T,INF) OF EXP(-C^2*W^2)*ERF(A*W)*ERF(B*W).
C
C     ON KODE=2, I3COMP COMPARES SERIES FROM INTEGI3 AND DQUAD8
C     QUADRATURE EVALUATIONS OF
C
C     I3C=INT ON (T,INF) OF EXP(-C^2*W^2)*ERFC(A*W)*ERFC(B*W)
C
C     WHERE C>0 AND PARAMETERS A AND B ARE NON-NEGATIVE.
C
C     OUTPUT IS WRITTEN TO FILE I3COMP.TXT
C----------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      OPEN(UNIT=7,FILE="I3COMP.TXT")
C
      DO 5 KODE=1,2
      WRITE(7,105)KODE
 105  FORMAT('KODE = ',I2)
      WRITE (7,110)
 110  FORMAT('     A         B          C          T         QUAD
     &REL ERR  IERR KFORM')
```

```
      DO 10 IT=1,11,2
        T=DBLE(FLOAT(IT))/4.0D0
        DO 20 IA=1,11,2
          A=DBLE(FLOAT(IA-1))/4.0D0
          DO 30 IB=1,11,2
            B=DBLE(FLOAT(IB-1))/2.0D0
            DO 40 IC=1,11,2
              C=DBLE(FLOAT(IC))/2.0D0
              CALL INTEGI3(A,B,C,T,KODE,ANS3,I3ERR,KFORM)
              IF((I3ERR.NE.0)) THEN
                WRITE (7,405) A,B,C,T,I3ERR,KFORM
 405            FORMAT( 4D11.4,1X,'IN INTEGI3, I3ERR=',I1,' KFORM=',I1)
                GOTO 40
              ENDIF
C-----------------------------------------------------------------------
C     KFORM IS A RETURN VARIABLE TELLING WHICH FORMULA WAS USED IN THE
C     COMPUTATION OF ANSI3 WHEN IERR=0:
C
C     KFORM=0 MEANS EITHER A OR B WAS ZERO. ANSI3=0.0D0 RETURNED
C     KFORM=1 MEANS THE ENHANCEMENT (A)- ERF(*)=1, BOTH AT AND BT.GE.6
C     KFORM=2 MEANS THE ENHANCEMENT (B)- POWER SERIES AT,BT,CT.LE.2.0D0
C     KFORM=3 MEANS THE ENHANCEMENT (C)- LARGE C FORMULA
C     KFORM=4,5 MEANS MAIN FORMULA CASE I  FOR C.LE.XA.LE.XB
C     KFORM=6,7 MEANS MAIN FORMULA CASE I  FOR XA.LE.C.LE.XB
C     KFORM=8   MEANS MAIN FORMULA CASE II FOR XA.LE.XB.LE.C
C               WHERE XA=MIN(A,B) AND XB=MAX(A,B)
C------------------------------------------------------------------------
            CALL I3QUAD(A,B,C,T,KODE,QANS,IERR)
            ERR=DABS(ANS3-QANS)
            IF (QANS.NE.0.0D0) THEN
              ERR=ERR/DABS(QANS)
            ENDIF
            WRITE (7,200) A,B,C,T,QANS,ERR,IERR,KFORM
 200        FORMAT(4D11.4,D12.5,D12.4,2I4)
   40       CONTINUE
   30     CONTINUE
   20   CONTINUE
   10 CONTINUE
   5  CONTINUE
      END
      SUBROUTINE I3QUAD(A,B,C,T,KODE,QANSI3,IERR)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON /CFI3/ AA,BB,CC,C2,KKODE
      DATA RTPI       /1.772453850905516D0/
      EXTERNAL DQI3
      IERR=0
      AA=A
      BB=B
      CC=C
      C2=C*C
      KKODE=KODE
      IF(KODE.EQ.1) THEN
        JD=50
        XA=MIN(A,B)
        QANS=0.0D0
        IF(XA.EQ.0.0D0) THEN
           QANSI3=0.0D0
           RETURN
        ENDIF
C       SEE FOLDER 7d FOR THE METHOD OF INTEGRATION
        WSM=MIN(6.0D0/A,6.0D0/B)
        WBM=MAX(6.0D0/A,6.0D0/B)
        IF(T.LE.WSM) THEN
          SIGC=3.0D0/C
          SIG=MIN(SIGC,0.455D0*WSM)
          IFLAG=0
          X2=T
```

```
          DO 50 J=1,JD
            X1=X2
            X2=X1+SIG
            REL=0.50D-14
            IF(X2.GE.WSM) THEN
              X2=WSM
              IFLAG=1
            ENDIF
            CALL DGAUS8(DQI3,X1,X2,REL,ANS,IERRQ)
            QANS=QANS+ANS
            IF(IERRQ.EQ.1)THEN
              IERR=0
            ELSE
              IERR=IERRQ
            ENDIF
            IF(DABS(ANS).LE.REL*DABS(QANS)) THEN
              QANSI3=QANS
              RETURN
            ENDIF
            IF(IFLAG.EQ.1) GOTO 55
 50       CONTINUE
          WRITE (7,300) JD
 300      FORMAT('NO CONVERGENCE FROM DGAUS8 LOOP IN I3COMP AFTER',I3,
     &' STEPS')
  55      CONTINUE
          CALL INTEGJ5(C,XA,WSM,ANS1,IERR5)
          CALL INTEGI5(C,XA,WBM,ANS2,IERR5)
          SS=ANS1+ANS2
          QANSI3=QANS+SS
        ELSE
          IF(T.LE.WBM) THEN
            CALL INTEGJ5(C,XA,T,ANS1,IERR5)
            CALL INTEGI5(C,XA,WBM,ANS2,IERR5)
            SS=ANS1+ANS2
            QANSI3=SS
          ELSE
            IKODE=2
            CT=C*T
            ERFC=DRERF(CT,IKODE,NZ)
            QANSI3=0.5D0*RTPI*ERFC/C
          ENDIF
        ENDIF
      ELSE
        INIT=0
        X1=T
        SIG=3.0D0/DSQRT(A*A+B*B+C*C)
        REL=0.5D-14
        CALL DQUAD8(DQI3,INIT,X1,SIG,REL,X2,QANS,IERR)
        QANSI3=QANS
      ENDIF
      RETURN
      END
      DOUBLE PRECISION FUNCTION DQI3(T)
C
C     DQI3 COMPUTES THE INTEGRAND FOR THE QUADRATURE COMPARISON
C     OF I3 OR I3C IN I3COMP
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/ CFI3/ AA,BB,CC,C2,KKODE
      ARG=BB*T
      NZ=0
      ERFB=DRERF(ARG,KKODE,NZ)
      ARG=AA*T
      ERFA=DRERF(ARG,KKODE,NZ)
      T2=T*T
      ARG=C2*T2
      DQI3=ERFA*ERFB*DEXP(-ARG)
```

```
      RETURN
      END
      PROGRAM I5COMP
C
C     DONALD E. AMOS    AUGUST, 2000
C-----------------------------------------------------------------------
C     A DOUBLE PRECISION ROUTINE
C-----------------------------------------------------------------------

C     REF: FOLDER 5
C
C     I5COMP COMPARES THE SERIES FROM INTEGI5 AND DGAUS8 QUADRATURE
C     EVALUATIONS OF
C
C            I5=INT ON (X,INF) OF  EXP(-A^2*T^2)*ERFC(B*T)
C
C     WHERE ALL PARAMETERS ARE NONNEGATIVE WITH A AND B NOT ZERO AT THE
C     SAME TIME.
C
C     OUTPUT IS WRITTEN TO FILE I5COMP.TXT
C-----------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON /CFI5/ AA,BB
      EXTERNAL DQI5
      OPEN(UNIT=7,FILE="I5COMP.TXT")
      WRITE (7,110)
 110  FORMAT('      A           B           X         SERIES        QUAD
     &        REL ERR IERR')
      DO 10 IX=1,20,2
        X=DBLE(FLOAT(IX-1))/1.00D0
        DO 20 IA=1,11,2
          A=DBLE(FLOAT(IA-1))/10.0D0
          DO 30 IB=1,11,2
            B=DBLE(FLOAT(IB-1))/5.0D0
            CALL INTEGI5(A,B,X,ANS5,I5ERR)
            IF ((I5ERR.NE.0)) THEN
              WRITE (7,405) A,B,X,I5ERR
 405          FORMAT(3D12.4,'  FROM INTEGI5, ERROR FLAG = ',I3)
              GOTO 30
            ENDIF
            X1=X
            SIG=1.5D0/MAX(A,B)
            AA=A*A
            BB=B
            REL=1.0D-14
            INIT=0
            CALL DQUAD8(DQI5,INIT,X1,SIG,REL,X2,QANS,IERR)
            ERR=DABS(ANS5-QANS)
            IF (QANS.NE.0.0D0) THEN
              ERR=ERR/DABS(QANS)
            ENDIF
            WRITE (7,200) A,B,X,ANS5,QANS,ERR,IERR
  200       FORMAT(3D12.4,2D13.5,D12.4,I3)
   30     CONTINUE
   20   CONTINUE
   10 CONTINUE
      END
      DOUBLE PRECISION FUNCTION DQI5(T)
C
C     DQI5 COMPUTES THE INTEGRAND FOR THE QUADRATURE COMPARISON
C     OF I5 IN I5COMP
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/ CFI5/ AA,BB
      ARG=BB*T
      KODE=2
      NZ=0
```

```
      ERF2=DRERF(ARG,KODE,NZ)
      ARG=-AA*T*T
      DQI5=ERF2*DEXP(ARG)
      RETURN
      END
      PROGRAM J5COMP
C
C     DONALD E. AMOS  AUGUST, 2000
C-----------------------------------------------------------------------
C     A DOUBLE PRECISION ROUTINE
C-----------------------------------------------------------------------
C
C     REF: FOLDER 5
C
C     J5COMP COMPARES THE SERIES FROM INTEGJ5 AND DGAUS8 QUADRATURE
C     EVALUATIONS OF
C
C        INT ON (X,INF) OF  EXP(-A*A*T*T)*ERF(B*T)
C
C     WHERE A.GT.0.0D0, B.GE.0.0D0 AND T.GE.0.0D0
C
C     OUTPUT IS WRITTEN TO FILE J5COMP.TXT
C-----------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON /CFJ5/ AA,BB
      EXTERNAL DQJ5
      OPEN(UNIT=7,FILE="J5COMP.TXT")
      WRITE (7,110)
 110  FORMAT('      A           B           X         SERIES        QUAD
     &        REL ERR JERR')
      DO 10 IX=1,10
        X=DBLE(FLOAT(IX-1))/1.0D0
        DO 20 IA=1,21,2
          A=DBLE(FLOAT(IA))/5.0D0
          DO 30 IB=1,21,2
            B=DBLE(FLOAT(IB))/5.0D0
            CALL INTEGJ5(A,B,X,ANS5,JERR)
            JD=30
            X2=X
            SIG=3.0D0/MAX(A,B)
            AA=A*A
            BB=B
            QANS=0.0D0
            REL=1.0D-14
            IFLAG=0
            DO 50 J=1,JD
              X1=X2
              X2=X2+SIG
              CALL DGAUS8(DQJ5,X1,X2,REL,ANS,IERR)
              QANS=QANS+ANS
              ERR=DABS(ANS)
              IF (ERR.LE.REL*DABS(QANS)) GOTO 55
              IF (IFLAG.EQ.0) THEN
                IF(B*X2.GT.3.0D0) THEN
                  SIG=3.0D0/A
                  IFLAG=1
                ENDIF
              ENDIF
  50        CONTINUE
            WRITE (7,300)
 300        FORMAT('NO ANSWER FROM DGAUS8 AFTER 30 STEPS')
  55        CONTINUE
            IF (QANS.NE.0.0D0) THEN
              ERR=DABS(QANS-ANS5)/DABS(QANS)
            ENDIF
            WRITE (7,200) A,B,X,ANS5,QANS,ERR,JERR
 200        FORMAT(3D12.4,2D13.5,D12.4,I3)
```

```
   30     CONTINUE
   20   CONTINUE
   10 CONTINUE
      END
      DOUBLE PRECISION FUNCTION DQJ5(T)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/ CFJ5/ AA,BB
      ARG=BB*T
      KODE=1
      NZ=0
      ERF2=DRERF(ARG,KODE,NZ)
      ARG=-AA*T*T
      DQJ5=ERF2*DEXP(ARG)
      RETURN
      END
      PROGRAM V5COMP
C
C     DONALD E. AMOS      JANUARY, 2003; APRIL 2006
C
C     REF: FOLDER 5
C
C     V5COMP COMPARES THE SERIES FROM INTEGV5 AND DGAUS8 QUADRATURE
C     EVALUATIONS OF
C
C            V5 = INT ON (0,X) OF  EXP(-A^2*W^2)*ERF(B*W)
C
C     WHERE A.GE.0 , B.GE.0, X.GE.0
C
C     OUTPUT IS WRITTEN TO FILE V5COMP.TXT
C-----------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON /CFV5/ AA,BB
      OPEN(UNIT=7,FILE="V5COMP.TXT")
      WRITE (7,110)
 110  FORMAT('      A           B           X           QUAD        REL E
     &RR   IERR IERR5')
      DO 10 IX=1,41,2
        X=DBLE(FLOAT(IX-1))/5.0D0
        DO 20 IA=1,81,8
          A=DBLE(FLOAT(IA-1))/10.0D0
          DO 30 IB=1,81,8
            B=DBLE(FLOAT(IB-1))/2.0D0
            CALL INTEGV5(A,B,X,ANS5,IERR5)
            CALL V5QUAD(A,B,X,QANS,IERR)
            ERR=DABS(ANS5-QANS)
            IF (QANS.NE.0.0D0) THEN
              ERR=ERR/DABS(QANS)
            ENDIF
            WRITE (7,200) A,B,X,QANS,ERR,IERR,IERR5
 200        FORMAT(3D12.4,D13.5,D12.4,2I5)
   30     CONTINUE
   20   CONTINUE
   10 CONTINUE
      END
      SUBROUTINE V5QUAD(A,B,T,QANS,IERR)
C     V5 BY QUADRATURE
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION MQ(2),XX(2),SS(2)
      COMMON/ CFV5/ AA,BB
      DATA TOL/0.5D-14/
      EXTERNAL DQV5
      IERR=0
      IF(B.EQ.0.0D0) THEN
        QANS=0.0D0
        IERR=1
        RETURN
      ENDIF
```

```
C     SET COMMON VARIABLES
      AA=A*A
      BB=B
      ABM=MAX(A,B)
      IF(ABM.NE.0.0D0) THEN
        SIGM=6.0D0/ABM
C       IF SIGM IS SMALL, MOST OF THE AREA COULD BE NEAR T=0 OR A LARGE
C       GRADIENT COULD OCCUR AND A REFINEMENT NEAR T=0 MAKES
C       THE COMPUTATION BETTER OVER A WIDER RANGE OF VARIABLES.
        IF(SIGM.LT.0.25D0*T) THEN
          JD=2
          MMQ=2
          MQ(1)=MMQ
          XX(1)=0.0D0
          SS(1)=SIGM/DBLE(FLOAT(MMQ))
          TL=T-SIGM
          MMQ=INT(SNGL(TL/2.0D0))+2
          MQ(2)=MMQ
          XX(2)=SIGM
          SS(2)=TL/DBLE(FLOAT(MMQ))
        ELSE
          JD=1
          MMQ=INT(SNGL(T/2.0D0))+2
          MQ(1)=MMQ
          XX(1)=0.0D0
          SS(1)=T/DBLE(FLOAT(MMQ))
        ENDIF
      ELSE
C       THE INTEGRAND IS ZERO HERE FOR MAX(A,B)=0.0D0
        QANS=0.0D0
        IERR=1
        RETURN
      ENDIF
      QANS=0.0D0
      DO 10 J=1,JD
        INIT=-MQ(J)
        X1=XX(J)
        SIG=SS(J)
        REL=TOL
        CALL DQUAD8(DQV5,INIT,X1,SIG,REL,X2,ANS,IERR)
        QANS=QANS+ANS
 10   CONTINUE
      RETURN
      END
      DOUBLE PRECISION FUNCTION DQV5(T)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/ CFV5/ AA,BB
      ARG=BB*T
      KODE=1
      NZ=0
      ERF2=DRERF(ARG,KODE,NZ)
      ARG=-AA*T*T
      DQV5=ERF2*DEXP(ARG)
      RETURN
      END
      PROGRAM I6COMP
C
C     DONALD E. AMOS    DECEMBER, 2001; FEBRUARY, 2006
C-----------------------------------------------------------------------
C     A DOUBLE PRECISION ROUTINE
C-----------------------------------------------------------------------
C
C     ON KODE=1, I6COMP COMPUTES THE INTEGRAL
C
C            INT ON (T,INF) OF ERF(A*W)*ERF(B*W)/W**2,
C
C     ON KODE=2, I6COMP COMPUTES THE INTEGRAL
```

```
C
C            INT ON (T,INF) OF ERFC(A*W)*ERFC(B*W)/W**2
C
C     FOR A, B, T NON-NEGATIVE. T=0 IS NOT PERMITTED FOR KODE=2.
C
C     BY QUADRATURE (I6QUAD) AND BY SERIES (INTEGI6) AND PRINTS
C     THE RELATIVE ERROR. FORMULAE ARE CONTAINED IN FOLDER 3 FOR
C     THE (REFINED) QUADRATURE PROCEDURE AND FOLDERS 6 AND 15 FOR THE
C     SERIES.
C
C     OUTPUT IS WRITTEN TO FILE I6COMP.TXT
C-----------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      EXTERNAL FINTI6
      OPEN(UNIT=7,FILE="I6COMP.TXT")
      DO 5 KODE=1,2
        WRITE(7,201) KODE
 201    FORMAT('                           KODE = ', I2)
        WRITE(7,200)
 200    FORMAT('     A          B          T         QUAD       SERIES
     &    REL ERR   IERR')
        DO 10 IA=6,41,5
          A=DBLE(FLOAT(IA-1))/10.0D0
C          A=0.001D0+0.05D0*DBLE(FLOAT(IA-1))
C          A=100.0D0/10.0D0**(IA-1)
          DO 20 IB=6,41,5
            B=DBLE(FLOAT(IB-1))/10.0D0
C         B=0.001D0+0.05D0*DBLE(FLOAT(IB-1))
C         B=100.0D0/10.0D0**(IB-1)
            DO 30 IT=1,26
              T=0.0D0+0.2D0*DBLE(FLOAT(IT-1))
              CALL INTEGI6(A,B,T,KODE,SS,JERR)
              REL=0.5D-14
              CALL I6QUAD(A,B,T,REL,KODE,ANS,IERR)
              ERR=DABS(SS-ANS)
              IF(ANS.NE.0.0D0) THEN
                ERR=ERR/DABS(ANS)
              ENDIF
              WRITE(7,100) A,B,T,ANS,SS,ERR,JERR
  100         FORMAT(3D11.4,3D12.5,I3)
  30        CONTINUE
  20      CONTINUE
  10    CONTINUE
  5   CONTINUE
      END
      SUBROUTINE I6QUAD(A,B,X,REL,KODE,ANS,IERR)
C
C     REF.: REFINED QUADRATURE OF FOLDER 3
C-----------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      EXTERNAL FINTI6
      COMMON /CFINTI6/AA,BB,KKODE
      AA=A
      BB=B
      KKODE=KODE
      IF(KODE.EQ.1) THEN
        XAB=MIN(A,B)
        WSM=MIN(6.0D0/A,6.0D0/B)
        WLM=MAX(6.0D0/A,6.0D0/B)
        JL=1
        SS=0.0D0
        IF (X.LE.WSM) THEN
          W2=X
          JL=5
          DW=(WSM-X)/DBLE(FLOAT(JL))
          DO 42 L=1,JL
            W1=W2
```

```
            W2=W1+DW
            TOL=REL
            CALL DGAUS8(FINTI6,W1,W2,TOL,QANS,IERR)
            SS=SS+QANS
   42     CONTINUE
          SS=SS+QPI6(XAB,WSM,WLM,REL)
        ELSE
          IF (X.LE.WLM)  THEN
            SS=QPI6(XAB,X,WLM,REL)
          ELSE
            SS=1.0D0/X
          ENDIF
        ENDIF
        ANS=SS
        RETURN
      ELSE
        IF(X.EQ.0.0D0) THEN
          ANS=0.0D0
          IERR=2
          RETURN
        ENDIF
        INIT=0
        X1=X
        SIG=3.0D0/DSQRT(A*A+B*B)
        TOL=REL
        CALL DQUAD8(FINTI6,INIT,X1,SIG,TOL,X2,QANS,IERR)
        CALL DQUAD8(FINTI6,INIT,X1,SIG,TOL,X2,QANS,IERR)
        ANS=QANS
        RETURN
      ENDIF
      END
      DOUBLE PRECISION FUNCTION FINTI6(W)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON /CFINTI6/AA,BB,KKODE
      ARG=AA*W
      ERF1=DRERF(ARG,KKODE,NZ)/W
      ARG=BB*W
      ERF2=DRERF(ARG,KKODE,NZ)/W
      FINTI6=ERF1*ERF2
      RETURN
      END
      PROGRAM W3COMP
C
C     DONALD E. AMOS   OCTOBER, 2001; MARCH, 2003
C
C     REF: FOLDER1, FOLDER2, FOLDER 10
C
C     W3COMP COMPARES THE CLOSED FORM OF W3 OF FOLDER 10 WITH A
C     REFINED QUADRATURE FOR W3 IN THE MANNER OF FOLDER 3. ON KODE=1,
C
C          W3=INT ON (T,INF) OF ERF(A*W)*ERF(B*W)/W**3.
C
C     ON KODE=2, THE COMPLEMENTARY FUNCTION IS COMPUTED
C
C          W3C=INT ON (T,INF) OF ERFC(A*W)*ERFC(B*W)/W**3
C
C     WHERE A.GE.0.0D0, B.GE.0.0D0, AND T.GT.0.0D0. THE CLOSED FORMS ARE
C     IMPLEMENTED IN DOUBLE PRECISION SUBROUTINE INTEGW3.
C
C     OUTPUT IS WRITTEN TO FILE W3COMP.TXT
C-----------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      OPEN(UNIT=7,FILE="W3COMP.TXT")
      WRITE (7,110)
 110  FORMAT('       A          B           T         QUAD          ERR'
     *)
      KODE=1
```

```
      WRITE(7,115) KODE
 115  FORMAT(/,2X,'THE FOLLOWING COMPARISONS ARE FOR KODE = ',I3/)
      DO 10 IT=1,111,10
        T=DBLE(FLOAT(IT))/25.0D0
        DO 20 IA=1,111,10
          A=DBLE(FLOAT(IA-1))/25.0D0
          DO 30 IB=1,111,10
            B=DBLE(FLOAT(IB-1))/25.0D0
            REL=1.0D-14
            CALL INTEGW3(A,B,T,KODE,REL,ANS1,IERR2)
            REL=1.0D-14
            IF(KODE.EQ.1) THEN
              CALL W3BYQ1(A,B,T,REL,QANS,IERR)
            ELSE
              CALL W3BYQ2(A,B,T,REL,QANS,IERR)
            ENDIF
  40        CONTINUE
            ERR=DABS(QANS-ANS1)
            IF(QANS.NE.0.0D0) THEN
              ERR=ERR/DABS(QANS)
            ENDIF
            WRITE(7,100) A,B,T,QANS,ERR
 100        FORMAT(3D12.4,D13.5,D12.4)
 101        FORMAT(3I5)
   30     CONTINUE
   20   CONTINUE
   10 CONTINUE
      END
      SUBROUTINE W3BYQ1(A,B,X,REL,ANS,IERR)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON /CQFW3/AA,BB,KKODE
      EXTERNAL DQFW3
C     REFINED QUADRATURE OF FOLDER 3
      IF((A.EQ.0.0D0).OR.(B.EQ.0.0D0)) THEN
        ANS=0.0D0
        RETURN
      ENDIF
      AA=A
      BB=B
      KKODE=1
      XAB=MIN(A,B)
      WSM=MIN(6.0D0/A,6.0D0/B)
      WLM=MAX(6.0D0/A,6.0D0/B)
      JL=1
      SS=0.0D0
      IF (X.LE.WSM) THEN
        W2=X
        JL=5
        DW=(WSM-X)/DBLE(FLOAT(JL))
        DO 42 L=1,JL
          W1=W2
          W2=W1+DW
          CALL DGAUS8(DQFW3,W1,W2,REL,QANS,IERR)
          SS=SS+QANS
   42   CONTINUE
        SS=SS+PW3(XAB,WSM,WLM)+0.50D0/(WLM*WLM)
      ELSE
        IF (X.LE.WLM)  THEN
          SS=PW3(XAB,X,WLM)+0.50D0/(WLM*WLM)
        ELSE
          SS=0.50D0/(X*X)
        ENDIF
      ENDIF
      ANS=SS
      RETURN
      END
      DOUBLE PRECISION FUNCTION PW3(XAB,FL,FU)
```

```
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      T1=XAB*FL
      T2=XAB*FU
      KODE=1
      NZ=0
      TRM1=DRERF(T1,KODE,NZ)/FL
      KODE=1
      NZ=0
      TRM2=DRERF(T2,KODE,NZ)/FU
      S=0.5D0*(TRM1/FL-TRM2/FU)
      KODE=1
      CALL DIERFC(T1,KODE,ANSL,IERR)
      CALL DIERFC(T2,KODE,ANSU,IERR)
      PW3=S+(ANSL/FL-ANSU/FU)*XAB
      RETURN
      END
      SUBROUTINE W3BYQ2(A,B,X,REL,ANS,IERR)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON /CQFW3/AA,BB,KKODE
      EXTERNAL DQFW3
C     REFINED QUADRATURE OF FOLDER 3
      IF((A.EQ.0.0D0).AND.(B.EQ.0.0D0)) THEN
        ANS=0.50D0/(X*X)
        RETURN
      ENDIF
      KODE=2
      AS=A*A
      BS=B*B
C     SET COMMON PARAMETERS
      AA=A
      BB=B
      KKODE=KODE
      INIT=0
      X1=X
      SIG=2.0D0/DSQRT(AS+BS)
      TOL=REL
      CALL DQUAD8(DQFW3,INIT,X1,SIG,TOL,X2,QANS,IERR)
      ANS=QANS
      RETURN
      END
      DOUBLE PRECISION FUNCTION DQFW3(W)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON /CQFW3/AA,BB,KKODE
      ARG=AA*W
      ERF1=DRERF(ARG,KKODE,NZ)/W
      ARG=BB*W
      ERF2=DRERF(ARG,KKODE,NZ)/W
      DQFW3=ERF1*ERF2/W
      RETURN
      END
      PROGRAM PCOMP
C
C     DONALD E. AMOS    AUGUST, 2000;  JANUARY, 2006
C-----------------------------------------------------------------------
C     A DOUBLE PRECISION ROUTINE
C-----------------------------------------------------------------------
C
C     REF: FOLDER 11
C
C     PCOMP COMPUTES THE INTEGRAL
C
C       P  =  INT ON (T,INF) OF EXP(-(A*W)^2)*ERF(B*W)/W   ON KODE=1
C
C     AND THE COMPLEMENTARY INTEGRAL
C
C       PC =  INT ON (T,INF) OF EXP(-(A*W)^2)*ERFC(B*W)/W  ON KODE=2
C
```

```
C     BY QUADRATURE AND BY SERIES AND PRINTS THE RELATIVE ERROR.
C     OUTPUT IS WRITTEN TO FILE PCOMP.TXT
C-----------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      OPEN(UNIT=7,FILE="PCOMP.TXT")
C
      DO 5 KODE=1,2
      WRITE(7,201) KODE
 201  FORMAT(/,32X,'  KODE = ',I3)
      WRITE(7,200)
 200  FORMAT('     A          B          T         QUAD       SERIES
     C  REL ERR JERR IERR')
      DO 10 IA=1,31,5
         A=DBLE(FLOAT(IA-1))/10.0D0
        DO 20 IB=1,31,5
          B=DBLE(FLOAT(IB-1))/10.0D0
          DO 30 IT=1,31,2
            T=0.0D0+0.10D0*DBLE(FLOAT(IT-1))
            REL=0.5D-14
            CALL INTEGP(A,B,T,KODE,REL,SS,JERR)
            IF((JERR.NE.0).AND.(JERR.NE.2)) THEN
              SS=0.0D0
              QANS=0.0D0
              IERR=8
              GOTO 40
            ENDIF
            CALL PQUAD(A,B,T,REL,KODE,QANS,IERR)
  40        CONTINUE
            ERR=DABS(SS-QANS)
            IF(QANS.NE.0.0D0)THEN
              ERR=ERR/QANS
            ENDIF
            WRITE(7,100) A,B,T,QANS,SS,ERR,JERR,IERR
  100       FORMAT(3D11.4,3D12.5,2I3)
  30      CONTINUE
  20    CONTINUE
  10  CONTINUE
  5   CONTINUE
      END
      SUBROUTINE PQUAD(A,B,T,REL,KODE,ANSP,IERR)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON /CFINTP/AA,BB,KKODE
      EXTERNAL FINTP
      IERR=1
      AA=A
      BB=B
      KKODE=KODE
      IF(KODE.EQ.1) THEN
        IF(A.EQ.0.0D0) THEN
          ANSP=0.0D0
          IERR=6
          RETURN
        ENDIF
        IF(B.EQ.0.0D0) THEN
          ANSP=0.0D0
          RETURN
        ENDIF
        WSM=6.0D0/B
        X2=T
        QANS=0.0D0
        DO 10 K=1,50
          X1=X2
          IF(X1.LT.WSM) THEN
            SIG=0.25D0*WSM
          ELSE
            SIG=3.0D0/A
          ENDIF
```

```
          X2=X1+SIG
          TOL=REL
          CALL DGAUS8(FINTP,X1,X2,TOL,ANS,IERR)
          QANS=QANS+ANS
          IF(DABS(ANS).LT.REL*DABS(QANS)) GOTO 15
   10   CONTINUE
   15   CONTINUE
        ANSP=QANS
      ELSE
        IF((A.EQ.0.0D0).AND.(B.EQ.0.0D0)) THEN
          ANSP=0.0D0
          IERR=7
          RETURN
        ENDIF
        IF(T.EQ.0.0D0) THEN
          IERR=9
          ANSP=0.0D0
          RETURN
        ENDIF
        A2=A*A
        B2=B*B
        INIT=0
        X1=T
        SIG=6.0D0/DSQRT(A2+B2)
        PREL=REL
        CALL DQUAD8(FINTP,INIT,X1,SIG,PREL,X2,QANS,IERR)
        ANSP=QANS
      ENDIF
      RETURN
      END
      DOUBLE PRECISION FUNCTION FINTP(W)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON /CFINTP/AA,BB,KKODE
      ARG=BB*W
      ERFB=DRERF(ARG,KKODE,NZ)
      ARG=AA*W
      ARG2=ARG*ARG
      FINTP=DEXP(-ARG2)*ERFB/W
      RETURN
      END
      PROGRAM QCOMP
C
C     DONALD E. AMOS    AUGUST, 2000; JANUARY, 2006
C-----------------------------------------------------------------------
C     A DOUBLE PRECISION ROUTINE
C-----------------------------------------------------------------------
C
C     REF. FOLDER 11, FOLDER 6
C
C     QCOMP COMPUTES THE INTEGRAL
C
C       Q =  INT ON (T,INF) OF EXP(-(A*W)^2)*E(1,(B*W)^2)
C
C     BY QUADRATURE AND BY SERIES AND PRINTS THE RELATIVE ERROR.
C     OUTPUT IS WRITTEN TO FILE QCOMP.TXT.
C-----------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      OPEN(UNIT=7,FILE="QCOMP.TXT")
      WRITE(7,200)
  200 FORMAT('     A           B           T          QUAD       SERIES
     C  REL ERR JERR IERR')
      DO 10 IA=1,31,5
        A=DBLE(FLOAT(IA-1))/10.0D0
        DO 20 IB=1,31,5
          B=DBLE(FLOAT(IB-1))/10.0D0
          DO 30 IT=1,40,2
            T=0.0D0+0.10D0*DBLE(FLOAT(IT))
```

```
          REL=0.5D-14
          CALL INTEGQ(A,B,T,REL,SS,JERR)
          IF((JERR.NE.0).AND.(JERR.NE.2)) THEN
            SS=0.0D0
            QANS=0.0D0
            IERR=8
            GOTO 40
          ENDIF
          CALL QQUAD(A,B,T,REL,QANS,IERR)
 40       CONTINUE
          IF(IERR.NE.1) THEN
            QANS=0.0D0
          ENDIF
          ERR=DABS(SS-QANS)
          IF(QANS.NE.0.0D0)THEN
            ERR=ERR/QANS
          ENDIF
          WRITE(7,100) A,B,T,QANS,SS,ERR,JERR,IERR
 100      FORMAT(3D11.4,3D12.5,2I3)
 30     CONTINUE
 20   CONTINUE
 10 CONTINUE
    END
    SUBROUTINE QQUAD(A,B,T,REL,ANSQ,IERR)
    IMPLICIT DOUBLE PRECISION (A-H,O-Z)
    COMMON /CFINTQ/AA,BB,CREL
    EXTERNAL FINTQ
    AA=A
    BB=B
    CREL=REL
    A2=A*A
    B2=B*B
    INIT=0
    X1=T
    SIG=4.0D0/DSQRT(A2+B2)
    QREL=REL
    CALL DQUAD8(FINTQ,INIT,X1,SIG,QREL,X2,QANS,IERR)
    SIG=SIG+SIG
    CALL DQUAD8(FINTQ,INIT,X1,SIG,QREL,X2,QANS,IERR)
    ANSQ=QANS
    RETURN
    END
    DOUBLE PRECISION FUNCTION FINTQ(W)
    IMPLICIT DOUBLE PRECISION (A-H,O-Z)
    DIMENSION EN(1)
    COMMON /CFINTQ/AA,BB,CREL
    ARG=BB*W
    ARG2=ARG*ARG
    N=1
    KKODE=1
    M=1
    TOL=CREL
    CALL DEXINT(ARG2, N, KKODE, M, TOL, EN, NZ, IERR)
    ARG=AA*W
    ARG2=ARG*ARG
    FINTQ=DEXP(-ARG2)*EN(1)
    RETURN
    END
    PROGRAM I21COMP
C
C   DONALD E. AMOS   OCTOBER, 2002; MAY, 2006
C
C   REF: FOLDER 21
C
C   ON KODE=1, I21COMP COMPARES THE PROCEDURE FROM SUBROUTINE INTEGI21
C   AND DGAUS8 QUADRATURE EVALUATIONS OF
C
```

```
C          I21=INT ON (0,T) OF EXP(A^2*W+2*A*B)*ERFC(X)*ERF(C/RW)
C
C     ON KODE=2, I21COMP COMPARES THE PROCEDURE FROM SUBROUTINE INTEGI21
C     AND DGAUS8 QUADRATURE EVALUATIONS OF THE COMPLEMENTARY FUNCTION
C
C          I21=INT ON (0,T) OF EXP(A^2*W+2*A*B)*ERFC(X)*ERFC(C/RW)
C              WITH X=A*RW+B/RW AND RW=SQRT(W)
C
C     AND A, B AND C ARE POSITIVE.
C
C     OUTPUT IS WRITTEN TO FILE I21COMP.TXT
C-----------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      OPEN(UNIT=7,FILE="I21COMP.TXT")
      WRITE (7,109)
 109  FORMAT('-----------------------------------------------------------
     &-----------------'/
     &'LEGENDS FOR KFORM:'//
     &'KFORM DIGIT=0 MEANS ONE OR MORE PARMETERS ARE ZERO--A SPECIAL FOR
     &MULA IS USED'/
     &'KFORM DIGIT=1 MEANS T.LE.(C*C/36) WHERE ERF(C/DSQRT(T))=1.0D0'/
     &'KFORM DIGIT=2 MEANS THE CLOSED FORM IS USED FOR VOFT'/
     &'KFORM DIGIT=3 MEANS THE LARGE L.GE.2 SERIES IS USED FOR W3'/
     &'KFORM DIGIT=4 MEANS THE A*SQRT(T).LT.1 SERIES IS USED FOR I21'/
     &'KFORM DIGIT=5 MEANS THE THE QUADRATURE IS USED FOR W3'//
     &'EXAMPLE:'/
     &'KFORM=12 MEANS A COMBINATION OF 1 AND 2 WAS USED'/
     &'-----------------------------------------------------------------
     &----------')
C
      DO 5 KODE=1,2
C
      WRITE (7,105) KODE
 105  FORMAT(/'THE FOLLOWING TABLE OF COMPARISONS IS FOR KODE = ',I3/)
      WRITE (7,110)
 110  FORMAT('      A         B         C          T         QUAD
     &REL ERR  IERR KFORM')
      DO 10 IT=1,11,2
        T=DBLE(FLOAT(IT))/10.0D0
        DO 20 IA=1,11,2
          A=DBLE(FLOAT(IA-1))/5.0D0
          DO 30 IB=1,11,2
            B=DBLE(FLOAT(IB-1))/2.0D0
            DO 40 IC=1,11,2
              C=DBLE(FLOAT(IC-1))/5.0D0
              CALL INTEGI21(A,B,C,T,KODE,ANS2,I2ERR,KFORM)
              IF((I2ERR.NE.0)) THEN
                WRITE (7,405) A,B,C,T,I2ERR,KFORM
 405            FORMAT( 4D11.4,1X,'IN INTEGI21, I2ERR=',I2,' KFORM=',I2)
                GOTO 40
              ENDIF
C-----------------------------------------------------------------------
C     LEGENDS FOR KFORM:
C
C     KFORM DIGIT=1 MEANS T.LE.(C*C/36) WHERE ERF(C/DSQRT(T))=1.0D0
C     KFORM DIGIT=2 MEANS THE CLOSED FORM IS USED FOR VOFT
C     KFORM DIGIT=3 MEANS THE LARGE L.GE.2 SERIES IS USED FOR W3
C     KFORM DIGIT=4 MEANS THE A*SQRT(T).LT.1 SERIES IS USED FOR I21
C     KFORM DIGIT=5 MEANS THE THE QUADRATURE IS USED FOR W3
C
C     EXAMPLE:
C     KFORM=12 MEANS A COMBINATION OF 1 AND 2 WAS USED
C-----------------------------------------------------------------------
              CALL I21QUAD(A,B,C,T,KODE,QANS,IERR)
              ERR=DABS(ANS2-QANS)
              IF (QANS.NE.0.0D0) THEN
                ERR=ERR/DABS(QANS)
```

```
                ENDIF
                WRITE (7,200) A,B,C,T,QANS,ERR,IERR,KFORM
 200            FORMAT(4D11.4,D12.5,D12.4,2I4)
   40         CONTINUE
   30       CONTINUE
   20     CONTINUE
   10 CONTINUE
   5  CONTINUE
      END
      DOUBLE PRECISION FUNCTION DQI21(X)
C
C     DQI21 COMPUTES THE INTEGRAND FOR THE QUADRATURE COMPARISON
C     OF I21 WITH INTEGI21
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/ CFI21/ CA,CB,CC,IKODE
C     A CHANGE OF VARIABLE T=X*X REMOVES THE SQRT FROM THE INTEGRAND
C     THE UPPER LIMIT ON THE INTEGRAL IS THEN DSQRT(T)
      ARG=CA*X+CB/X
      KODE=3
      NZ=0
      ERFX=DRERF(ARG,KODE,NZ)
      X2=X*X
      ARG=CB*CB/X2
      ERFX=ERFX*DEXP(-ARG)
      ARG=CC/X
      NZ=0
      ERFA=DRERF(ARG,IKODE,NZ)
      DQI21=2.0D0*X*ERFX*ERFA
      RETURN
      END
      SUBROUTINE I21QUAD(A,B,C,T,KODE,QANS21,IERR)
C     QUADRATURE FOR THE I21 INTEGRAL. A CHANGE OF VARIABLES TAU=X*X
C     IS USED. THE UPPER LIMIT ON THE X INTEGRAL IS SQRT(T)=RT.
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/ CFI21/ CA,CB,CC,IKODE
      DATA TOL /0.5D-14/
      EXTERNAL DQI21
      IERR=0
      IF((C.EQ.0.0D0).AND.(KODE.EQ.1)) THEN
        QANS21=0.0D0
        IERR=1
        RETURN
      ENDIF
      IF((A.EQ.0.0D0).AND.(B.EQ.0.0D0).AND.(C.EQ.0.0D0)) THEN
C       KODE=2 HERE BECAUSE C=0 AND THE PREVIOUS TEST FAILED
        QANS21=T
        IERR=1
        RETURN
      ENDIF
      IF(T.EQ.0.0D0) THEN
        QANS21=0.0D0
        IERR=1
        RETURN
      ENDIF
      IF(A.EQ.0.0D0) THEN
        IF(KODE.EQ.1) THEN
          XLIM=MIN(B,C)/6.0D0
        ELSE
          XLIM=DSQRT(B*B+C*C)/6.0D0
        ENDIF
      ELSE
        IF(B.EQ.0.0D0) THEN
          XLIM=C/6.0D0
        ELSE
          IF(KODE.EQ.1) THEN
            XLIM=MIN(B,C)/6.0D0
```

```
          ELSE
            XLIM=DSQRT(B*B+C*C)/6.0D0
          ENDIF
        ENDIF
      ENDIF
      CA=A
      CB=B
      CC=C
      IKODE=KODE
      RT=DSQRT(T)
      JD=10
      INIT=-JD
      X1=0.0D0
      REL=TOL
      SS=0.0D0
      IF (RT.LE.XLIM) THEN
        SIG=RT/DBLE(FLOAT(JD))
        CALL DQUAD8(DQI21,INIT,X1,SIG,REL,X2,QANS,IERR)
        SS=SS+QANS
      ELSE
        SIG=XLIM/DBLE(FLOAT(JD))
        CALL DQUAD8(DQI21,INIT,X1,SIG,REL,X2,QANS,IERR)
        SS=SS+QANS
        INIT=-JD
        X1=X2
        SIG=(RT-X2)/DBLE(FLOAT(JD))
C       TEST FOR RT TOO CLOSE TO X2 FOR AN ACCURATE QUADRATURE
C       (SEE THE ERROR MESSAGE AND TEST IN DGAUS8)
        XMID=X1+0.5D0*SIG
        FMID=DQI21(XMID)
        IF(DABS(FMID*SIG).LT.2.0D0*TOL*DABS(SS)) GOTO 20
        REL=TOL
        CALL DQUAD8(DQI21,INIT,X1,SIG,REL,X2,QANS,KERR)
        IF(KERR.NE.1) THEN
          IERR=KERR
        ENDIF
        SS=SS+QANS
  20    CONTINUE
      ENDIF
      QANS21=SS
      RETURN
      END
      PROGRAM J21COMP
C
C     DONALD E. AMOS   OCTOBER, 2002; MAY, 2006
C
C     REF: FOLDER 21
C
C     J21COMP COMPARES THE PROCEDURE FROM SUBROUTINE INTEGJ21
C     AND DGAUS8 QUADRATURE EVALUATIONS OF
C
C          J21=INT ON (0,T) OF EXP(A^2*W+2*A*B)*EXP(-C*C/W)/(W**3/2)
C
C     AND A IS NON-NGATIVE AND B AND C ARE NON-NEGATIVE, BUT NOT ZERO AT
C     THE SAME TIME.
C
C     OUTPUT IS WRITTEN TO FILE J21COMP.TXT
C-----------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      OPEN(UNIT=7,FILE="J21COMP.TXT")
      WRITE (7,109)
 109  FORMAT('-----------------------------------------------------
     &-----------------'/
     &'LEGENDS FOR KFORM:'//
     &'KFORM DIGIT=0 MEANS ONE OR MORE PARMETERS ARE ZERO--A SPECIAL FOR
     &MULA IS USED'/
     &'KFORM DIGIT=1 MEANS T.LE.(C*C/36) WHERE ERF(C/DSQRT(T))=1.0D0'/
```

```
     &'KFORM DIGIT=2 MEANS THE CLOSED FORM IS USED FOR VOFT'/
     &'KFORM DIGIT=3 MEANS THE LARGE L.GE.2 SERIES IS USED FOR W3'/
     &'KFORM DIGIT=4 MEANS THE A*SQRT(T).LT.1 SERIES IS USED FOR I21'/
     &'KFORM DIGIT=5 MEANS THE THE QUADRATURE IS USED FOR W3'//
     &'EXAMPLE:'/
     &'KFORM=12 MEANS A COMBINATION OF 1 AND 2 WAS USED'/
     &'-----------------------------------------------------------------
     &----------'/)
C
      WRITE (7,110)
 110  FORMAT('      A         B         C          T         QUAD
     &REL ERR  IERR KFORM')
      DO 10 IT=1,11,2
        T=DBLE(FLOAT(IT))/10.0D0
        DO 20 IA=1,11,2
          A=DBLE(FLOAT(IA-1))/5.0D0
          DO 30 IB=1,11,2
            B=DBLE(FLOAT(IB-1))/2.0D0
            DO 40 IC=1,11,2
              C=DBLE(FLOAT(IC-1))/5.0D0
              CALL INTEGJ21(A,B,C,T,ANS2,J2ERR,KFORM)
              IF((J2ERR.NE.0)) THEN
                WRITE (7,405) A,B,C,T,J2ERR,KFORM
 405            FORMAT( 4D11.4,1X,'IN INTEGJ21, J2ERR=',I2,' KFORM=',I2)
                GOTO 40
              ENDIF
C-----------------------------------------------------------------------
C     LEGENDS FOR KFORM:
C
C     KFORM DIGIT=1 MEANS T.LE.(C*C/36) WHERE ERF(C/DSQRT(T))=1.0D0
C     KFORM DIGIT=2 MEANS THE CLOSED FORM IS USED FOR VOFT
C     KFORM DIGIT=3 MEANS THE LARGE L.GE.2 SERIES IS USED FOR W3
C     KFORM DIGIT=4 MEANS THE A*SQRT(T).LT.1 SERIES IS USED FOR I21
C     KFORM DIGIT=5 MEANS THE THE QUADRATURE IS USED FOR W3
C
C     EXAMPLE:
C     KFORM=12 MEANS A COMBINATION OF 1 AND 2 WAS USED
C-----------------------------------------------------------------------
              CALL J21QUAD(A,B,C,T,QANS,IERR)
              ERR=DABS(ANS2-QANS)
              IF (QANS.NE.0.0D0) THEN
                ERR=ERR/DABS(QANS)
              ENDIF
              WRITE (7,200) A,B,C,T,QANS,ERR,IERR,KFORM
 200          FORMAT(4D11.4,D12.5,D12.4,2I4)
   40       CONTINUE
   30     CONTINUE
   20   CONTINUE
   10 CONTINUE
      END
      DOUBLE PRECISION FUNCTION DQJ21(X)
C
C     DQJ21 COMPUTES THE INTEGRAND FOR THE QUADRATURE COMPARISON
C     OF J21 WITH INTEGJ21
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/ CFJ21/ CA,CB,CC
C     A CHANGE OF VARIABLE T=X*X REMOVES THE SQRT FROM THE INTEGRAND
C     THE UPPER LIMIT ON THE INTEGRAL IS THEN DSQRT(T)
      ARG=CA*X+CB/X
      KODE=3
      NZ=0
      ERFX=DRERF(ARG,KODE,NZ)
      X2=X*X
      ARG=CB*CB/X2
      ERFX=ERFX*DEXP(-ARG)
      ARG=CC*CC/X2
```

```
      DQJ21=2.0D0*ERFX*DEXP(-ARG)/X2
      RETURN
      END
      SUBROUTINE J21QUAD(A,B,C,T,QANS21,IERR)
C     QUADRATURE FOR THE J21 INTEGRAL. A CHANGE OF VARIABLES TAU=X*X
C     IS USED. THE UPPER LIMIT ON THE X INTEGRAL IS SQRT(T)=RT.
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/ CFJ21/ CA,CB,CC
      DATA TOL /0.5D-14/
      EXTERNAL DQJ21
      IERR=0
      IF((B.EQ.0.0D0).AND.(C.EQ.0.0D0)) THEN
C       THE INTEGRAL DOES NOT EXIST--WRONG INPUT
        QANS21=0.0D0
        IERR=2
        RETURN
      ENDIF
      IF(T.EQ.0.0D0) THEN
        QANS21=0.0D0
        IERR=1
        RETURN
      ENDIF
      CA=A
      CB=B
      CC=C
      RT=DSQRT(T)
      JD=10
      INIT=-JD
      X1=0.0D0
      REL=TOL
      SS=0.0D0
      XLIM=DSQRT(B*B+C*C)/6.0D0
      IF (RT.LE.XLIM) THEN
        SIG=RT/DBLE(FLOAT(JD))
        CALL DQUAD8(DQJ21,INIT,X1,SIG,REL,X2,QANS,IERR)
        SS=SS+QANS
      ELSE
        SIG=XLIM/DBLE(FLOAT(JD))
        CALL DQUAD8(DQJ21,INIT,X1,SIG,REL,X2,QANS,IERR)
        SS=SS+QANS
        INIT=-JD
        X1=X2
        SIG=(RT-X2)/DBLE(FLOAT(JD))
C       TEST FOR RT TOO CLOSE TO X2 FOR AN ACCURATE QUADRATURE
C       (SEE THE ERROR MESSAGE AND TEST IN DGAUS8)
        XMID=X1+0.5D0*SIG
        FMID=DQJ21(XMID)
        IF(DABS(FMID*SIG).LT.2.0D0*TOL*DABS(SS)) GOTO 20
        REL=TOL
        CALL DQUAD8(DQJ21,INIT,X1,SIG,REL,X2,QANS,KERR)
        IF(KERR.NE.1) THEN
          IERR=KERR
        ENDIF
        SS=SS+QANS
   20   CONTINUE
      ENDIF
      QANS21=SS
      RETURN
      END
      PROGRAM I22COMP
C
C     DONALD E. AMOS   NOVENBER, 2002; MAY, 2006
C
C     REF: FOLDER 22
C
C     I22COMP COMPARES THE PROCEDURE FROM SUBROUTINE INTEGI22
C     AND DGAUS8 QUADRATURE EVALUATIONS OF
```

```
C
C          I22=INT ON (0,T) OF EXP(A^2*W+2*A*B)*EXP(-C*C/W)/SQRT(W)
C
C     AND A IS NON-NGATIVE AND B AND C ARE NON-NEGATIVE, BUT NOT ZERO AT
C     THE SAME TIME.
C
C     OUTPUT IS WRITTEN TO FILE I22COMP.TXT
C-----------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      OPEN(UNIT=7,FILE="I22COMP.TXT")
      WRITE (7,109)
 109  FORMAT('-------------------------------------------------------------
     &-----------------'/
     &'LEGENDS FOR KFORM:'//
     &'KFORM DIGIT=0 MEANS ONE OR MORE PARMETERS ARE ZERO--A SPECIAL FOR
     &MULA IS USED'/
     &'KFORM DIGIT=1 MEANS T.LE.(C*C/36) WHERE ERF(C/DSQRT(T))=1.0D0'/
     &'KFORM DIGIT=2 MEANS THE CLOSED FORM IS USED FOR VOFT'/
     &'KFORM DIGIT=3 MEANS THE LARGE L.GE.2 SERIES IS USED FOR W3'/
     &'KFORM DIGIT=4 MEANS THE A*SQRT(T).LT.1 SERIES IS USED FOR I21'/
     &'KFORM DIGIT=5 MEANS THE THE QUADRATURE IS USED FOR W3'//
     &'EXAMPLE:'/
     &'KFORM=12 MEANS A COMBINATION OF 1 AND 2 WAS USED'/
     &'-------------------------------------------------------------
     &----------'/)
C
      WRITE (7,110)
 110  FORMAT('      A         B         C          T         QUAD
     &REL ERR  IERR KFORM')
      DO 10 IT=1,11,2
        T=DBLE(FLOAT(IT))/10.0D0
        DO 20 IA=1,11,2
          A=DBLE(FLOAT(IA-1))/5.0D0
          DO 30 IB=1,11,2
            B=DBLE(FLOAT(IB-1))/2.0D0
            DO 40 IC=1,11,2
              C=DBLE(FLOAT(IC-1))/5.0D0
              CALL INTEGI22(A,B,C,T,ANS2,I2ERR,KFORM)
              IF((I2ERR.NE.0)) THEN
                WRITE (7,405) A,B,C,T,I2ERR,KFORM
 405            FORMAT( 4D11.4,1X,'IN INTEGI22, I2ERR=',I2,' KFORM=',I2)
                GOTO 40
              ENDIF
C-----------------------------------------------------------------------
C     LEGENDS FOR KFORM:
C
C     KFORM DIGIT=1 MEANS T.LE.(C*C/36) WHERE ERF(C/DSQRT(T))=1.0D0
C     KFORM DIGIT=2 MEANS THE CLOSED FORM IS USED FOR VOFT
C     KFORM DIGIT=3 MEANS THE LARGE L.GE.2 SERIES IS USED FOR W3
C     KFORM DIGIT=4 MEANS THE A*SQRT(T).LT.1 SERIES IS USED FOR I21
C     KFORM DIGIT=5 MEANS THE THE QUADRATURE IS USED FOR W3
C
C     EXAMPLE:
C     KFORM=12 MEANS A COMBINATION OF 1 AND 2 WAS USED
C-----------------------------------------------------------------------
              CALL I22QUAD(A,B,C,T,QANS,IERR)
              ERR=DABS(ANS2-QANS)
              IF (QANS.NE.0.0D0) THEN
                ERR=ERR/DABS(QANS)
              ENDIF
              WRITE (7,200) A,B,C,T,QANS,ERR,IERR,KFORM
 200          FORMAT(4D11.4,D12.5,D12.4,2I4)
   40       CONTINUE
   30     CONTINUE
   20   CONTINUE
   10 CONTINUE
      END
```

```
      DOUBLE PRECISION FUNCTION DQI22(X)
C
C     DQI22 COMPUTES THE INTEGRAND FOR THE QUADRATURE COMPARISON
C     OF I22 WITH INTEGI22
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/ CFI22/ CA,CB,CC
C     A CHANGE OF VARIABLE T=X*X REMOVES THE SQRT FROM THE INTEGRAND.
C     THE UPPER LIMIT ON THE INTEGRAL IS THEN DSQRT(T).
      ARG=CA*X+CB/X
      KODE=3
      NZ=0
      ERFX=DRERF(ARG,KODE,NZ)
      X2=X*X
      ARG=CB*CB/X2
      ERFX=ERFX*DEXP(-ARG)
      ARG=CC*CC/X2
      DQI22=2.0D0*ERFX*DEXP(-ARG)
      RETURN
      END
      SUBROUTINE I22QUAD(A,B,C,T,QANS22,IERR)
C     QUADRATURE FOR THE I22 INTEGRAL. A CHANGE OF VARIABLES TAU=X*X
C     IS USED. THE UPPER LIMIT ON THE X INTEGRAL IS SQRT(T)=RT.
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/ CFI22/ CA,CB,CC
      DATA TOL /0.5D-14/
      EXTERNAL DQI22
      IERR=0
      IF((B.EQ.0.0D0).AND.(C.EQ.0.0D0)) THEN
        XLIM=4.0D0
      ELSE
        XLIM=DSQRT(B*B+C*C)/6.0D0
      ENDIF
      IF(T.EQ.0.0D0) THEN
        QANS21=0.0D0
        IERR=1
        RETURN
      ENDIF
      CA=A
      CB=B
      CC=C
      RT=DSQRT(T)
      JD=10
      INIT=-JD
      X1=0.0D0
      REL=TOL
      SS=0.0D0
      IF (RT.LE.XLIM) THEN
        SIG=RT/DBLE(FLOAT(JD))
        CALL DQUAD8(DQI22,INIT,X1,SIG,REL,X2,QANS,IERR)
        SS=SS+QANS
      ELSE
        SIG=XLIM/DBLE(FLOAT(JD))
        CALL DQUAD8(DQI22,INIT,X1,SIG,REL,X2,QANS,IERR)
        SS=SS+QANS
        INIT=-JD
        X1=X2
        SIG=(RT-X2)/DBLE(FLOAT(JD))
C       TEST FOR RT TOO CLOSE TO X2 FOR AN ACCURATE QUADRATURE
C       (SEE THE ERROR MESSAGE AND TEST IN DGAUS8)
        XMID=X1+0.5D0*SIG
        FMID=DQI22(XMID)
        IF(DABS(FMID*SIG).LT.2.0D0*TOL*DABS(SS)) GOTO 20
        REL=TOL
        CALL DQUAD8(DQI22,INIT,X1,SIG,REL,X2,QANS,KERR)
        IF(KERR.NE.1) THEN
          IERR=KERR
```

```
        ENDIF
        SS=SS+QANS
  20    CONTINUE
      ENDIF
      QANS22=SS
      RETURN
      END
      PROGRAM J22COMP
C
C     DONALD E. AMOS   NOVEMBER, 2002; MAY, 2006
C
C     REF: FOLDER 22
C
C     J22COMP COMPARES THE PROCEDURE FROM SUBROUTINE INTEGJ22
C     AND DGAUS8 QUADRATURE EVALUATIONS OF
C
C          J22=INT ON (0,T) OF EXP(A^2*W+2*A*B)*EXP(-C*C/W)*SQRT(W)
C
C     AND A IS NON-NGATIVE AND B AND C ARE NON-NEGATIVE, BUT NOT ZERO AT
C     THE SAME TIME.
C
C     OUTPUT IS WRITTEN TO FILE J22COMP.TXT
C-------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      OPEN(UNIT=7,FILE="J22COMP.TXT")
      WRITE (7,109)
 109  FORMAT('---------------------------------------------------------
     &-----------------'/
     &'LEGENDS FOR KFORM:'//
     &'KFORM DIGIT=0 MEANS ONE OR MORE PARMETERS ARE ZERO--A SPECIAL FOR
     &MULA IS USED'/
     &'KFORM DIGIT=1 MEANS T.LE.(C*C/36) WHERE ERF(C/DSQRT(T))=1.0D0'/
     &'KFORM DIGIT=2 MEANS THE CLOSED FORM IS USED FOR VOFT'/
     &'KFORM DIGIT=3 MEANS THE LARGE L.GE.2 SERIES IS USED FOR W3'/
     &'KFORM DIGIT=4 MEANS THE A*SQRT(T).LT.1 SERIES IS USED FOR I21'/
     &'KFORM DIGIT=5 MEANS THE THE QUADRATURE IS USED FOR W3'//
     &'EXAMPLE:'/
     &'KFORM=12 MEANS A COMBINATION OF 1 AND 2 WAS USED'/
     &'-----------------------------------------------------------------
     &----------'/)
C
      WRITE (7,110)
 110  FORMAT('      A         B         C          T         QUAD
     &REL ERR  IERR KFORM')
      DO 10 IT=1,11,2
        T=DBLE(FLOAT(IT))/10.0D0
        DO 20 IA=1,11,2
          A=DBLE(FLOAT(IA-1))/5.0D0
          DO 30 IB=1,11,2
            B=DBLE(FLOAT(IB-1))/2.0D0
            DO 40 IC=1,11,2
              C=DBLE(FLOAT(IC-1))/5.0D0
              CALL INTEGJ22(A,B,C,T,ANS2,J2ERR,KFORM)
              IF((J2ERR.NE.0)) THEN
                WRITE (7,405) A,B,C,T,J2ERR,KFORM
 405            FORMAT( 4D11.4,1X,'IN INTEGJ22, J2ERR=',I2,' KFORM=',I2)
                GOTO 40
              ENDIF
C--------------------------------------------------------------------
C     LEGENDS FOR KFORM:
C
C     KFORM DIGIT=1 MEANS T.LE.(C*C/36) WHERE ERF(C/DSQRT(T))=1.0D0
C     KFORM DIGIT=2 MEANS THE CLOSED FORM IS USED FOR VOFT
C     KFORM DIGIT=3 MEANS THE LARGE L.GE.2 SERIES IS USED FOR W3
C     KFORM DIGIT=4 MEANS THE A*SQRT(T).LT.1 SERIES IS USED FOR I21
C     KFORM DIGIT=5 MEANS THE THE QUADRATURE IS USED FOR W3
C
```

```
C     EXAMPLE:
C     KFORM=12 MEANS A COMBINATION OF 1 AND 2 WAS USED
C-----------------------------------------------------------------------
              CALL J22QUAD(A,B,C,T,QANS,IERR)
              ERR=DABS(ANS2-QANS)
              IF (QANS.NE.0.0D0) THEN
                ERR=ERR/DABS(QANS)
              ENDIF
              WRITE (7,200) A,B,C,T,QANS,ERR,IERR,KFORM
 200          FORMAT(4D11.4,D12.5,D12.4,2I4)
   40       CONTINUE
   30     CONTINUE
   20   CONTINUE
   10 CONTINUE
      END
      DOUBLE PRECISION FUNCTION DQJ22(X)
C
C     DQJ22 COMPUTES THE INTEGRAND FOR THE QUADRATURE COMPARISON
C     OF J22 WITH INTEGJ22
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/ CFJ22/ CA,CB,CC
C     A CHANGE OF VARIABLE T=X*X REMOVES THE SQRT FROM THE INTEGRAND
C     THE UPPER LIMIT ON THE INTEGRAL IS THEN DSQRT(T)
      ARG=CA*X+CB/X
      KODE=3
      NZ=0
      ERFX=DRERF(ARG,KODE,NZ)
      X2=X*X
      ARG=CB*CB/X2
      ERFX=ERFX*DEXP(-ARG)
      ARG=CC*CC/X2
      DQJ22=2.0D0*ERFX*DEXP(-ARG)*X2
      RETURN
      END
      SUBROUTINE J22QUAD(A,B,C,T,QANS21,IERR)
C     QUADRATURE FOR THE J22 INTEGRAL. A CHANGE OF VARIABLES TAU=X*X
C     IS USED. THE UPPER LIMIT ON THE X INTEGRAL IS SQRT(T)=RT.
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/ CFJ22/ CA,CB,CC
      DATA TOL /0.5D-14/
      EXTERNAL DQJ22
      IERR=0
      IF((B.EQ.0.0D0).AND.(C.EQ.0.0D0)) THEN
        XLIM=4.0D0
      ELSE
        XLIM=DSQRT(B*B+C*C)/6.0D0
      ENDIF
      IF(T.EQ.0.0D0) THEN
        QANS21=0.0D0
        IERR=1
        RETURN
      ENDIF
      CA=A
      CB=B
      CC=C
      RT=DSQRT(T)
      JD=10
      INIT=-JD
      X1=0.0D0
      REL=TOL
      SS=0.0D0
      IF (RT.LE.XLIM) THEN
        SIG=RT/DBLE(FLOAT(JD))
        CALL DQUAD8(DQJ22,INIT,X1,SIG,REL,X2,QANS,IERR)
        SS=SS+QANS
      ELSE
```

```
        SIG=XLIM/DBLE(FLOAT(JD))
        CALL DQUAD8(DQJ22,INIT,X1,SIG,REL,X2,QANS,IERR)
        SS=SS+QANS
        INIT=-JD
        X1=X2
        SIG=(RT-X2)/DBLE(FLOAT(JD))
C       TEST FOR RT TOO CLOSE TO X2 FOR AN ACCURATE QUADRATURE
C       (SEE THE ERROR MESSAGE AND TEST IN DGAUS8)
        XMID=X1+0.5D0*SIG
        FMID=DQJ22(XMID)
        IF(DABS(FMID*SIG).LT.2.0D0*TOL*DABS(SS)) GOTO 20
        REL=TOL
        CALL DQUAD8(DQJ22,INIT,X1,SIG,REL,X2,QANS,KERR)
        IF(KERR.NE.1) THEN
          IERR=KERR
        ENDIF
        SS=SS+QANS
  20    CONTINUE
      ENDIF
      QANS21=SS
      RETURN
      END
      PROGRAM I29COMP
C
C     DONALD E. AMOS   AUGUST, 2006
C
C     REF: FOLDER 29, FOLDER 28
C
C     I29COMP COMPARES THE N MEMBER SEQUENCE GENERATED BY INTEGI29 FOR
C
C        INT ON (T,INF) OF EXP(-A^2*W^2)*I(N0+2*K-2)ERFC(B*W), K=1,N
C
C     WITH QUADRATURES ON (T,INF). THE N MEMBERS OF THE SEQUENCE START
C     WITH ORDER N0 AND CONSECUTIVE MEMBERS HAVE ORDERS 2 APART, ENDING
C     AT ORDER N0+2*N-2. IF N0 IS EVEN, INTEGI29 GENERATES EVEN ORDERS;
C     IF N0 IS ODD, INTEGI29 GENERATES ODD ORDERS. HERE I(M)ERFC(X) IS
C     THE ITERATED COERROR FUNCTION OF ORDER M.
C
C     OUTPUT IS WRITTEN TO FILE I29COMP.TXT
C-----------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION YN(10)
      COMMON /CQI29/AA,BB,KK
      DATA RTPI /1.772453850905516D0/
      DATA REL  /0.5D-14/
      EXTERNAL QI29
      OPEN(FILE='I29COMP.TXT',UNIT=7)
      DO 2 NSEQ=1,3
      WRITE(7,50)NSEQ
 50   FORMAT(25X,'SEQUENCE LENGTH = ',I3)
      WRITE(7,100)
 100  FORMAT(5X,"A",10X,"B",10X,'T',7X,'N0',3X,'QANS',7X,'YANS',6X,'REL
     &ERR',3X,'IRY',1X,'IRQ')
      DO 5 NI=0,3
        N0=NI
        DO 10 IB=3,9,2
          B=DBLE(FLOAT(IB))/2.0D0
          DO 20 IA=3,9,2
            A=DBLE(FLOAT(IA))/2.0D0
            DO 30 IT=1,11,2
              T=DBLE(FLOAT(IT-1))/5.0D0
              CALL INTEGI29(A,B,T,N0,NSEQ,YN,IERRY)
C             COMPARE A QUADRATURE WITH EACH MEMBER OF THE SEQUENCE
C             YN(K),K=1,NSEQ
              DO 40 K=1,NSEQ
              AA=A
              BB=B
```

```
                KK=N0+2*K-2
                INIT=0
                X1=T
                SIG=6.0D0/DSQRT(A*A+B*B)
                TOL=REL
                CALL DQUAD8(QI29,INIT,X1,SIG,TOL,X2,QANS,IERRQ)
                ERR=DABS(QANS-YN(K))
                IF(DABS(QANS).NE.0.0D0) THEN
                  ERR=ERR/DABS(QANS)
                ENDIF
                IF(K.EQ.1)THEN
                  WRITE (7,101) A,B,T,N0,QANS,YN(K),ERR,IERRY,IERRQ
 101              FORMAT(3D11.4,I3,3D11.4,2I4)
                ELSE
                  WRITE(7,102)QANS,YN(K),ERR,IERRY,IERRQ
 102              FORMAT(36X,3D11.4,2I4)
                ENDIF
 40             CONTINUE
 30           CONTINUE
 20         CONTINUE
 10       CONTINUE
 5      CONTINUE
 2      CONTINUE
        END
        DOUBLE PRECISION FUNCTION QI29(W)
C-----------------------------------------------------------------------
C       COMPUTATION OF THE INTEGRAND FOR THE QUADRATURE ON
C
C          INT ON (T,INF) OF EXP(-AA^2*W^2)*I(KK)ERFC(BB*W)
C
C       AA, BB, KK ARE THE PARAMETERS IN COMMON/CQI29/
C-----------------------------------------------------------------------
        IMPLICIT DOUBLE PRECISION (A-H,O-Z)
        DIMENSION YY(1)
        COMMON /CQI29/AA,BB,KK
        DATA RTPI /1.77245385090551 6D0/
        DATA REL /0.5D-14/
        AW=AA*W
        BW=BB*W
        ARGA=AW*AW
        EX=DEXP(-ARGA)
        X=BW
        N0=KK
        KODE=1
        NI=1
        TOL=REL
        CALL DINERFC(X,N0,KODE,NI,TOL,YY,NZ)
        QI29=EX*YY(1)
        RETURN
        END
        PROGRAM GNCOMP
C
C       DONALD E. AMOS,   SEPTEMBER, 2002
C
C       REF: FOLDER 21
C
C       GNCOMP STUDIES THE FORWARD AND BACKWARD 3-TERM RECURRENCE FOR
C       EVALUATIONS OF
C
C        G(N)=INT ON (CAPT,INF) OF EXP(-A^2*X^2)*I(N)ERFC(B*X)/X**N
C                          CAPT=1/DSQRT(T)
C
C       THE RECURRENCE IS N(N+1)*Y(N+1)+2*N*B*CAPT*Y(N)+ABTRM*Y(N-1)
C                         =[I(N-1)ERFC(B*CAPT)*EXP(B^2*CAPT^2)]
C
C       WHERE Y(N)=2.0D0*(CAPT**(N-1))*EXP(ABTRM)*G(N), WHERE
C       I(N)ERFC IS THE ITERATED COERROR FUNCTION OF ORDER N, AND
```

```
C     ABTRM=(A^2+B^2)/T=(A^2+B^2)*CAPT*CAPT.
C
C     Y(N) CAN BE ESTIMATED FOR AN ORDER OF MAGNITUDE RESULT BY THE
C     UPPER BOUND
C
C             [EXP(ABTRM)*I(N)ERFC(B*CAPT)]*E((N+1)/2,C^2*T^2)
C
C     A STABILITY ANALYSIS INDICATES THAT FORWARD AND BACKWARD FROM
C     INDEX NFB=INT(2.0*ABTRM) IS STABLE. THE NUMERICAL RESULTS
C     SEEM TO CONFIRM THIS RESULT.
C
C     THE CODE SETS A VALUE OF M=50 FOR THE UPPER INDEX OF THE SEQUENCE
C     AND COMPUTES LOWER INDICES BY RECURRING UP AND DOWN FROM INDEX
C     NFB IN SUBROUTINE RECURR. THE RESULTS ARE COMPARED TO A QUADRATURE
C     ON THE ORIGINAL INTEGRAL USING SUBROUTINE DQUAD8.
C
C     OUTPUT IS WRITTEN TO FILE GNCOMP.TXT
C-----------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION YN(100),SY(1)
      COMMON /CGSEQ/ KN,CREL,CABTRM,CBT
      EXTERNAL DGNFUN
      OPEN(UNIT=7,FILE="GNCOMP.TXT")
      WRITE(7,109)
 109  FORMAT('  NFB=INDEX WHICH SEPARATES FORWARD AND BACKWARD RECURRENC
     *E')
      WRITE (7,110)
 110  FORMAT('      A          B         T     N  NFB     QANS      RELE
     &RR  IERR UPPER BOUND')
      DO 5 IT=1,11,2
      T=DBLE(FLOAT(IT))/5.0D0
      RT=DSQRT(T)
      CAPT=1.0D0/RT
      DO 10 IA=1,11,2
        A=DBLE(FLOAT(IA-1))/5.00D0
        DO 20 IB=1,11,2
          B=DBLE(FLOAT(IB))/5.0D0
          M=40
C          MM=51
          REL=0.50D-13
          DREL=REL
C         YN(K+1) IS G(K)*(CAPT**(K-1))*DEXP(ABTRM), K=0,1,...
C         ABTRM=(A*A+B*B)*CAPT*CAPT, CAPT=1/DSQRT(T)
          CALL GNSEQ(A,B,CAPT,M,DREL,YN)
          DO 30 IN=1,M
C           SET COMMON PARAMETERS FOR DGFUN
            A2=A*A
            B2=B*B
            CBT=B*CAPT
            ABTRM=(A2+B2)/T
            CABTRM=ABTRM
            CREL=REL
            ARG=ABTRM+ABTRM
            NFB=INT(ARG)
            KN=IN-1
            ABSUM=A2+B2
            DSIG=RT/DSQRT(ABSUM)
            X1=0.0D0
            INIT=0
            DREL=REL
            CALL DQUAD8(DGNFUN,INIT,X1,DSIG,DREL,X2,QANS,IERR)
            GANS=YN(KN+1)
            ERR=DABS(GANS-QANS)
            IF (QANS.NE.0.0D0) THEN
              ERR=ERR/DABS(QANS)
            ENDIF
C       COMPUTE AN ESTIMATE OF G(KN)*(CAPT**KN)*[EXP(ABTRM)]
```

```
          N0=KN
          IF((A.EQ.0.0D0).AND.(N0.LT.2)) THEN
            WRITE (7,200) A,B,T,KN,NFB,QANS,ERR,IERR
            GO TO 30
          ENDIF
          KODE=2
          MM=1
          BT=B*CAPT
          CALL DINERFC(BT,N0,KODE,MM,REL,SY,NZ)
          EST=SY(1)
          AT=A*CAPT
          ARG=AT*AT
          KODE=2
          MM=1
          TOL=REL
          IF(MOD(N0+1,2).EQ.0) THEN
            IK=(N0+1)/2
            CALL DEXINT(ARG, IK, KODE, MM, TOL, SY, NZ, IERR)
          ELSE
            FNH=DBLE(FLOAT(N0/2))+0.5D0
            CALL DHEXINT(ARG, FNH, KODE, MM, TOL, SY, NZ, IERR)
          ENDIF
          EST=EST*SY(1)
          WRITE (7,200) A,B,T,KN,NFB,QANS,ERR,IERR,EST
 200      FORMAT(3D10.3,2I4,2D12.4,I4,D12.4)
  30    CONTINUE
        WRITE (7,111)
 111    FORMAT(/)
        PAUSE
  20  CONTINUE
  10 CONTINUE
  5  CONTINUE
     END
     PROGRAM VTCOMP
C
C    DONALD E. AMOS   DECEMBER, 2002; MAY, 2006
C
C    REF: FOLDER 21
C
C    VTCOMP COMPARES THE PROCEDURE FROM DOUBLE PRECISION FUNCTION DVOFT
C    AND DGAUS8 QUADRATURE EVALUATIONS OF
C
C         DVOFT=INT ON (0,T) OF U(A,B,W)
C
C    WHERE  U(A,B,W)=EXP(A*A*W+2*A*B)*ERFC(A*RW+B/RW), RW=SQRT(W)
C
C    AND A, B ARE NON-NGATIVE.
C
C    OUTPUT IS WRITTEN TO FILE VTCOMP.TXT
C-----------------------------------------------------------------------
     IMPLICIT DOUBLE PRECISION (A-H,O-Z)
     OPEN(UNIT=7,FILE="VTCOMP.TXT")
C
     WRITE (7,110)
 110 FORMAT('     A         B          T         QUAD        REL ERR
    &IERR')
     DO 10 IT=1,11,2
       T=DBLE(FLOAT(IT))/5.0D0
       DO 20 IA=1,11,2
         A=DBLE(FLOAT(IA-1))/5.0D0
         DO 30 IB=1,15,2
           B=DBLE(FLOAT(IB-1))/2.0D0
           REL=0.5D-14
           ANS2=DVOFT(A,B,T,REL,IVERR,KFORM)
           IF((IVERR.NE.0)) THEN
             WRITE (7,405) A,B,T,IVERR
 405         FORMAT( 3D11.4,1X,'IN DVOFT, IVERR=',I2)
```

```
            GOTO 30
          ENDIF
          CALL VTQUAD(A,B,T,QANS,IERR)
          ERR=DABS(ANS2-QANS)
          IF (QANS.NE.0.0D0) THEN
            ERR=ERR/DABS(QANS)
          ENDIF
          WRITE (7,200) A,B,T,QANS,ERR,IERR
 200      FORMAT(3D11.4,D12.5,D12.4,I4)
   30   CONTINUE
   20 CONTINUE
   10 CONTINUE
      END
      DOUBLE PRECISION FUNCTION DQVT(X)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/ CFVT/ CA,CB
C     A CHANGE OF VARIABLE TAU=X*X REMOVES THE SQRT FROM THE INTEGRAND
C     THE UPPER LIMIT ON THE INTEGRAL IS THEN DSQRT(T)
      ARG=CA*X+CB/X
      KODE=3
      ERFX=DRERF(ARG,KODE,NZ)
      X2=X*X
      ARG=CB*CB/X2
      DQVT=2.0D0*X*ERFX*DEXP(-ARG)
      RETURN
      END
      SUBROUTINE VTQUAD(A,B,T,QANSVT,IERR)
C     QUADRATURE FOR THE VOFT FUNCTION. A CHANGE OF VARIABLES TAU=X*X
C     IS USED. THE UPPER LIMIT ON THE X INTEGRAL IS SQRT(T)=RT.
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/ CFVT/ CA,CB
      DATA TOL /0.50D-14/
      EXTERNAL DQVT
      IERR=0
      IF(B.EQ.0.0D0) THEN
        XLIM=4.0D0
      ELSE
        XLIM=B/6.0D0
      ENDIF
      IF(T.EQ.0.0D0) THEN
        QANSVT=0.0D0
        IERR=1
        RETURN
      ENDIF
      CA=A
      CB=B
      RT=DSQRT(T)
      JD=10
      INIT=-JD
      X1=0.0D0
      REL=TOL
      SS=0.0D0
      IF (RT.LE.XLIM) THEN
        SIG=RT/DBLE(FLOAT(JD))
        CALL DQUAD8(DQVT,INIT,X1,SIG,REL,X2,QANS,IERR)
        SS=SS+QANS
      ELSE
        SIG=XLIM/DBLE(FLOAT(JD))
        CALL DQUAD8(DQVT,INIT,X1,SIG,REL,X2,QANS,IERR)
        SS=SS+QANS
        INIT=-JD
        X1=X2
        SIG=(RT-X2)/DBLE(FLOAT(JD))
C       TEST FOR RT TOO CLOSE TO X2 FOR AN ACCURATE QUADRATURE
C       (SEE THE ERROR MESSAGE AND TEST IN DGAUS8)
        XMID=X1+0.5D0*SIG
        FMID=DQVT(XMID)
```

```
        IF(DABS(FMID*SIG).LT.2.0D0*TOL*DABS(SS)) GOTO 20
        REL=TOL
        CALL DQUAD8(DQVT,INIT,X1,SIG,REL,X2,QANS,KERR)
        IF(KERR.NE.1) THEN
          IERR=KERR
        ENDIF
        SS=SS+QANS
 20     CONTINUE
      ENDIF
      QANSVT=SS
      RETURN
      END
```

```
      PROGRAM APRGINFO
C-----------------------------------------------------------------------
C     DONALD E. AMOS, MAY 1, 2002
C
C     THIS FILE CONTAINS THE DRIVERS WHICH WILL EXERCIZE SUBROUTINES IN
C     FILE AMOSSUBS.FOR. THEY SERVE TO DEMOMSTRATE CALL PROCEDURES, A
C     NOMINAL USE OF ERROR FLAGS AND TYPICAL RELATIVE ERRORS.
C
C     USAGE:
C
C     EXTRACT ONE OF THE PROGRAMS BELOW ALONG WITH ANY SUCCEEDING
C     DOUBLE PRECISION FUNCTION(S) OR SUBROUTINES, COMPILE, LINK
C     COMPILED FILE AMOSSUBS.FOR, AND THEN EXECUTE THE LINKED FILES.
C     OUTPUT IS WRITTEN TO A TEXT FILE(*.TXT) WITH THE NAME OF THE
C     PROGRAM BEING EXECUTED.
C
C     PROGRAMS:                 OUTPUT FILE:                 FOLDER
C              PROGRAM GECOMP               GECOMP.TXT         18
C              PROGRAM GHECOMP              GHECOMP.TXT        18
C              PROGRAM DFCOMP               DFCOMP.TXT         16
C              PROGRAM DGCOMP               DGCOMP.TXT         16
C              PROGRAM HERFCOMP             HERFCOMP.TXT       23
C-----------------------------------------------------------------------
      END
      PROGRAM GECOMP
C
C     COMPARES INTEGER ORDER ROUTINES DGEXINT AND DEXINT
C     AGAINST A QUADRATURE. OUTPUT IS WRITTEN TO FILE GECOMP.TXT
C-----------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION ENI(100),EN(100)
      COMMON /CQFUN/AX, NU, KTEST
      EXTERNAL DQFUN
      OPEN(UNIT=7,FILE="GECOMP.TXT")
      WRITE (7,110)
 110  FORMAT('     X           QUAD          RELERR1  IERR1 IERR2')
 111  CONTINUE
      PRINT *,'INPUT 1 TO TEST DGEXINT; 2 TO TEST DEXINT: '
      READ *,ICASE
      IF((ICASE.LT.1).OR.(ICASE.GT.2)) THEN
        GOTO 111
      ENDIF
      IF(ICASE.EQ.1) THEN
      PRINT *,'INPUT 1 TO TEST G SEQUENCE; INPUT 2 TO TEST E SEQUENCE: '
      READ *, ITEST
      ELSE
        ITEST=2
      ENDIF
      PRINT *,'INPUT STARTING VALUE OF ORDER (INTEGER > 0): '
      READ *, NIN
      IF((ITEST.LT.1).OR.(ITEST.GT.2)) GOTO 111
      IF(NIN.LT.1) GOTO 111
      DO 3 KODE=1,2
        WRITE(7,308)KODE
 308    FORMAT(/'                         KODE = ',I3/)
      KTEST=ITEST
      N=NIN
      DO 5 MM=1,5
        IF(ICASE.EQ.1) THEN
          IF(ITEST.EQ.1) THEN
            WRITE(7,305) MM,N,KODE
 305    FORMAT(/'TESTING G SEQ IN DGEXINT',' MM = ',I3,' N = ',I3,
     &' KODE = ',I3)
          ELSE
            WRITE(7,306) MM,N,KODE
 306    FORMAT(/'TESTING E SEQ IN DGEXINT',' MM = ',I3,' N = ',I3,
     &' KODE = ',I3)
```

```
          ENDIF
        ELSE
          WRITE(7,307) MM,N,KODE
 307    FORMAT(/'TESTING E SEQ IN DEXINT',' MM = ',I3,' N = ',I3,
     &' KODE = ',I3)
        ENDIF
        NIX=50
        DO 10 IX=1,NIX
          X=DBLE(FLOAT(IX))/5.0D0
C         KODE IS DEFINED ABOVE
          REL=0.5D-15
          TOL=REL
          IF(ICASE.EQ.1) THEN
            CALL DGEXINT(X, N, KODE, MM, TOL, ENI, EN, NZ, IERR)
          ELSE
            CALL DEXINT(X, N, KODE, MM, TOL,  EN, NZ, IERR)
          ENDIF
          DEX=DEXP(X)
        DO 50 J=1,MM
          IF(ITEST.EQ.1) THEN
            Y=ENI(J)
          ELSE
            Y=EN(J)
          ENDIF
          INIT=0
          TOL=0.5D-14
          AX=X
          NU=N+J-1
          X1=1.0D0
          SIG=3.0D0/X
          CALL DQUAD8(DQFUN,INIT,X1,SIG,TOL,X2,QANS,IERR1)
C         CONTINUE QUADRATURE PAST X2 TO BE SURE THE TRUNCATION IS
C         ADEQUATE
          SIG=6.0D0/X
          CALL DQUAD8(DQFUN,INIT,X1,SIG,TOL,X2,QANS,IERR2)
          IF(KODE.EQ.2) QANS=QANS*DEX
          ERR=DABS(Y-QANS)
          IF (QANS.NE.0.0D0) THEN
            ERR1=ERR/DABS(QANS)
          ENDIF
          IF (J.EQ.1) THEN
            WRITE (7,200) X,QANS,ERR1,IERR1,IERR
 200        FORMAT(D11.4,2D15.5,2I4)
          ELSE
            WRITE (7,201) QANS,ERR1,IERR1,IERR
 201        FORMAT(11X,2D15.5,2I4)
          ENDIF
 50     CONTINUE
 10   CONTINUE
 5    CONTINUE
 3    CONTINUE
      END
      DOUBLE PRECISION FUNCTION DQFUN(T)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION EN(1)
      COMMON /CQFUN/ AX,NU,KTEST
      N=NU
      IF (KTEST.EQ.1) THEN
        KODE=1
        MM=1
        TOL=0.5D-15
        ARG=AX*T
        CALL DEXINT(ARG, N, KODE, MM, TOL, EN, NZ, IERR)
        DQFUN=EN(1)/(T**N)
      ELSE
        DQFUN=DEXP(-AX*T)/(T**N)
      ENDIF
```

```
      RETURN
      END
      PROGRAM GHECOMP
C
C     COMPARES HALF ODD INTEGER ORDER ROUTINES DGHEXINT AND
C     DHEXINT AGAINST A QUADRATURE. OUTPUT IS WRITTEN TO FILE
C     GHECOMP.TXT
C-----------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION ENI(100),EN(100)
      COMMON /CQFUN/AX, FNU, KTEST
      EXTERNAL DQFUN
      OPEN(UNIT=7,FILE="GHECOMP.TXT")
      WRITE (7,110)
 110  FORMAT('     X           QUAD          RELERR1  IERR1 IERR2')
 111  CONTINUE
      PRINT *,'INPUT 1 TO TEST DGHEXINT; 2 TO TEST DHEXINT: '
      READ *,ICASE
      IF((ICASE.LT.1).OR.(ICASE.GT.2)) THEN
        GOTO 111
      ENDIF
      IF(ICASE.EQ.1) THEN
      PRINT *,'INPUT 1 TO TEST G SEQUENCE; INPUT 2 TO TEST E SEQUENCE: '
      READ *, ITEST
      ELSE
        ITEST=2
      ENDIF
      PRINT *,'INPUT STARTING VALUE OF ORDER (DP HALF ODD INTEGER > 0):
     &'
      READ *, FNHI
      IF((ITEST.LT.1).OR.(ITEST.GT.2)) GOTO 111
      IF((FNHI-DBLE(FLOAT(INT(FNHI)))).NE.0.5D0) GOTO 111
      DO 3 KODE=1,2
        WRITE(7,308)KODE
 308    FORMAT(/'                         KODE = ',I3/)
      KTEST=ITEST
      FNH=FNHI
      DO 5 MM=1,5
        IF(ICASE.EQ.1) THEN
          IF(ITEST.EQ.1) THEN
            WRITE(7,305) MM,FNH,KODE
 305    FORMAT(/'TESTING G SEQ IN DGHEXINT',' MM = ',I3,' FNH = ',D10.3,
     &' KODE = ',I3)
          ELSE
            WRITE(7,306) MM,FNHI,KODE
 306    FORMAT(/'TESTING E SEQ IN DGHEXINT',' MM = ',I3,' FNH = ',D10.3,
     &' KODE = ',I3)
          ENDIF
        ELSE
          WRITE(7,307) MM,FNHI,KODE
 307    FORMAT(/'TESTING E SEQ IN DHEXINT',' MM = ',I3,' FNH = ',D10.3,
     &' KODE = ',I3)
        ENDIF
        NIX=50
        DO 10 IX=1,NIX
          X=DBLE(FLOAT(IX))/5.0D0
C         KODE IS DEFINED ABOVE
          REL=0.5D-15
          TOL=REL
          FNH=FNHI
          IF(ICASE.EQ.1) THEN
            CALL DGHEXINT(X, FNH, KODE, MM, TOL, ENI, EN, NZ, IERR)
          ELSE
            CALL DHEXINT(X, FNH, KODE, MM, TOL,  EN, NZ, IERR)
          ENDIF
          DEX=DEXP(X)
        DO 50 J=1,MM
```

```
          IF(ITEST.EQ.1) THEN
            Y=ENI(J)
          ELSE
            Y=EN(J)
          ENDIF
          INIT=0
          TOL=0.5D-14
          AX=X
          FNU=FNHI+DBLE(FLOAT(J-1))
          X1=1.0D0
          SIG=3.0D0/X
          CALL DQUAD8(DQFUN,INIT,X1,SIG,TOL,X2,QANS,IERR1)
C         CONTINUE QUADRATURE PAST X2 TO BE SURE THE TRUNCATION IS
C         ADEQUATE
          SIG=6.0D0/X
          CALL DQUAD8(DQFUN,INIT,X1,SIG,TOL,X2,QANS,IERR2)
          IF(KODE.EQ.2) QANS=QANS*DEX
          ERR=DABS(Y-QANS)
          IF (QANS.NE.0.0D0) THEN
            ERR1=ERR/DABS(QANS)
          ENDIF
          IF (J.EQ.1) THEN
            WRITE (7,200) X,QANS,ERR1,IERR1,IERR
 200        FORMAT(D11.4,2D15.5,2I4)
          ELSE
            WRITE (7,201) QANS,ERR1,IERR1,IERR
 201        FORMAT(11X,2D15.5,2I4)
          ENDIF
 50     CONTINUE
 10   CONTINUE
 5    CONTINUE
 3    CONTINUE
      END
      DOUBLE PRECISION FUNCTION DQFUN(T)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION EN(1)
      COMMON /CQFUN/ AX,FNU,KTEST
      FNH=FNU
      IF (KTEST.EQ.1) THEN
        KODE=1
        MM=1
        TOL=0.5D-15
        ARG=AX*T
        CALL DHEXINT(ARG, FNH, KODE, MM, TOL, EN, NZ, IERR)
        DQFUN=EN(1)/(T**FNH)
      ELSE
        DQFUN=DEXP(-AX*T)/(T**FNH)
      ENDIF
      RETURN
      END
      PROGRAM DFCOMP
C
C     COMPARES THE COMPUTATION OF
C         DFERF=  INTEGRAL ON [0,X] OF ERF(T)/T
C     WITH A QUADRATURE.  SEE FOLDER 16
C-----------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      EXTERNAL DQFUN
      OPEN(UNIT=7,FILE="DFCOMP.TXT")
      WRITE (7,110)
 110  FORMAT('     X           QUAD           RELERR1   IERR')
      DO 10 IX=1,1000
        X=DBLE(FLOAT(IX))/100.0D0
        REL=0.5D-15
        Y=DFERF(X,REL,IERR)
        X1=0.0D0
        X2=X
```

```
        TOL=0.5D-14
        CALL DGAUS8(DQFUN,X1,X2,TOL,QANS,IERR)
        ERR=DABS(Y-QANS)
        IF (QANS.NE.0.0D0) THEN
          ERR1=ERR/DABS(QANS)
        ENDIF
        WRITE (7,200) X,QANS,ERR1,IERR
 200    FORMAT(D11.4,2D15.5,I4)
 10   CONTINUE
      END
      DOUBLE PRECISION FUNCTION DQFUN(T)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      KODE=1
      NZ=0
      DQFUN=DRERF(T,KODE,NZ)/T
      RETURN
      END
      PROGRAM DGCOMP
C
C     COMPARES G FUNCTION OF FOLDER 16 WITH A QUADRATURE
C
C-----------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      EXTERNAL DQFUN
      OPEN(UNIT=7,FILE="DGCOMP.TXT")
      WRITE (7,110)
 110  FORMAT('     X           QUAD          RELERR1    IERR IERR1 IERR2
     &')
      DO 5 KODE=1,2
      WRITE(7,120)KODE
 120  FORMAT(/'            KODE = ',I5/)
      DO 10 IX=1,200
        X=DBLE(FLOAT(IX))/10.0D0
        REL=0.5D-15
        Y=DGERFC(X,KODE,REL,IERR)
        INIT=0
        TOL=REL*10.0D0
        X1=X
        SIG=3.0D0
        CALL DQUAD8(DQFUN,INIT,X1,SIG,TOL,X2,QANS,IERR1)
        SIG=6.0D0
        CALL DQUAD8(DQFUN,INIT,X1,SIG,TOL,X2,QANS,IERR2)
        IF(KODE.EQ.2) THEN
          QANS=QANS*DEXP(X*X)
        ENDIF
        ERR=DABS(Y-QANS)
        IF (QANS.NE.0.0D0) THEN
          ERR1=ERR/DABS(QANS)
        ENDIF
        WRITE (7,200) X,QANS,ERR1,IERR,IERR1,IERR2
 200    FORMAT(D11.4,2D15.5,3I5)
 10   CONTINUE
 5    CONTINUE
      END
      DOUBLE PRECISION FUNCTION DQFUN(T)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      KODE=2
      NZ=0
      DQFUN=DRERF(T,KODE,NZ)/T
      RETURN
      END
      PROGRAM HERFCOMP
C
C     DONALD E. AMOS, MODIFIED MAY,2005
C
C     COMPARES H23 SERIES OF FOLDER 23 AND THE ASYMPTOTIC EXPANSION
C     OF FOLDER 23 WITH A QUADRATURE ON
```

```
C
C     HANS=DHERFC(X)= INT ON (0,X) OF EXP(W*W)*ERFC(W).
C
C     COMPARISONS ARE WRITTEN TO FILE HERFCOMP.TXT IN TERMS OF RELATIVE
C     ERRORS
C-----------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      EXTERNAL DQHERFC
      OPEN(UNIT=7,FILE="HERFCOMP.TXT")
      WRITE(7,201)
 201  FORMAT(2X,'COMPARISON OF SERIES AND ASYMPTOTIC EXPANSION AGAINST A
     & QUADRATURE FOR H23'//
     &        '  ERRS =RELATIVE ERROR FOR THE SERIES',/
     &        '  ERRA =RELATIVE ERROR FOR THE ASYMPTOTIC EXPANSION',/
     &        '  IERR =1, A NORMAL RETURN FROM DGAUS8',/
     &        '  TRMSA=NUMBER OF TERMS OF THE ASYMPTOTIC EXPANSION'/)
      WRITE(7,200)
 200  FORMAT(7X,"X",11X,'QUAD',9X,'ERRS',9X,'ERRA',5X,'IERR',2X,'TRMSA')
      DO 10 IX=1,1000
        X=DBLE(FLOAT(IX))/10.0D0
        NSIG=INT(X/3.0D0)+1
        SIG=X/DBLE(FLOAT(NSIG))
        QSUM=0.0D0
        X2=0.0D0
        REL=0.5D-14
        DO 20 J=1,NSIG
          X1=X2
          X2=X2+SIG
          TOL=REL
          CALL DGAUS8(DQHERFC,X1,X2,TOL,QANS,IERR)
          QSUM=QSUM+QANS
 20     CONTINUE
        HANS=DHERFC(X)
        ERR1=DABS(HANS-QSUM)
        IF(QANS.NE.0.0D0) THEN
          ERR1=ERR1/DABS(QANS)
        ENDIF
        IF(X.GE.4.0D0) THEN
          HASY=DHASY(X,NTRMS)
          ERR2=DABS(HASY-QSUM)
          IF(QANS.NE.0.0D0) THEN
            ERR2=ERR2/DABS(QANS)
          ENDIF
        ENDIF
        IF(X.LT.4.0D0) THEN
          WRITE(7,100) X, QSUM, ERR1,IERR
 100      FORMAT(3D13.5,13X,I5)
        ELSE
          WRITE(7,101) X, QSUM, ERR1,ERR2,IERR,NTRMS
 101      FORMAT(4D13.5,2I5)
        ENDIF
 10   CONTINUE
      END
      DOUBLE PRECISION FUNCTION DQHERFC(W)
C     INTEGRAND FOR DGAUS8 IN DHERFCOMP
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      KODE=3
      NZ=0
      DQHERFC=DRERF(W,KODE,NZ)
      RETURN
      END
      DOUBLE PRECISION FUNCTION DHASY(X,NLAST)
C     ASYMPTOTIC EXPANSION FOR X TO INFINITY FOR H23
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DATA RTPI  /1.772453850905516D0/
      DATA EULER /5.77215664901532861D-01/
      DATA TOL   /0.5D-14/
```

```
      NLAST=0
      S0=EULER+2.0D0*DLOG(X+X)
C     K=1 TERM
      X2=X*X
      TM=-0.5D0/X2
      HN=1.5D0
      SS=TM
      AN=2.0D0
      DO 10 K=2,20
        TM=-TM*HN/X2
        TRM=TM/AN
        IF(DABS(TRM).LE.DABS(SS)*TOL) GOTO 20
        SS=SS+TRM
        HN=HN+1.0D0
        AN=AN+1.0D0
   10 CONTINUE
   20 CONTINUE
      NLAST=K
      DHASY=0.5D0*(S0-SS)/RTPI
      RETURN
      END
```

```
      PROGRAM RPRGINFO
C-----------------------------------------------------------------------
C     DONALD E. AMOS, MAY 1, 2002
C
C     THIS FILE CONTAINS RESEARCH PROGRAMS WHICH EVALUATE FUNCTIONS
C     BEING STUDIED. MOST FUNCTIONS ARE INTEGRALS WITH SERIES
C     REPRESENTATIONS AND ARE COMPARED TO A DIRECT QUADRATURE. FILES
C     BECKSUBS.FOR AND AMOSSUBS.FOR MUST BE LINKED TO THE COMPILED FILES
C     FOR EXECUTION.
C
C     USAGE:
C
C     EXTRACT ONE OF THE PROGRAMS BELOW ALONG WITH ANY SUCCEEDING
C     DOUBLE PRECISION FUNCTION(S) OR SUBROUTINES, COMPILE, LINK
C     COMPILED FILES BECKSUBS.FOR AND AMOSSUBS.FOR, AND THEN EXECUTE THE
C     LINKED FILES. OUTPUT IS WRITTEN TO A TEXT FILE (*.TXT) WITH THE
C     NAME OF THE PROGRAM BEING EXECUTED.
C
C     PROGRAMS:                    OUTPUT:                 FOLDER
C              PROGRAM I1COMPB            I1COMPB.TXT        1,2
C              PROGRAM I4COMP             I4COMP.TXT          8
C              PROGRAM J4COMP             J4COMP.TXT          8
C              PROGRAM ERFINT             ERFINT.TXT         12
C              PROGRAM I13COMP            I13COMP.TXT        13
C              PROGRAM I14COMP            I14COMP.TXT        14
C              PROGRAM I19COMP            I19COMP.TXT        19
C              PROGRAM I20COMP            I20COMP.TXT        20
C              PROGRAM I24COMP            I24COMP.TXT        24
C              PROGRAM J24COMP            J24COMP.TXT        24
C              PROGRAM V24COMP            V24COMP.TXT        24
C              PROGRAM I25COMP            I25COMP.TXT        25
C              PROGRAM I26COMP            I26COMP.TXT        26
C              PROGRAM I26ACOMP           I26ACOMP.TXT       26
C              PROGRAM DGSCOMP            DGSCOMP.TXT        16
C-----------------------------------------------------------------------
      END
      PROGRAM I1COMPB
C
C     DONALD E. AMOS   AUGUST, 2003
C
C     REF: FOLDER 1, FOLDER 2, SECTION 2.6 OF "INTEGRALS RELATED TO
C          HEAT CONDUCTION AND DIFFUSION"
C
C     I1COMPB COMPARES THE CLOSED FORM OF I1 OF FOLDERS 1 & 2
C     WITH A QUADRATURE FOR I1 FOR A.LE.B:
C
C          I1=INT ON (T,INF) OF EXP(-A^2*W^2)*ERF(B*W)/W^2
C
C     WHERE A.GE.0.0D0, B.GE.0.0D0, AND T.GT.0.0D0. THE CLOSED FORM IS
C     IMPLEMENTED IN DOUBLE PRECISION SUBROUTINE INTEGI1B WHERE THE SUM
C     FOR A.LE.B IN FOLDER 2 IS IMPLEMENTED.
C
C     PROGRAM I1COMPB IS THE SAME AS I1COMP, BUT RESTRICTED TO KODE=1
C     AND A.LE.B. SUBROUTINE INTEGI1B TAKES THE PLACE OF INTEGI1.

C     OUTPUT IS WRITTEN TO FILE I1COMPB.TXT
C----------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION EN(1)
      COMMON /CFI1/ CA2,CB,KKODE
      DATA RTPI       /1.772453850905516D0/
      EXTERNAL DQI1
      OPEN(UNIT=7,FILE="I1COMPB.TXT")
      WRITE (7,110)
 110  FORMAT('      A          B           T          QUAD           ERR
     *     INIT IERR2')
      KODE=1
```

```
      WRITE(7,115) KODE
 115  FORMAT(/,2X,'THE FOLLOWING COMPARISONS ARE FOR KODE = ',I3/)
      DO 10 IT=1,101,10
        T=DBLE(FLOAT(IT))/40.0D0
        DO 20 IA=1,101,10
          A=DBLE(FLOAT(IA-1))/40.0D0
          DO 30 IB=1,101,10
            B=DBLE(FLOAT(IB-1))/40.0D0
            IF(B.LT.A) GOTO 30
            REL=0.50D-14
            CALL INTEGI1B(A,B,T,KODE,REL,ANS1,IERR2)
            IF(B.EQ.0.0D0) THEN
              IF(KODE.EQ.1) THEN
                QANS=0.0D0
                GOTO 40
              ELSE
                IF(A.EQ.0.0D0) THEN
                  QANS=1.0D0/T
                  GOTO40
                ENDIF
              ENDIF
            ENDIF
            XXA=A
            IF (A.EQ.0.0D0) THEN
              IF(KODE.EQ.1) THEN
                BT=B*T
                IF(BT.GE.6.0D0) THEN
                  QANS=1.0D0/T
                ELSE
                  ERFB=DRERF(BT,KODE,NZ)
                  BT2=BT*BT
                  N=1
                  IKODE=1
                  M=1
                  TOL=REL
                  NZ=0
                  IERR=0
                  CALL DEXINT(BT2, N, IKODE, M, TOL, EN, NZ, IERR)
                  QANS=ERFB/T+B*EN(1)/RTPI
                  GOTO 40
                ENDIF
                GOTO 40
              ELSE
                XXA=B
              ENDIF
            ENDIF
C           SET COMMON VARIABLES
            CA2=A*A
            CB=B
            KKODE=KODE
            SIG=3.0D0/XXA
            X1=T
            QREL=0.50D-14
            INIT=0
            CALL DQUAD8(DQI1,INIT,X1,SIG,QREL,X2,QANS,IERR)
C           GET BETTER ESTIMATE OF THE TRUNCATION ERROR
C            SIG=SIG+SIG
            CALL DQUAD8(DQI1,INIT,X1,SIG,QREL,X2,QANS,IERR)
  40        CONTINUE
            ERR=DABS(QANS-ANS1)
            IF(QANS.NE.0.0D0) THEN
              ERR=ERR/DABS(QANS)
            ENDIF
            WRITE(7,100) A,B,T,QANS,ERR,INIT,IERR2
 100        FORMAT(3D12.4,D13.5,D12.4,2I5)
 101        FORMAT(3I5)
   30     CONTINUE
```

```
   20   CONTINUE
   10 CONTINUE
      END
      DOUBLE PRECISION FUNCTION DQI1(T)
C
C     DQI1 COMPUTES THE INTEGRAND FOR THE QUADRATURE COMPARISON
C     WITH THE CLOSED FORM OF I1
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/ CFI1/ CA2,CB,KKODE
      T2=T*T
      ARG=CB*T
      IF(KKODE.EQ.1) THEN
        ERFX=DRERF(ARG,KKODE,NZ)
        DQI1=ERFX*DEXP(-CA2*T2)/T2
      ELSE
        IKODE=3
        ERFX=DRERF(ARG,IKODE,NZ)
        ARG=(CA2+CB*CB)*T2
        DQI1=ERFX*DEXP(-ARG)/T2
      ENDIF
      RETURN
      END
      SUBROUTINE INTEGI1B(A,B,T,KODE,REL,ANS1,IERR2)
C
C     INTEGI1B COMPUTES I1 FROM THE FORMULA OF FOLDER2 FOR A.LE.B WHERE
C     SQRT(t)=1/T
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION EN(1)
      DATA RTPI                  /1.772453850905516D0/
      IF(B.EQ.0.0D0) THEN
        ANS1=0.0D0
        RETURN
      ENDIF
      A2B2=A*A+B*B
      ARG=A2B2*T*T
      N=1
      IKODE=1
      M=1
      TOL=REL
      NZ=0
      IERR=0
      CALL DEXINT(ARG, N, IKODE, M, TOL, EN, NZ, IERR)
      X=A*T
      CALL DIERFC(X,IKODE,ANS,IERR)
      SS=RTPI*ANS/T+B*EN(1)/RTPI
      X=T
      CALL SER26(A,B,X,SUM,IERR)
      ANS1=SS-SUM
      RETURN
      END
      SUBROUTINE SER26(A,B,X,SUM,IERR)
C
C     SER26 COMPUTES THE SUM OF FOLDER 2 FOR A.LE.B.
C
C     IERR=0 NORMAL RETURN
C     IERR=1 UNDERFLOW, SUM=0.0D0
C     IERR=2 NO CONVERGENCE IN 50 TERMS OF THE SERIES
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION EN(52)
      DATA RTPI                  /1.772453850905516D0/
      IERR=0
      ARG=(A*A+B*B)
      DEN=ARG
      ARG=ARG*X*X
```

```
      FNH=0.5D0
      KODE=1
      M=52
      TOL=1.0D-14
      NZ=0
      KERR=0
      CALL DHEXINT(ARG, FNH, KODE, M, TOL, EN, NZ, KERR)
      IF(NZ.NE.0) THEN
        IERR=1
        SUM=0.0D0
        RETURN
      ENDIF
      AK=1.0D0
      S=EN(1)
      DH=-0.5D0
      Z=A*A/DEN
      FAC=1.0D0
      DO 10 K=1,M-1
        AK=AK*DH/FAC
        AK=AK*Z
        TRM=AK*EN(K+1)
        S=S+TRM
        IF (DABS(TRM).LE.TOL*DABS(S)) GOTO 15
        DH=DH+1.0D0
        FAC=FAC+1.0D0
  10  CONTINUE
       IERR=2
C      WRITE (7,300)
C 300  FORMAT('SER26-SERIES NOT CONVERGED IN 52 TERMS')
  15  CONTINUE
      SUM=DSQRT(DEN)*S/RTPI
      RETURN
      END
      PROGRAM I4COMP
C
C     ON KODE=1, I4COMP COMPUTES
C
C     I4=INT(T,INF) OF W*W*EXP(-C*C*W*W)*ERF(A*W)*ERF(B*W)
C
C     AND ON KODE=2, I4COMP COMPUTES
C
C     I4C=INT(T,INF) OF W*W*EXP(-C*C*W*W)*ERFC(A*W)*ERFC(B*W)
C
C                       A.GE.0,  B.GE.0,  C.GT.0
C
C     BY SERIES AND QUADRATURE DEVELOPED IN FOLDERS 8a AND 8b AND
C     COMPARES THE RESULTS.
C
C     OUTPUT IS WRITTEN TO FILE I4COMP.TXT
C------------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DATA RTPI /1.772453850905516D0/
      OPEN(UNIT=7,FILE="I4COMP.TXT")
C
      DO 5 KODE=1,2
C
      WRITE(7,105)KODE
 105  FORMAT('KODE = ',I2)
      WRITE (7,110)
 110  FORMAT('      A          B          C          T         QUAD
     &REL ERR IERR IERR4')
      DO 10 IT=1,9,2
        T=DBLE(FLOAT(IT-1))/1.0D0
        DO 20 IA=1,11,2
          A=DBLE(FLOAT(IA-1))/2.0D0
          DO 30 IB=1,11,2
            B=DBLE(FLOAT(IB-1))/2.0D0
```

```
            DO 40 IC=1,11,2
              C=DBLE(FLOAT(IC))/2.0D0
              CALL I4SER(A,B,C,T,KODE,ANS4,IERR4)
              CALL I4QUAD(A,B,C,T,KODE,QANS4,IERR)
              ERR=DABS(ANS4-QANS4)
              IF (QANS4.NE.0.0D0) THEN
                ERR=ERR/DABS(QANS4)
              ENDIF
              WRITE (7,200) A,B,C,T,QANS4,ERR,IERR,IERR4
 200          FORMAT(4D11.4,2D11.4,2I4)
   40       CONTINUE
   30     CONTINUE
   20   CONTINUE
   10 CONTINUE
   5  CONTINUE
      END
      SUBROUTINE I4SER(A,B,C,T,KODE,ANSI4,IERR)
C     I4SER EVALUATES THE SERIES FORM OF I4=INT (T,INF) OF
C     W*W*EXP(-C*C*W*W)*ERF(A*W)*ERF(B*W), DEVELOPED IN FOLDER 8
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DATA RTPI /1.772453850905516D0/
      IERR=0
      AT=A*T
      NZ=0
      ERFA=DRERF(AT,KODE,NZ)
      BT=B*T
      NZ=0
      ERFB=DRERF(BT,KODE,NZ)
      A2=A*A
      B2=B*B
      C2=C*C
      X2=A2+B2+C2
      X=DSQRT(X2)
      XT=X*T
      KKODE=2
      NZ=0
      ERFC=DRERF(XT,KKODE,NZ)
      SAC2=A2+C2
      SBC2=B2+C2
      CT=C*T
      CT2=CT*CT
      AT2=AT*AT
      BT2=BT*BT
      ECT2=DEXP(-CT2)
      S=T*ERFA*ERFB*ECT2
      TERM=(A*ERFB*ECT2*DEXP(-AT2)/SAC2+B*ERFA*ECT2*DEXP(-BT2)/SBC2)/
     *RTPI
      IF(KODE.EQ.1) THEN
        S=S+TERM
      ELSE
        S=S-TERM
      ENDIF
      PAB=A*B
      S=S+(PAB/SAC2+PAB/SBC2)*ERFC/(X*RTPI)
      CALL INTEGI3(A,B,C,T,KODE,ANSI3,IERR,KFORM)
      S=S+ANSI3
      TC2=C2+C2
      ANSI4=S/TC2
      RETURN
      END
      SUBROUTINE I4QUAD(A,B,C,T,KODE,ANS,IERR)
C     I4 BY QUADRATURE
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON /CFI4/AA,BB,CC,CC2,KKODE
      DATA RTPI /1.772453850905516D0/
      DATA TOL /0.5D-14/
```

```
      EXTERNAL DQI4
      IERR=1
      AA=A
      BB=B
      CC=C
      C2=C*C
      CC2=C2
      KKODE=KODE
      IF(KODE.EQ.1) THEN
C       REFINED QUADRATURE OF LIKE THAT OF FOLDERS 3 AND 6
        XAB=MIN(A,B)
        IF(XAB.EQ.0.0D0) THEN
          ANS=0.0D0
          RETURN
        ENDIF
        WSM=MIN(6.0D0/A,6.0D0/B)
        WLM=MAX(6.0D0/A,6.0D0/B)
        SS=0.0D0
        IF (T.LE.WSM) THEN
          W2=T
          SIGC=3.0D0/C
          SIG=MIN(SIGC,0.5D0*WSM)
          LN=50
          IFLAG=0
          DO 42 L=1,LN
            W1=W2
            W2=W1+SIG
            IF(W2.GT.WSM) THEN
              W2=WSM
              IFLAG=1
C             TEST FOR SMALL ERROR DUE TO A SMALL INTERVAL TO AVOID AN
C             ERROR FLAG FROM DGAUS8
              WMID=0.5D0*(W1+W2)
              DTEMP=DQI4(WMID)*DABS(W2-W1)
              IF(DTEMP.LT.2.0D0*TOL*DABS(SS)) GOTO 43
            ENDIF
            REL=TOL
            CALL DGAUS8(DQI4,W1,W2,REL,QANS,IERR)
            SS=SS+QANS
            IF(DABS(QANS).LT.TOL*DABS(SS)) GOTO 43
            IF(IFLAG.EQ.1) GOTO 43
   42     CONTINUE
   43     CONTINUE
          SS=SS+DQPI4(C,XAB,WSM,WLM)
        ELSE
          IF (T.LE.WLM)  THEN
            SS=DQPI4(C,XAB,T,WLM)
          ELSE
            C2=C*C
            ARG1=C2*T*T
            TC=C+C
            TC2=C2+C2
            KKODE=2
            NZ=0
            ARG2=C*T
            ERFD=DRERF(ARG2,KKODE,NZ)
            SS=(T*DEXP(-ARG1)+RTPI*ERFD/TC)/TC2
          ENDIF
        ENDIF
        ANS=SS
        RETURN
      ELSE
        INIT=0
        X1=T
        SIG=3.0D0/DSQRT(A*A+B*B+C*C)
        REL=TOL
        CALL DQUAD8(DQI4,INIT,X1,SIG,REL,X2,QANS,IERR)
```

```
          ANS=QANS
        ENDIF
        END
        DOUBLE PRECISION FUNCTION DQI4(T)
        IMPLICIT DOUBLE PRECISION (A-H,O-Z)
        COMMON/ CFI4/ AA,BB,CC,C2,KKODE
        ARG=BB*T
        ERFB=DRERF(ARG,KKODE,NZ)
        ARG=AA*T
        KODE=1
        ERFA=DRERF(ARG,KKODE,NZ)
        T2=T*T
        ARG=C2*T2
        DQI4=T2*ERFA*ERFB*DEXP(-ARG)
        RETURN
        END
        DOUBLE PRECISION FUNCTION DQPI4(C,X,FL,FU)
        IMPLICIT DOUBLE PRECISION (A-H,O-Z)
        DATA RTPI/1.77245385090552D0/
C       X=MIN(A,B) OF FOLDER8
        T1=X*FL
        T2=X*FU
        TC=C+C
        KODE=1
        NZ=0
        ERFA=DRERF(T1,KODE,NZ)
        NZ=0
        KODE=2
        ERFB=DRERF(T2,KODE,NZ)
        C2=C*C
        TC2=C2+C2
        ARG1=C2*FL*FL
        ARG2=C2*FU*FU
        S=FL*DEXP(-ARG1)*ERFA+FU*DEXP(-ARG2)*ERFB
        X2=X*X
        SXC2=X2+C2
        ARG1=SXC2*FL*FL
        ARG2=SXC2*FU*FU
        S=S+((DEXP(-ARG1)-DEXP(-ARG2))*(X/SXC2))/RTPI
        CALL INTEGJ5(C,X,FL,ANS1,IERR)
        CALL INTEGI5(C,X,FU,ANS2,IERR)
        S=S+ANS1+ANS2
        DQPI4=S/TC2
        RETURN
        END
        PROGRAM J4COMP
C
C       ON KODE=1, J4COMP COMPUTES
C
C       J4=INT(T,INF) OF W*EXP(-C*C*W*W)*ERF(A*W)*ERF(B*W)
C
C       AND ON KODE=2, J4COMP COMPUTES
C
C       J4C=INT(T,INF) OF W*EXP(-C*C*W*W)*ERFC(A*W)*ERFC(B*W)
C
C                         A.GE.0,  B.GE.0, C.GT.0
C
C       BY SERIES AND QUADRATURE DEVELOPED IN FOLDERS 8c AND 8d AND
C       COMPARES THE RESULTS.
C
C       OUTPUT IS WRITTEN TO FILE J4COMP.TXT
C----------------------------------------------------------------------
        IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C       DATA RTPI /1.772453850905516D0/
        OPEN(UNIT=7,FILE="J4COMP.TXT")
C
        DO 5 KODE=1,2
```

```
C
      WRITE(7,105) KODE
 105  FORMAT('KODE = ',I2)
      WRITE (7,110)
 110  FORMAT('      A          B          C          T        QUAD
     &REL ERR  IERR IERR4')
      DO 10 IT=1,11,2
        T=DBLE(FLOAT(IT-1))/5.0D0
        DO 20 IA=1,11,2
          A=DBLE(FLOAT(IA-1))/2.0D0
          DO 30 IB=1,11,2
            B=DBLE(FLOAT(IB-1))/2.0D0
            DO 40 IC=1,11,2
              C=DBLE(FLOAT(IC))/2.0D0
              CALL J4SER(A,B,C,T,KODE,ANS4,IERR4)
              CALL J4QUAD(A,B,C,T,KODE,QANS4,IERR)
              ERR=DABS(ANS4-QANS4)
              IF (QANS4.NE.0.0D0) THEN
                ERR=ERR/DABS(QANS4)
              ENDIF
              WRITE (7,200) A,B,C,T,QANS4,ERR,IERR,IERR4
 200          FORMAT(4D11.4,2D11.4,2I4)
   40       CONTINUE
   30     CONTINUE
   20   CONTINUE
   10 CONTINUE
   5  CONTINUE
      END
      SUBROUTINE J4SER(A,B,C,T,KODE,ANSJ4,IERR)
C     J4SER EVALUATES THE SERIES FORM OF J4=INT (T,INF) OF
C              W*EXP(-C*C*W*W)*ERF(A*W)*ERF(B*W)   ON KODE=1
C     OR
C              W*EXP(-C*C*W*W)*ERFC(A*W)*ERFC(B*W) ON KODE=2
C     DEVELOPED IN FOLDER 8C
C-----------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DATA RTPI /1.772453850905516D0/
      IERR=0
      AT=A*T
      NZ=0
      ERFA=DRERF(AT,KODE,NZ)
      BT=B*T
      NZ=0
      ERFB=DRERF(BT,KODE,NZ)
      A2=A*A
      B2=B*B
      C2=C*C
      CT=C*T
      CT2=CT*CT
      ECT2=DEXP(-CT2)
      S=0.5D0*ERFA*ERFB*ECT2
      RTB=DSQRT(B2+C2)
      RTA=DSQRT(A2+C2)
      IF(KODE.EQ.1) THEN
        CALL INTEGJ5(RTB,A,T,SUMB,IERR)
        CALL INTEGJ5(RTA,B,T,SUMA,IERR)
        S=S+(B*SUMB+A*SUMA)/RTPI
      ELSE
        CALL INTEGI5(RTB,A,T,SUMB,IERR)
        CALL INTEGI5(RTA,B,T,SUMA,IERR)
        S=S-(B*SUMB+A*SUMA)/RTPI
      ENDIF
      ANSJ4=S/C2
      RETURN
      END
      SUBROUTINE J4QUAD(A,B,C,T,KODE,ANS,IERR)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
```

```
      COMMON /CFJ4/AA,BB,CC,CC2,KKODE
      EXTERNAL DQJ4
C       DATA RTPI /1.772453850905516D0/
      DATA TOL /0.5D-14/
C     REFINED QUADRATURE LIKE THAT OF FOLDERS 3 AND 6
      IERR=1
      AA=A
      BB=B
      CC=C
      C2=C*C
      CC2=C2
      KKODE=KODE
      IF(KODE.EQ.1) THEN
        XAB=MIN(A,B)
        IF(XAB.EQ.0.0D0) THEN
          ANS=0.0D0
          RETURN
        ENDIF
        WSM=MIN(6.0D0/A,6.0D0/B)
        WLM=MAX(6.0D0/A,6.0D0/B)
        SS=0.0D0
        TC2=C2+C2
        IF (T.LE.WSM) THEN
          W2=T
          SIGC=3.0D0/C
          SIG=MIN(SIGC,0.5D0*WSM)
          LN=50
          IFLAG=0
          DO 42 L=1,LN
            W1=W2
            W2=W1+SIG
            IF(W2.GT.WSM) THEN
              W2=WSM
              IFLAG=1
C             TEST FOR SMALL ERROR DUE TO A SMALL INTERVAL TO AVOID AN
C             ERROR FLAG FROM DGAUS8
              WMID=0.5D0*(W1+W2)
              DTEMP=DQJ4(WMID)*DABS(W2-W1)
              IF(DTEMP.LT.2.0D0*TOL*DABS(SS)) GOTO 43
            ENDIF
            REL=TOL
            CALL DGAUS8(DQJ4,W1,W2,REL,QANS,IERR)
            SS=SS+QANS
            IF(IFLAG.EQ.1) GOTO 43
            IF(DABS(QANS).LT.TOL*DABS(SS)) GOTO 43
   42     CONTINUE
   43     CONTINUE
          SS=SS+DQPJ4(C,XAB,WSM,WLM)
        ELSE
          IF (T.LE.WLM)  THEN
            SS=DQPJ4(C,XAB,T,WLM)
          ELSE
            ARG1=C2*T*T
            SS=DEXP(-ARG1)/TC2
          ENDIF
        ENDIF
        ANS=SS
        RETURN
      ELSE
        INIT=0
        X1=T
        SIG=3.0D0/DSQRT(A*A+B*B+C*C)
        REL=TOL
        CALL DQUAD8(DQJ4,INIT,X1,SIG,REL,X2,ANS,IERR)
      ENDIF
      END
      DOUBLE PRECISION FUNCTION DQPJ4(C,X,FL,FU)
```

```
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C      DATA RTPI/1.77245385090552D0/
C     X=MIN(A,B) OF FOLDER8
      T1=X*FL
      T2=X*FU
      X2=X*X
      KODE=1
      NZ=0
      ERFA=DRERF(T1,KODE,NZ)
      NZ=0
      KODE=2
      ERFB=DRERF(T2,KODE,NZ)
      C2=C*C
      TC2=C2+C2
      ARG1=C2*FL*FL
      ARG2=C2*FU*FU
      S=DEXP(-ARG1)*ERFA+DEXP(-ARG2)*ERFB
      RT=DSQRT(C2+X2)
      T1=FL*RT
      KODE=2
      NZ=0
      ERFCL=DRERF(T1,KODE,NZ)
      T2=FU*RT
      KODE=2
      NZ=0
      ERFCU=DRERF(T2,KODE,NZ)
      S=S+X*(ERFCL-ERFCU)/RT
      DQPJ4=S/TC2
      RETURN
      END
      DOUBLE PRECISION FUNCTION DQJ4(T)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/ CFJ4/ AA,BB,CC,C2,KKODE
      ARG=BB*T
      ERFB=DRERF(ARG,KKODE,NZ)
      ARG=AA*T
      ERFA=DRERF(ARG,KKODE,NZ)
      T2=T*T
      ARG=C2*T2
      DQJ4=T*ERFA*ERFB*DEXP(-ARG)
      RETURN
      END
      PROGRAM ERFINT
C
C     DONALD E. AMOS   JULY, 2001
C
C     REF: FOLDER 12
C          INTEGRAL  X*EXP((A^2-B^2)*X)*ERFC(A*DX+C/DX), DX=SQRT(X)
C          BECK E-MAIL DATED 7/12/01
C
C     OUTPUT IS WRITTEN TO FILE ERFINT.TXT
C----------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON /CFI3/ CA,CA2,CB2,CC
      EXTERNAL DGFUN
      OPEN(UNIT=7,FILE="ERFINT.TXT")
      X1=1.772D0
      X2=4.14159D0
      WRITE (7,109) X1,X2
 109  FORMAT(' X1 = ',D12.4,'  X2 = ',D12.4)
      WRITE (7,110)
 110  FORMAT('      A         B         C        QANS       RELERR   IER
     *R')
      DO 10 IA=1,10,2
        A=DBLE(FLOAT(IA))/2.00D0
        DO 20 IB=1,11,2
          B=DBLE(FLOAT(IB))/2.0D0
```

```
          IF (B.EQ.A) GOTO 20
          DO 30 IC=1,5
            C=DBLE(FLOAT(IC))/1.0D0
            CALL ERFINTX(X2,A,B,C,GANS2)
            CALL ERFINTX(X1,A,B,C,GANS1)
            GANS=GANS2-GANS1
C           SET COMMON PARAMETERS FOR DGFUN
            CA=A
            CA2=A*A
            CB2=B*B
            CC=C
            REL=0.50D-12
            CALL DGAUS8(DGFUN, X1, X2, REL, QANS, IERR)
            ERR=DABS(GANS-QANS)
            IF (QANS.NE.0.0D0) THEN
              ERR=ERR/DABS(QANS)
            ENDIF
            WRITE (7,200) A,B,C,QANS,ERR,IERR
 200        FORMAT(3D10.3,2D12.4,I4)
   30     CONTINUE
   20   CONTINUE
   10 CONTINUE
      END
      DOUBLE PRECISION FUNCTION DGFUN(T)
C
C     DGFUN COMPUTES THE INTEGRAND T*EXP((A^2-B^2)*T)*ERFC(A*RT+C/RT)
C     WHERE RT=SQRT(T)
C
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      COMMON/ CFI3/ CA,CA2,CB2,CC
      DT=DSQRT(T)
      ARG=CA*DT+CC/DT
      KODE=2
      Y=DRERF(ARG,KODE,NZ)
      DGFUN=T*Y*DEXP((CA2-CB2)*T)
      RETURN
      END
      SUBROUTINE ERFINTX(X,A,B,C,ANS)
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      A2=A*A
      B2=B*B
      A2MB2=A2-B2
      T1=-DEXP(A2MB2*X)/A2MB2
      T2=-X+1.0D0/A2MB2
      KODE=2
      DX=DSQRT(X)
      ARG=A*DX+C/DX
      Y=DRERF(ARG,KODE,NZ)
      TRM1=T1*T2*Y
      CALL FN(DX,A,B,C,FANS1)
      CALL FN(DX,-A,B,-C,FANS2)
      ARG=-2.0D0*A*C
      ANS=TRM1+0.5D0*DEXP(ARG)*(FANS1-FANS2)
      RETURN
      END
      SUBROUTINE FN(DX,A,B,C,FANS)
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      DATA RTPI       /1.772453850905516D0/
      ARG=2.0D0*B*C
      A2MB2=A*A-B*B
      T1=DEXP(ARG)/A2MB2
      ARG=B*DX+C/DX
      KODE=2
      Y=DRERF(ARG,KODE,NZ)
      T2=(-DX/B)*(1.0D0+A/B)/RTPI
      T3=(C/B+1.0D0/A2MB2)*(1.0D0+A/B)-0.5D0*(A/B)/(B*B)
      FANS=T1*(T2*DEXP(-ARG*ARG)+T3*Y)
```

```
      RETURN
      END
      PROGRAM I13COMP
C
C     DONALD E. AMOS   DECEMBER, 2001; DECEMBER, 2005
C
C     REF: FOLDER 13, FOLDER 10, FOLDER 11
C
C     ON KODE=1, I13COMP COMPUTES THE INTEGRAL
C
C          I13=INT ON (T,INF) OF EXP(-A^2*W^2)ERF(B*W)/(W**3);
C
C     ON KODE=2 I13COMP COMPUTES THE COMPLEMENTARY INTEGRAL
C
C         I13C=INT ON (T,INF) OF EXP(-A^2*W^2)ERFC(B*W)/(W**3)
C
C     AND EACH INTEGRAL IS COMPARED WITH ITS RESPECTIVE QUADRATURE,
C     WHERE     A.GT.0.0D0, B.GE.0.0D0, AND T.GT.0.0D0.
C
C     OUTPUT IS WRITTEN TO FILE I13COMP.TXT
C-------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON /CFI13/ CA2,CB,KKODE
      EXTERNAL DQI13
      OPEN(UNIT=7,FILE="I13COMP.TXT")
      DO 5 KODE=1,2
        KKODE=KODE
        WRITE(7,109) KODE
 109    FORMAT(/,27X,"KODE = ",I1)
        WRITE (7,110)
 110    FORMAT('       A          B           T          QUAD       REL
     &ERR    IERR')
        DO 10 IT=1,25,2
          T=DBLE(FLOAT(IT))/5.0D0
          DO 20 IA=1,21,2
            A=DBLE(FLOAT(IA))/10.0D0
            A2=A*A
            CA2=A2
            DO 30 IB=1,21,2
              B=DBLE(FLOAT(IB-1))/10.0D0
              CB=B
              B2=B*B
              CALL INTEGI13(A,B,T,KODE,ANS13)
              INIT=0
              X1=T
              SIG=3.0D0/A
              IF(KODE.EQ.2) THEN
                SIG=3.0D0/DSQRT(A2+B2)
              ENDIF
              REL=0.5D-15
              CALL DQUAD8(DQI13,INIT,X1,SIG,REL,X2,QANS,IERR)
              ERR=DABS(QANS-ANS13)
              IF(QANS.NE.0.0D0) THEN
                ERR=ERR/DABS(QANS)
              ENDIF
              WRITE(7,100) A,B,T,QANS,ERR,IERR
 100          FORMAT(3D12.4,D13.5,D12.4,I4)
   30       CONTINUE
   20     CONTINUE
   10   CONTINUE
    5 CONTINUE
      END
      DOUBLE PRECISION FUNCTION DQI13(T)
C
C     DQI13 COMPUTES THE INTEGRAND FOR THE QUADRATURE COMPARISON
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
```

```
      COMMON/ CFI13/ CA2,CB,KKODE
      T2=T*T
      ARG1=CA2*T2
      NZ=0
      ARG=CB*T
      ERFB=DRERF(ARG,KKODE,NZ)
      DQI13=DEXP(-ARG1)*ERFB/(T2*T)
      RETURN
      END
      SUBROUTINE INTEGI13(A,B,T,KODE,ANS13)
C
C     REF: FOLDER 13
C
C     ON KODE=1, INTEGI13 COMPUTES THE INTEGRAL
C
C          ANS13=INT ON (T,INF) OF EXP(-A^2*W^2)*ERF(B*W)/(W**3);
C
C     ON KODE=2 INTEGI13 COMPUTES THE COMPLEMENTARY INTEGRAL
C
C          ANS13=INT ON (T,INF) OF EXP(-A^2*W^2)*ERFC(B*W)/(W**3)
C
C     WHERE    A.GT.0.0D0, B.GE.0.0D0, AND T.GT.0.0D0.
C
C     CALLS ROUTINES: DRERF, DIERFC, DEXINT, INTEGP, INTEGQ
C-----------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION EN(1)
      DATA RTPI       /1.772453850905516D0/
      T2=T*T
      A2=A*A
      B2=B*B
      NZ=0
      REL=0.5D-14
      IF(KODE.EQ.1) THEN
C       KODE=1 EVALUATES FORMULAE FOR I13
C       IFORM=1 AND 2 EVALUATE 2 DIFFERENT FORMULAS FOR I13
C       IFORM=2 SEEMS TO GIVE BETTER OVERALL RESULTS
        IFORM=2
        LKODE=1
        ARG=B*T
        ERFB=DRERF(ARG,LKODE,NZ)
        X=T*DSQRT(A2+B2)
        LKODE=1
        CALL DIERFC(X,LKODE,ANS,IERR)
        IF(IFORM.EQ.1) THEN
          ARG=A2*T2
          S=0.5D0*ERFB*DEXP(-ARG)/T2+B*ANS/T
          TOL=REL
          KODE=1
          CALL INTEGP(A,B,T,KODE,TOL,PANS,IERR)
          ANS13=S-A2*PANS
        ELSE
          X=A2*T2
          N=2
          LKODE=1
          M=1
          TOL=REL
          CALL DEXINT(X, N, LKODE, M, TOL, EN, NZ, IERR)
          TOL=REL
          CALL INTEGQ(B,A,T,TOL,QANS,IERR)
          ANS13=0.5D0*ERFB*EN(1)/T2+B*ANS/T-A2*B*QANS/RTPI
        ENDIF
      ELSE
C       KODE=2 EVALUATES FORMULAE FOR I13C
C       IFORM=1 AND 2 EVALUATE 2 DIFFERENT FORMULAS FOR I13C
C       IFORM=2 SEEMS TO GIVE BETTER OVERALL RESULTS
        IFORM=2
```

```
        LKODE=2
        ARG=B*T
        ERFB=DRERF(ARG,LKODE,NZ)
        X=T*DSQRT(A2+B2)
        LKODE=1
        CALL DIERFC(X,LKODE,ANS,IERR)
        X=A2*T2
        LKODE=1
        M=1
        TOL=REL
        CALL DEXINT(X, IFORM, LKODE, M, TOL, EN, NZ, IERR)
        IF(IFORM.EQ.1) THEN
          ARG=A2*T2
          S=0.5D0*ERFB*DEXP(-ARG)/T2-0.5D0*A2*EN(1)-B*ANS/T
          TOL=REL
          KODE=1
          CALL INTEGP(A,B,T,KODE,TOL,PANS,IERR)
          ANS13=S+A2*PANS
        ELSE
          TOL=REL
          CALL INTEGQ(B,A,T,TOL,QANS,IERR)
          ANS13=0.5D0*EN(1)*ERFB/T2-B*ANS/T+A2*B*QANS/RTPI
        ENDIF
      ENDIF
      RETURN
      END
      PROGRAM I14COMP
C
C     DONALD E. AMOS   DECEMBER, 2001; DECEMBER, 2005
C
C     REF: FOLDER 14
C
C     ON KODE=1, I14COMP COMPUTES THE INTEGRAL
C
C       I14=INT ON (T,INF) OF EXP(-C^2*W^2)*ERF(A*W)*ERF(B*W)/(W**2);
C
C     ON KODE=2 I14COMP COMPUTES  THE INTEGRAL
C
C       I14C=INT ON (T,INF) OF EXP(-C^2*W^2)*ERFC(A*W)*ERFC(B*W)/(W**2)
C
C     IN SUBROUTINE INTEGI14 AND EACH INTEGRAL IS COMPARED WITH ITS
C     RESPECTIVE QUADRATURE WHERE
C
C             A.GE.0.0D0, B.GE.0.0D0, C.GT.0.0D0 AND T.GT.0.0D0.
C
C     OUTPUT IS WRITTEN TO FILE I14COMP.TXT
C-----------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON /CFI14/ AA,BB,CC,C2,KKODE
      EXTERNAL DQI14
      OPEN(UNIT=7,FILE="I14COMP.TXT")
      DO 5 KODE=1,1
        KKODE=KODE
        WRITE(7,105) KODE
 105    FORMAT(/' KODE = ',I2)
        WRITE (7,110)
 110    FORMAT('      A         B          C          T         QUAD
     &  REL ERR  IERR')
        DO 10 IT=1,25,2
          T=DBLE(FLOAT(IT))/5.0D0
          DO 20 IA=1,21,2
            A=DBLE(FLOAT(IA-1))/10.0D0
            DO 30 IB=1,21,2
              B=DBLE(FLOAT(IB-1))/10.0D0
              DO 40 IC=1,21,2
                C=DBLE(FLOAT(IC))/10.0D0
                CALL INTEGI14(A,B,C,T,KODE,ANS14,I3ERR)
```

```
                IF((I3ERR.NE.0)) THEN
                  WRITE (7,405) A,B,C,T,I3ERR
 405              FORMAT( 4D11.4,1X,'IN INTEGI14, I3ERR=',I1)
                  GOTO 40
                ENDIF
                JD=50
                X2=T
                IF(KODE.EQ.1) THEN
                  XA=MIN(A,B)
                  IF(XA.EQ.0.0D0) THEN
                    QANS=0.0D0
                    IERR=1
                    GOTO 55
                  ENDIF
                  WSM=MIN(6.0D0/A,6.0D0/B)
                  WBM=MAX(6.0D0/A,6.0D0/B)
                ELSE
                  SIG=3.0D0/DSQRT(A*A+B*B+C*C)
                ENDIF
                AA=A
                BB=B
                CC=C
                C2=C*C
                SIGC=3.0D0/C
                QANS=0.0D0
                REL=0.50D-14
                DO 50 J=1,JD
                  X1=X2
                  IF(KODE.EQ.1) THEN
                    IF(X1.LE.WSM) THEN
                      SIG=MIN(SIGC,0.5D0*WSM)
                    ELSE
                      IF(X1.LE.WBM) THEN
                        SIG=MIN(SIGC,0.5D0*WBM)
                      ENDIF
                    ENDIF
                  ENDIF
                  X2=X1+SIG
                  CALL DGAUS8(DQI14,X1,X2,REL,ANS,IERR)
                  QANS=QANS+ANS
                  IF (DABS(ANS).LE.REL*DABS(QANS)) GOTO 55
  50            CONTINUE
                WRITE (7,300)
 300  FORMAT('NO CONVERGENCE FROM DGAUS8 LOOP IN I14COMP AFTER 50 STEPS'
     &)
  55            CONTINUE
                ERR=DABS(ANS14-QANS)
                IF (QANS.NE.0.0D0) THEN
                  ERR=ERR/DABS(QANS)
                ENDIF
                WRITE (7,200) A,B,C,T,QANS,ERR,IERR
 200            FORMAT(4D11.4,D12.5,D12.4,I4)
  40          CONTINUE
  30        CONTINUE
  20      CONTINUE
  10    CONTINUE
   5  CONTINUE
      END
      DOUBLE PRECISION FUNCTION DQI14(T)
C
C     DQI14 COMPUTES THE INTEGRAND FOR THE QUADRATURE COMPARISON
C     OF I14
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/ CFI14/ AA,BB,CC,C2,KKODE
      ARG=BB*T
      NZ=0
```

```
      ERFB=DRERF(ARG,KKODE,NZ)/T
      ARG=AA*T
      ERFA=DRERF(ARG,KKODE,NZ)/T
      T2=T*T
      ARG=C2*T2
      DQI14=ERFA*ERFB*DEXP(-ARG)
      RETURN
      END
      SUBROUTINE INTEGI14(A,B,C,T,KODE,ANS14,IERR14)
C
C     REF: FOLDER 14
C
C     ON KODE=1, I14COMP COMPUTES THE INTEGRAL
C
C       I14=INT ON (T,INF) OF EXP(-C^2*W^2)*ERF(A*W)*ERF(B*W)/(W*W);
C
C     ON KODE=2 I14COMP COMPUTES  THE INTEGRAL
C
C       I14C=INT ON (T,INF) OF EXP(-C^2*W^2)*ERFC(A*W)*ERFC(B*W)/(W*W)
C
C     WHERE    A.GE.0.0D0, B.GE.0.0D0, C.GT.0.0D0, AND T.GT.0.0D0.
C
C     CALLS ROUTINES: INTEGI1,INTEGI2,INTEGI3,INTEGP,INTEGI13
C-----------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DATA RTPI/ 1.772453850905516D0/
      IERR14=0
      KFORM=0
      REL=0.5D-14
      XA=MIN(A,B)
      XB=MAX(A,B)
      AT=XA*T
      BT=XB*T
      IF(KODE.EQ.1) THEN
        IF(XA.EQ.0.0D0) THEN
          ANS14=0.0D0
          RETURN
        ENDIF
        IF(BT.GT.6.0D0) THEN
          IF(AT.GT.6.0D0) THEN
            KKODE=1
            X=C*T
            CALL DIERFC(X,KKODE,ANS,IERR)
            ANS14=RTPI*ANS/T
            RETURN
          ELSE
            CALL INTEGI1(C,XA,T,KODE,REL,ANS14,IERR)
            RETURN
          ENDIF
        ENDIF
      ELSE
        IF(AT.EQ.0.0D0) THEN
          TOL=REL
C         ON KODE=2 AND ONE OF A OR B EQUAL TO ZERO, I14C BECOMES I1C OF
C         FOLDERC 1,2, AND 10
          CALL INTEGI1(C,XB,T,KODE,TOL,ANS14,IERR)
          RETURN
        ENDIF
      ENDIF
      A2=A*A
      B2=B*B
      C2=C*C
      T2=T*T
      CALL INTEGI3(A,B,C,T,KODE,ANSI3,IERR,KFORM)
      RTA2C2=DSQRT(A2+C2)
      RTB2C2=DSQRT(B2+C2)
      RTA2B2=DSQRT(A2+B2)
```

```
      IF(KODE.EQ.1) THEN
        SS=ANSI3
        TOL=REL
        CALL INTEGP(RTA2C2,B,T,KODE,TOL,PANS,IERR)
        SS=SS+PANS/(A*RTPI)
        TOL=REL
        CALL INTEGP(RTB2C2,A,T,KODE,TOL,PANS,IERR)
        SS=SS+PANS/(B*RTPI)
        TOL=REL
        CALL INTEGP(C,RTA2B2,T,KODE,TOL,PANS,IERR)
        SS=SS-RTA2B2*PANS/(A*B*RTPI)
        CALL INTEGI2(A,B,T,KODE,ANSI2,IERR)
        STEMP=DEXP(-C2*T2)*ANSI2/T2-2.0D0*C2*SS
        CALL INTEGI13(RTA2C2,B,T,KODE,CU)
        SS=CU/(A*RTPI)
        CALL INTEGI13(RTB2C2,A,T,KODE,CU)
        SS=SS+CU/(B*RTPI)
        CALL INTEGI13(C,RTA2B2,T,KODE,CU)
        SS=SS-RTA2B2*CU/(A*B*RTPI)
        ANS14=STEMP-2.0D0*SS
      ELSE
        SS=-ANSI3
        TOL=REL
        CALL INTEGP(RTA2C2,B,T,KODE,TOL,PANS,IERR)
        SS=SS+PANS/(A*RTPI)
        TOL=REL
        CALL INTEGP(RTB2C2,A,T,KODE,TOL,PANS,IERR)
        SS=SS+PANS/(B*RTPI)
        TOL=REL
        CALL INTEGP(C,RTA2B2,T,KODE,TOL,PANS,IERR)
        SS=SS-RTA2B2*PANS/(A*B*RTPI)
        CALL INTEGI2(A,B,T,KODE,ANSI2,IERR)
        STEMP=-DEXP(-C2*T2)*ANSI2/T2+2.0D0*C2*SS
        CALL INTEGI13(RTA2C2,B,T,KODE,CU)
        SS=CU/(A*RTPI)
        CALL INTEGI13(RTB2C2,A,T,KODE,CU)
        SS=SS+CU/(B*RTPI)
        CALL INTEGI13(C,RTA2B2,T,KODE,CU)
        SS=SS-RTA2B2*CU/(A*B*RTPI)
        ANS14=STEMP+2.0D0*SS
      ENDIF
      RETURN
      END
      SUBROUTINE INTEGI13(A,B,T,KODE,ANS13)
C
C     REF: FOLDER 13
C
C     ON KODE=1, INTEGI13 COMPUTES THE INTEGRAL
C
C          ANS13=INT ON (T,INF) OF EXP(-A^2*W^2)*ERF(B*W)/(W**3);
C
C     ON KODE=2 INTEGI13 COMPUTES THE COMPLEMENTARY INTEGRAL
C
C          ANS13=INT ON (T,INF) OF EXP(-A^2*W^2)*ERFC(B*W)/(W**3)
C
C     WHERE    A.GT.0.0D0, B.GE.0.0D0, AND T.GT.0.0D0.
C
C     CALLS ROUTINES: DRERF, DIERFC, DEXINT, INTEGP, INTEGQ
C-----------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION EN(1)
      DATA RTPI       /1.772453850905516D0/
      T2=T*T
      A2=A*A
      B2=B*B
      NZ=0
      REL=0.5D-14
```

```
      IF(KODE.EQ.1) THEN
C       KODE=1 EVALUATES FORMULAE FOR I13
C       IFORM=1 AND 2 EVALUATE 2 DIFFERENT FORMULAS FOR I13
C       IFORM=2 SEEMS TO GIVE BETTER OVERALL RESULTS
        IFORM=2
        LKODE=1
        ARG=B*T
        ERFB=DRERF(ARG,LKODE,NZ)
        X=T*DSQRT(A2+B2)
        LKODE=1
        CALL DIERFC(X,LKODE,ANS,IERR)
        IF(IFORM.EQ.1) THEN
          ARG=A2*T2
          S=0.5D0*ERFB*DEXP(-ARG)/T2+B*ANS/T
          TOL=REL
          KODE=1
          CALL INTEGP(A,B,T,KODE,TOL,PANS,IERR)
          ANS13=S-A2*PANS
        ELSE
          X=A2*T2
          N=2
          LKODE=1
          M=1
          TOL=REL
          CALL DEXINT(X, N, LKODE, M, TOL, EN, NZ, IERR)
          TOL=REL
          CALL INTEGQ(B,A,T,TOL,QANS,IERR)
          ANS13=0.5D0*ERFB*EN(1)/T2+B*ANS/T-A2*B*QANS/RTPI
        ENDIF
      ELSE
C       KODE=2 EVALUATES FORMULAE FOR I13C
C       IFORM=1 AND 2 EVALUATE 2 DIFFERENT FORMULAS FOR I13C
C       IFORM=2 SEEMS TO GIVE BETTER OVERALL RESULTS
        IFORM=2
        LKODE=2
        ARG=B*T
        ERFB=DRERF(ARG,LKODE,NZ)
        X=T*DSQRT(A2+B2)
        LKODE=1
        CALL DIERFC(X,LKODE,ANS,IERR)
        X=A2*T2
        LKODE=1
        M=1
        TOL=REL
        CALL DEXINT(X, IFORM, LKODE, M, TOL, EN, NZ, IERR)
        IF(IFORM.EQ.1) THEN
          ARG=A2*T2
          S=0.5D0*ERFB*DEXP(-ARG)/T2-0.5D0*A2*EN(1)-B*ANS/T
          TOL=REL
          KODE=1
          CALL INTEGP(A,B,T,KODE,TOL,PANS,IERR)
          ANS13=S+A2*PANS
        ELSE
          TOL=REL
          CALL INTEGQ(B,A,T,TOL,QANS,IERR)
          ANS13=0.5D0*EN(1)*ERFB/T2-B*ANS/T+A2*B*QANS/RTPI
        ENDIF
      ENDIF
      RETURN
      END
      PROGRAM I19COMP
C
C     DONALD E. AMOS    APRIL, 2002 ; JUNE, 2003
C
C     REFERENCE: FOLDER19
C----------------------------------------------------------------------
C     A DOUBLE PRECISION ROUTINE
```

```
C-----------------------------------------------------------------------
C
C     ON KODE=1, I19COMP COMPUTES THE INTEGRAL
C
C            INT ON (0,T) OF ERF(A*W)*ERF(B*W)/W,      A>0, B>0, T>0
C
C     AND ON KODE=2, I19COMP COMPUTES THE INTEGRAL
C
C            INT ON (T,INF) OF ERFC(A*W)*ERFC(B*W)/W,  A>0, B>0, T>0
C
C     BY QUADRATURE (DQUAD8) AND BY SERIES (INTEGI19) AND PRINTS THE
C     RELATIVE ERROR.
C-----------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON /QFUN/ CA,CB,KKODE
      EXTERNAL QFUNI19
      OPEN(UNIT=7,FILE="I19COMP.TXT")
      KODE=2
      WRITE(7,150) KODE
 150  FORMAT('KODE = ',I3)
      WRITE(7,200)
 200  FORMAT('     A          B          T         QUAD       SERIES
     C  REL ERR   IERR')
      DO 10 IA=1,21,2
         A=DBLE(FLOAT(IA))/10.0D0
        DO 20 IB=1,21,2
          B=DBLE(FLOAT(IB))/10.0D0
          DO 30 IT=1,41,2
            T=0.0D0+0.2D0*DBLE(FLOAT(IT))
            CALL INTEGI19(A,B,T,KODE,ANS,JERR)
            CA=A
            CB=B
            KKODE=KODE
            IF(KODE.EQ.1) THEN
              X1=0.0D0
              X2=T
              REL=0.5D-14
              CALL DGAUS8(QFUNI19,X1,X2,REL,QANS,IERR)
            ELSE
              INIT=0
              X1=T
              SIG=1.50D0
              REL=0.5D-14
              CALL DQUAD8(QFUNI19,INIT,X1,SIG,REL,X2,QANS,IERR)
C             CALL DQUAD8 AGAIN TO BE SURE THE TRUCATION IS ADEQUATE
              SIG=3.0D0
              REL=0.5D-14
              CALL DQUAD8(QFUNI19,INIT,X1,SIG,REL,X2,QANS,IERR)
            ENDIF
            ERR=DABS(ANS-QANS)
            IF(QANS.NE.0.0D0) THEN
              ERR=ERR/DABS(QANS)
            ENDIF
            WRITE(7,100) A,B,T,QANS,ANS,ERR,IERR
  100       FORMAT(3D11.4,3D12.5,I3)
  30      CONTINUE
  20    CONTINUE
  10  CONTINUE
      END
      SUBROUTINE INTEGI19(A,B,T,KODE,ANS,IERR)
C----------------------------------------------------------------------
C     REFERENCE: FOLDER19
C
C     ON KODE=1, I19COMP COMPUTES THE INTEGRAL
C
C            I19 =INT ON (0,T) OF ERF(A*W)*ERF(B*W)/W
C
```

```
C      AND ON KODE=2, I19COMP COMPUTES THE COMPLEMENTARY INTEGRAL
C
C             I19C=INT ON (T,INF) OF ERFC(A*W)*ERFC(B*W)/W
C
C      WHERE   A>0  ,  B>0  , T>0.
C-----------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION EGH(75),ENH(75),C(75),RTMP(2)
      DATA PI, RTPI/3.14159265358979324D0, 1.77245385090551603D0/
      IERR=0
      IF((KODE.NE.1).AND.(KODE.NE.2)) THEN
        IERR=5
        ANS=0.0D0
        RETURN
      ENDIF
      REL=0.5D-15
C     CHOOSING AA TO BE MIN AND BB TO B MAX OF A AND B
      IF(A.LE.B) THEN
        AA=A
        BB=B
      ELSE
        AA=B
        BB=A
      ENDIF
      AT=AA*T
      BT=BB*T
      KKODE=1
      GERFBT=DGERFC(BT,KKODE,REL,IERRG)
      IF(IERRG.NE.0) THEN
        ANS=0.0D0
        RETURN
      ENDIF
      ERFAT=DRERF(AT,KODE,NZ)
      IF(NZ.NE.0) THEN
        ANS=0.0D0
        RETURN
      ENDIF
      SS=ERFAT*GERFBT
C     HERE SS IS ERF(AT)*G(BT) ON KODE=1 OR ERFC(AT)*G(BT) ON KODE=2
C     COMPUTE THE REGULAR PART OF I19C
      DO 5 I=1,3-KODE
        IF(I.EQ.1) THEN
          TD=T
        ELSE
          TD=0.0D0
        ENDIF
        A2=AA*AA
        B2=BB*BB
        A2PB2=A2+B2
        R1=A2/A2PB2
        R2=B2/A2PB2
        X=TD*TD*A2PB2
        RTAB=DSQRT(A2PB2)
        CALL INTEGI5(AA,BB,TD,ANSI5,IERR5)
        TRMLN=AA*DLOG(R2)*ANSI5/RTPI
        FNH=1.5D0
        KKODE=1
        MM=60
        TOL=REL
        CALL DGHEXINT(X, FNH, KKODE, MM, TOL, EGH, ENH, NZ, IERR)
        IF(NZ.NE.0) THEN
          ANS=0.0D0
          RETURN
        ENDIF
C       COMBIME THE TERMS OF THE TWO SERIES. THE POWERS OF R1 REDUCE
C       THE EFFECT OF TERMS WHERE LOSSES OF SIGNIFICANCE OCCUR IN THE
C       SUBTRACTION
```

```
        PWR=R1
        C(1)=1.0D0
        AK=0.50D0
        BK=1.0D0
        S=0.0D0
        TOL=R1*REL/10.0D0
        DO 10 K=2,60
          CK=C(K-1)*AK/BK
          T1=CK*EGH(K)
          C(K)=CK
          CSUM=0.0D0
          AM=1.0D0
          DO 20 M=1,K-1
            CSUM=CSUM+C(K-M)/AM
            AM=AM+1.0D0
 20       CONTINUE
          T2=ENH(K)*CSUM
          S=S+PWR*(T2-T1)
          IF(DABS(PWR).LE.TOL) GOTO 30
          PWR=PWR*R1
          AK=AK+1.0D0
          BK=BK+1.0D0
 10       CONTINUE
          WRITE(7,190)
 190      FORMAT('IN INTEGI19, DROP THROUGH THE SERIES LOOP')
          PAUSE
  30    CONTINUE
        S=S-EGH(1)
C       RTMP(1) CONTAINS THE EVALUATION OF R AT T
C       RTMP(2) CONTAINS THE EVALUATION OF R AT T=0.0D0
        RTMP(I)=TRMLN+0.5D0*AA*S/(PI*RTAB)
 5    CONTINUE
      IF(KODE.EQ.2) THEN
        ANS=SS+RTMP(1)
      ELSE
        FAT=DFERF(AT,REL,KERR)
        ANS=FAT+RTMP(2)+SS-RTMP(1)
      ENDIF
      RETURN
      END
      DOUBLE PRECISION FUNCTION QFUNI19(W)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON /QFUN/ CA,CB,KKODE
      ARG=CA*W
      ERFAT=DRERF(ARG,KKODE,NZ)
      ARG=CB*W
      ERFBT=DRERF(ARG,KKODE,NZ)
      QFUNI19=ERFAT*ERFBT/W
      RETURN
      END
      PROGRAM I20COMP
C
C     DONALD E. AMOS    APRIL, 2002: JUNE, 2003
C-----------------------------------------------------------------------
C     A DOUBLE PRECISION ROUTINE
C-----------------------------------------------------------------------
C
C     REFERENCE: FOLDER 20
C
C     ON KODE=1, I20COMP COMPUTES THE INTEGRAL
C
C        I20 =  INT ON (T,INF) OF EXP(-A^2*X^2)*ERF(B*X)*LN(X)
C
C     ON KODE=2, I20COMP COMPUTES THE COMPLEMENTARY INTEGRAL
C
C        I20C=  INT ON (T,INF) OF EXP(-A^2*X^2)*ERFC(B*X)*LN(X)
C
```

```
C                     A > 0, B > 0, T > 0
C
C     BY QUADRATURE (DQUAD8) AND BY SERIES (INTEGI20) AND PRINTS THE
C     RELATIVE ERROR IN FILE I20COMP.TXT. SUBROUTINE INTEGI20 USES
C     SUBROUTINE INTEGI19.
C-----------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON /QFUN/ CA,CB,KKODE
      EXTERNAL QFUNI20
      OPEN(UNIT=7,FILE="I20COMP.TXT")
      KODE=2
      WRITE(7,150) KODE
 150  FORMAT(' KODE = ',I3)
      WRITE(7,200)
 200  FORMAT('     A          B          T         QUAD       SERIES
     C  REL ERR   IERR')
      DO 10 IA=1,21,2
        A=DBLE(FLOAT(IA))/10.0D0
        DO 20 IB=1,21,2
          B=DBLE(FLOAT(IB))/10.0D0
          DO 30 IT=1,41,2
            T=0.0D0+0.2D0*DBLE(FLOAT(IT))
            CALL INTEGI20(A,B,T,KODE,ANS,JERR)
            CA=A
            CB=B
            KKODE=KODE
            INIT=0
            X1=T
            SIG=1.50D0
            REL=0.5D-14
            CALL DQUAD8(QFUNI20,INIT,X1,SIG,REL,X2,QANS,IERR)
C           CALL DQUAD8 AGAIN TO BE SURE THE TRUCATION IS ADEQUATE
            SIG=3.0D0
            REL=0.5D-14
            CALL DQUAD8(QFUNI20,INIT,X1,SIG,REL,X2,QANS,IERR)
            ERR=DABS(ANS-QANS)
            IF(QANS.NE.0.0D0) THEN
              ERR=ERR/DABS(QANS)
            ENDIF
            WRITE(7,100) A,B,T,QANS,ANS,ERR,IERR
  100       FORMAT(3D11.4,3D12.5,I3)
  30      CONTINUE
  20    CONTINUE
  10  CONTINUE
      END
      SUBROUTINE INTEGI20(A,B,T,KODE,ANS,IERR)
C-----------------------------------------------------------------------
C     A DOUBLE PRECISION ROUTINE
C-----------------------------------------------------------------------
C
C     REFERENCE: FOLDER20
C
C     ON KODE=1, INTEGI20 COMPUTES THE INTEGRAL
C
C        ANS = INT ON (T,INF) OF EXP(-A^2*X^2)*ERF(B*X)*LN(X)
C
C     ON KODE=2, INTEGI20 COMPUTES THE COMPLEMENTARY INTEGRAL
C
C        ANS= INT ON (T,INF) OF EXP(-A^2*X^2)*ERFC(B*X)*LN(X)
C
C                       A > 0, B > 0, T > 0
C
C     INTEGI20 USES SUBROUTINE INTEGI19
C-----------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION EN(60)
      DATA RTPI/1.77245385090551603D0/
```

```
      IERR=0
      IF((KODE.NE.1).AND.(KODE.NE.2)) THEN
        ANS=0.0D0
        IERR=5
        RETURN
      ENDIF
      REL=0.5D-15
C     AA= MIN(A,B)= FIRST ARGUMENT OF I20C,
C     BB= MAX(A,B)= SECOND ARGUMENT OF I20C
      IF(A.LE.B) THEN
        AA=A
        BB=B
        IFLAG=0
      ELSE
        AA=B
        BB=A
        IFLAG=1
      ENDIF
C     COMPUTE I20C FOR I20C(AA,BB,T)
      CALL INTEGI5(AA,BB,T,ANSI5,IERR5)
      SS=DLOG(T)*ANSI5
      A2=AA*AA
      B2=BB*BB
      A2PB2=A2+B2
      R1=A2/A2PB2
C      R2=B2/A2PB2
      X=T*T*A2PB2
      N=1
      KKODE=1
      MM=1
      TOL=REL
      CALL DEXINT(X, N, KKODE, MM, TOL, EN, NZ, IERR)
      SS=SS+0.5D0*EN(1)*DATAN(AA/BB)/(AA*RTPI)
C     COMPUTE S SERIES
      FNH=0.5D0
      KKODE=1
      MM=60
      TOL=0.5D-15
      NZ=0
      KERR=0
      CALL DHEXINT(X, FNH, KKODE, MM, TOL, EN, NZ, KERR)
      IF(NZ.NE.0) THEN
        ANS=0.0D0
        IERR=1
        RETURN
      ENDIF
      AK=1.0D0
      S=2.0D0*EN(2)
      DH=0.5D0
      FAC=1.0D0
      M=59
      DO 10 K=2,M
        AK=AK*DH/FAC
        AK=AK*R1
        TRM=AK*EN(K+1)/(DH+1.0D0)
        S=S+TRM
        IF (DABS(TRM).LE.TOL*DABS(S)) GOTO 15
        DH=DH+1.0D0
        FAC=FAC+1.0D0
   10 CONTINUE
       IERR=2
C      WRITE (7,300)
C 300  FORMAT('S-SERIES NOT CONVERGED IN 55 TERMS')
   15 CONTINUE
      SS=SS-0.25D0*S/(RTPI*DSQRT(A2PB2))
      IF(IFLAG.EQ.0) THEN
        ANSI20C=SS
```

```
      ELSE
        KKODE=2
        CALL INTEGI19(BB,AA,T,KKODE,ANS19,IERR)
        ARG=AA*T
        ERFAT=DRERF(ARG,KKODE,NZ)
        ARG=BB*T
        ERFBT=DRERF(ARG,KKODE,NZ)
        S=0.5D0*(ANS19+DLOG(T)*ERFAT*ERFBT)*RTPI/BB
        ANSI20C=S-(AA/BB)*SS
      ENDIF
C     I20C = ANSI20C HERE
      IF(KODE.EQ.1) THEN
        ARG=A*T
        KKODE=2
        ERFAT=DRERF(ARG,KKODE,NZ)
        KKODE=1
        REL=0.5D-15
        TRM=RTPI*(ERFAT*DLOG(T)+DGERFC(ARG,KKODE,REL,IERR))/(A+A)
        ANS=TRM-ANSI20C
      ELSE
        ANS=ANSI20C
      ENDIF
      RETURN
      END
      SUBROUTINE INTEGI19(A,B,T,KODE,ANS,IERR)
C-----------------------------------------------------------------------
C     REFERENCE: FOLDER19
C
C     ON KODE=1, I19COMP COMPUTES THE INTEGRAL
C
C            I19 =INT ON (0,T) OF ERF(A*W)*ERF(B*W)/W
C
C     AND ON KODE=2, I19COMP COMPUTES THE COMPLEMENTARY INTEGRAL
C
C            I19C=INT ON (T,INF) OF ERFC(A*W)*ERFC(B*W)/W
C
C     WHERE   A>0  ,  B>0  , T>0.
C-----------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION EGH(75),ENH(75),C(75),RTMP(2)
      DATA PI, RTPI/3.14159265358979324D0, 1.77245385090551603D0/
      IERR=0
      IF((KODE.NE.1).AND.(KODE.NE.2)) THEN
        IERR=5
        ANS=0.0D0
        RETURN
      ENDIF
      REL=0.5D-15
C     CHOOSING AA TO BE MIN AND BB TO B MAX OF A AND B
      IF(A.LE.B) THEN
        AA=A
        BB=B
      ELSE
        AA=B
        BB=A
      ENDIF
      AT=AA*T
      BT=BB*T
      KKODE=1
      GERFBT=DGERFC(BT,KKODE,REL,IERRG)
      IF(IERRG.NE.0) THEN
        ANS=0.0D0
        RETURN
      ENDIF
      ERFAT=DRERF(AT,KODE,NZ)
      IF(NZ.NE.0) THEN
        ANS=0.0D0
```

```
          RETURN
        ENDIF
        SS=ERFAT*GERFBT
C       HERE SS IS ERF(AT)*G(BT) ON KODE=1 OR ERFC(AT)*G(BT) ON KODE=2
C       COMPUTE THE REGULAR PART OF I19C
        DO 5 I=1,3-KODE
          IF(I.EQ.1) THEN
            TD=T
          ELSE
            TD=0.0D0
          ENDIF
          A2=AA*AA
          B2=BB*BB
          A2PB2=A2+B2
          R1=A2/A2PB2
          R2=B2/A2PB2
          X=TD*TD*A2PB2
          RTAB=DSQRT(A2PB2)
          CALL INTEGI5(AA,BB,TD,ANSI5,IERR5)
          TRMLN=AA*DLOG(R2)*ANSI5/RTPI
          FNH=1.5D0
          KKODE=1
          MM=60
          TOL=REL
          CALL DGHEXINT(X, FNH, KKODE, MM, TOL, EGH, ENH, NZ, IERR)
          IF(NZ.NE.0) THEN
            ANS=0.0D0
            RETURN
          ENDIF
C         COMBIME THE TERMS OF THE TWO SERIES. THE POWERS OF R1 REDUCE
C         THE EFFECT OF TERMS WHERE LOSSES OF SIGNIFICANCE OCCUR IN THE
C         SUBTRACTION
          PWR=R1
          C(1)=1.0D0
          AK=0.50D0
          BK=1.0D0
          S=0.0D0
          TOL=R1*REL/10.0D0
          DO 10 K=2,60
            CK=C(K-1)*AK/BK
            T1=CK*EGH(K)
            C(K)=CK
            CSUM=0.0D0
            AM=1.0D0
            DO 20 M=1,K-1
              CSUM=CSUM+C(K-M)/AM
              AM=AM+1.0D0
 20         CONTINUE
            T2=ENH(K)*CSUM
            S=S+PWR*(T2-T1)
            IF(DABS(PWR).LE.TOL) GOTO 30
            PWR=PWR*R1
            AK=AK+1.0D0
            BK=BK+1.0D0
 10         CONTINUE
            WRITE(7,190)
 190        FORMAT('IN INTEGI19, DROP THROUGH THE SERIES LOOP')
            PAUSE
  30      CONTINUE
          S=S-EGH(1)
C         RTMP(1) CONTAINS THE EVALUATION OF R AT T
C         RTMP(2) CONTAINS THE EVALUATION OF R AT T=0.0D0
          RTMP(I)=TRMLN+0.5D0*AA*S/(PI*RTAB)
 5      CONTINUE
        IF(KODE.EQ.2) THEN
          ANS=SS+RTMP(1)
        ELSE
```

```
        FAT=DFERF(AT,REL,KERR)
        ANS=FAT+RTMP(2)+SS-RTMP(1)
      ENDIF
      RETURN
      END
      DOUBLE PRECISION FUNCTION QFUNI20(W)
C-----------------------------------------------------------------------
C     QFUNI20 COMPUTES THE INTEGRAND FOR I20 OR I20C FOR THE DQUAD8
C     EVALUATION OF THE INTEGRALS I20 OR I20C IN I20COMP.
C-----------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON /QFUN/ CA,CB,KKODE
      ARG=CB*W
      ERFBT=DRERF(ARG,KKODE,NZ)
      ARG=CA*CA*W*W
      QFUNI20=ERFBT*DLOG(W)*DEXP(-ARG)
      RETURN
      END
      PROGRAM I24COMP
C
C     DONALD E. AMOS   DECEMBER, 2002; MAY, 2006
C
C     REF: FOLDER 24
C
C     ON KODE=1, I24COMP COMPARES THE PROCEDURE FROM SUBROUTINE INTEGI24
C     AND DGAUS8 QUADRATURE EVALUATIONS OF
C
C          I24=INT ON (0,T) OF W*EXP(A^2*W+2*A*B)*ERFC(X)*ERF(C/RW)
C
C     ON KODE=2, I24COMP COMPARES THE PROCEDURE FROM SUBROUTINE INTEGI24
C     AND DGAUS8 QUADRATURE EVALUATIONS OF THE COMPLEMENTARY FUNCTION
C
C          I24=INT ON (0,T) OF W*EXP(A^2*W+2*A*B)*ERFC(X)*ERFC(C/RW)
C              WITH X=A*RW+B/RW AND RW=SQRT(W)
C
C     AND A, B AND C ARE POSITIVE.
C
C     OUTPUT IS WRITTEN TO FILE I24COMP.TXT
C----------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      OPEN(UNIT=7,FILE="I24COMP.TXT")
      WRITE (7,109)
 109  FORMAT('-----------------------------------------------------------
     &-----------------'/
     &'LEGENDS FOR KFORM:'//
     &'KFORM DIGIT=0 MEANS ONE OR MORE PARMETERS ARE ZERO--A SPECIAL FOR
     &MULA IS USED'/
     &'KFORM DIGIT=1 MEANS T.LE.(C*C/36) WHERE ERF(C/DSQRT(T))=1.0D0'/
     &'KFORM DIGIT=2 MEANS THE CLOSED FORM IS USED FOR VOFT'/
     &'KFORM DIGIT=3 MEANS THE LARGE L.GE.2 SERIES IS USED FOR W3'/
     &'KFORM DIGIT=4 MEANS THE A*SQRT(T).LT.1 SERIES IS USED FOR I24'/
     &'KFORM DIGIT=5 MEANS THE THE QUADRATURE IS USED FOR W3'//
     &'EXAMPLE:'/
     &'KFORM=12 MEANS A COMBINATION OF 1 AND 2 WAS USED'/
     &'-----------------------------------------------------------------
     &----------')
C
      DO 5 KODE=1,2
C
      WRITE (7,105) KODE
 105  FORMAT(/'THE FOLLOWING TABLE OF COMPARISONS IS FOR KODE = ',I3/)
      WRITE (7,110)
 110  FORMAT('      A         B         C          T         QUAD
     &REL ERR  IERR KFORM')
      DO 10 IT=1,11,2
        T=DBLE(FLOAT(IT))/10.0D0
        DO 20 IA=1,11,2
```

```
          A=DBLE(FLOAT(IA-1))/5.0D0
          DO 30 IB=1,11,2
            B=DBLE(FLOAT(IB-1))/2.0D0
            DO 40 IC=1,11,2
              C=DBLE(FLOAT(IC-1))/5.0D0
              CALL INTEGI24(A,B,C,T,KODE,ANS2,I2ERR,KFORM)
              IF((I2ERR.NE.0)) THEN
                WRITE (7,405) A,B,C,T,I2ERR,KFORM
 405            FORMAT( 4D11.4,1X,'IN INTEGI24, I2ERR=',I2,' KFORM=',I2)
                GOTO 40
              ENDIF
C-----------------------------------------------------------------------
C     LEGENDS FOR KFORM:
C
C     KFORM DIGIT=1 MEANS T.LE.(C*C/36) WHERE ERF(C/DSQRT(T))=1.0D0
C     KFORM DIGIT=2 MEANS THE CLOSED FORM IS USED FOR VOFT
C     KFORM DIGIT=3 MEANS THE LARGE L.GE.2 SERIES IS USED FOR W3
C     KFORM DIGIT=4 MEANS THE A*SQRT(T).LT.1 SERIES IS USED FOR I24
C     KFORM DIGIT=5 MEANS THE THE QUADRATURE IS USED FOR W3
C
C     EXAMPLE:
C     KFORM=12 MEANS A COMBINATION OF 1 AND 2 WAS USED
C-----------------------------------------------------------------------
              CALL I24QUAD(A,B,C,T,KODE,QANS,IERR)
              ERR=DABS(ANS2-QANS)
              IF (QANS.NE.0.0D0) THEN
                ERR=ERR/DABS(QANS)
              ENDIF
              WRITE (7,200) A,B,C,T,QANS,ERR,IERR,KFORM
 200          FORMAT(4D11.4,D12.5,D12.4,2I4)
   40       CONTINUE
   30     CONTINUE
   20   CONTINUE
   10 CONTINUE
   5  CONTINUE
      END
      DOUBLE PRECISION FUNCTION DQI24(X)
C
C     DQI24 COMPUTES THE INTEGRAND FOR THE QUADRATURE COMPARISON
C     OF I24 WITH INTEGI24
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/ CFI24/ CA,CB,CC,IKODE
C     A CHANGE OF VARIABLE T=X*X REMOVES THE SQRT FROM THE INTEGRAND
C     THE UPPER LIMIT ON THE INTEGRAL IS THEN DSQRT(T)
      ARG=CA*X+CB/X
      KODE=3
      NZ=0
      ERFX=DRERF(ARG,KODE,NZ)
      X2=X*X
      ARG=CB*CB/X2
      ERFX=ERFX*DEXP(-ARG)
      ARG=CC/X
      NZ=0
      ERFA=DRERF(ARG,IKODE,NZ)
      DQI24=2.0D0*X*X2*ERFX*ERFA
      RETURN
      END
      SUBROUTINE I24QUAD(A,B,C,T,KODE,QANS24,IERR)
C     QUADRATURE FOR THE I24 INTEGRAL. A CHANGE OF VARIABLES TAU=X*X
C     IS USED. THE UPPER LIMIT ON THE X INTEGRAL IS SQRT(T)=RT.
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/ CFI24/ CA,CB,CC,IKODE
      DATA TOL /0.5D-14/
      EXTERNAL DQI24
      IERR=0
      IF((C.EQ.0.0D0).AND.(KODE.EQ.1)) THEN
```

```
          QANS24=0.0D0
          IERR=1
          RETURN
        ENDIF
        IF((A.EQ.0.0D0).AND.(B.EQ.0.0D0).AND.(C.EQ.0.0D0)) THEN
C         KODE=2 HERE BECAUSE C=0 AND THE PREVIOUS TEST FAILED
          QANS24=0.5D0*T*T
          IERR=1
          RETURN
        ENDIF
        IF(T.EQ.0.0D0) THEN
          QANS24=0.0D0
          IERR=1
          RETURN
        ENDIF
        IF(A.EQ.0.0D0) THEN
          IF(KODE.EQ.1) THEN
            XLIM=MIN(B,C)/6.0D0
          ELSE
            XLIM=DSQRT(B*B+C*C)/6.0D0
          ENDIF
        ELSE
          IF(B.EQ.0.0D0) THEN
            XLIM=C/6.0D0
          ELSE
            IF(KODE.EQ.1) THEN
              XLIM=MIN(B,C)/6.0D0
            ELSE
              XLIM=DSQRT(B*B+C*C)/6.0D0
            ENDIF
          ENDIF
        ENDIF
        CA=A
        CB=B
        CC=C
        IKODE=KODE
        RT=DSQRT(T)
        JD=10
        INIT=-JD
        X1=0.0D0
        REL=TOL
        SS=0.0D0
        IF (RT.LE.XLIM) THEN
          SIG=RT/DBLE(FLOAT(JD))
          CALL DQUAD8(DQI24,INIT,X1,SIG,REL,X2,QANS,IERR)
          SS=SS+QANS
        ELSE
          SIG=XLIM/DBLE(FLOAT(JD))
          CALL DQUAD8(DQI24,INIT,X1,SIG,REL,X2,QANS,IERR)
          SS=SS+QANS
          INIT=-JD
          X1=X2
          SIG=(RT-X2)/DBLE(FLOAT(JD))
C         TEST FOR RT TOO CLOSE TO X2 FOR AN ACCURATE QUADRATURE
C         (SEE THE ERROR MESSAGE AND TEST IN DGAUS8)
          XMID=X1+0.5D0*SIG
          FMID=DQI24(XMID)
          IF(DABS(FMID*SIG).LT.2.0D0*TOL*DABS(SS)) GOTO 20
          REL=TOL
          CALL DQUAD8(DQI24,INIT,X1,SIG,REL,X2,QANS,KERR)
          IF(KERR.NE.1) THEN
            IERR=KERR
          ENDIF
          SS=SS+QANS
   20     CONTINUE
        ENDIF
        QANS24=SS
```

```
      RETURN
      END
      SUBROUTINE INTEGI24(A,B,C,T,KODE,ANS24,I2ERR)
C-------------------------------------------------------------------
C     REFERENCE: FOLDER24
C
C     ON KODE=1, INTEGI24 COMPUTES THE INTEGRAL
C
C          I24=INT ON (0,T) OF W*U(A,B,W)*ERF(C/RW)
C
C      AND ON KODE=2 INTEGI24 COMPUTES THE INTEGRAL
C
C          I24=INT ON (0,T) OF W*U(A,B,W)*ERFC(C/RW)
C
C     WHERE U(A,B,W)=EXP(A*A*W+2*A*B)*ERFC(A*RW+B/RW), RW=SQRT(W)
C
C     AND A, B ARE NON-NGATIVE.
C-------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION EN(1),Y(60),YN(60)
      DATA RTPI /1.772453850905516D0/
      I2ERR=0
      RT=DSQRT(T)
      AS=A*A
      BS=B*B
      CS=C*C
      CAPT=1.0D0/RT
      REL=0.5D-14
      ART=A*RT
      IF(ART.LE.1.0D0) THEN
        N0=0
        M=50
        KKODE=1
        X=B*CAPT
        CALL DINERFC(X,N0,KKODE,M,REL,Y,NZ)
        CALL GNSEQ(C,B,CAPT,M,REL,YN)
        S1=Y(3)
        S2=Y(5)
        S3=YN(4)
        S4=YN(5)+YN(5)
        X=ART+ART
        PW=-X
        AK=3.0D0
        J1=0
        J2=0
        J3=0
        J4=0
        JSTOP=0
        KL=M-4
        DO 10 K=2,KL
          IF(J1.EQ.0) THEN
            TRM1=PW*Y(K+2)
            S1=S1+TRM1
            IF(DABS(TRM1).LT.REL*DABS(S1)) THEN
              JSTOP=JSTOP+1
              J1=1
            ENDIF
          ENDIF
          IF(J2.EQ.0) THEN
            TRM2=PW*Y(K+4)
            S2=S2+TRM2
            IF(DABS(TRM2).LT.REL*DABS(S2)) THEN
              JSTOP=JSTOP+1
              J2=1
            ENDIF
          ENDIF
          IF(J3.EQ.0) THEN
```

```
            TRM3=PW*YN(K+3)
            S3=S3+TRM3
            IF(DABS(TRM3).LT.REL*DABS(S3)) THEN
              JSTOP=JSTOP+1
              J3=1
            ENDIF
          ENDIF
          IF(J4.EQ.0) THEN
            TRM4=AK*PW*YN(K+4)
            S4=S4+TRM4
            IF(DABS(TRM4).LT.REL*DABS(S4)) THEN
              JSTOP=JSTOP+1
              J4=1
            ENDIF
          ENDIF
          IF((K.EQ.KL).OR.(JSTOP.EQ.4)) GOTO 20
          PW=-X*PW
          AK=AK+1.0D0
 10     CONTINUE
 20     CONTINUE
        ARG=C*CAPT
        ERFX=DRERF(ARG,KODE,NZ)
        SUM=(4.0D0*S1-16.0D0*S2)*T*T*ERFX
        X=(BS+CS)/T
        COEF=8.0D0*C*T*DEXP(-X)/RTPI
        IF(KODE.EQ.1) THEN
          ANS24=SUM+COEF*(B*S3+RT*S4)
        ELSE
          ANS24=SUM-COEF*(B*S3+RT*S4)
        ENDIF
        RETURN
      ENDIF
      VT=DVOFT(A,B,T,REL,IERR,KFORM)
      ARG=C*CAPT
      ERFX=DRERF(ARG,KODE,NZ)
      SUM=T*VT*ERFX
C     CALC T0=T0C
      X=(BS+CS)/T
      N=2
      KKODE=1
      M=1
      TOL=REL
      CALL DEXINT(X, N, KKODE, M, TOL, EN, NZ, IERR)
      T0=(T+T)*EN(1)/(A*RTPI)
      COEF=2.0D0*(1.0D0/AS+(B+B)/A)
      KKODE=2
      CALL INTEGI1(C,B,CAPT,KKODE,REL,ANSI1,IERR)
      T0=T0-COEF*ANSI1
      CALL INTEGI22(A,B,C,T,ANSI22,I22ERR,KFORM)
      T0=T0+ANSI22/AS
      TRM=T0*C/RTPI
      IF(KODE.EQ.1) THEN
        SUM=SUM+TRM
      ELSE
        SUM=SUM-TRM
      ENDIF
C     CALC T1 OR T1C
      ARG=C*CAPT
      ERFX=DRERF(ARG,KODE,NZ)
      ARG=BS/T
      TRM1=T*RT*DEXP(-ARG)*ERFX
      TRM2=C*T*EN(1)/RTPI
      CALL INTEGI1(B,C,CAPT,KODE,REL,ANSI1,IERR)
      TRM3=2.0D0*BS*ANSI1
      COE1=4.0D0/(3.0D0*A*RTPI)
      IF(KODE.EQ.1) THEN
        T1=COE1*(TRM1+TRM2-TRM3)
```

```
      ELSE
        T1=COE1*(TRM1-TRM2-TRM3)
      ENDIF
      SUM=SUM-T1
C     CALC T2 OR T2C
      IF(KODE.EQ.1) THEN
        IFORM=2
        IF(IFORM.EQ.1) THEN
C       IF(C*CAPT.LT.1.0D0) THEN
C         T2 USING J3 AND W3
          ARG=C*CAPT
          JKODE=1
          COE=C
        ELSE
C         T2 USING J3C AND W3C
          ARG=B*CAPT
          JKODE=2
          COE=-B
        ENDIF
        IF(IFORM.EQ.1) THEN
          ERFX=DRERF(ARG,JKODE,NZ)
          IKODE=1
          CALL DIERFC(ARG,IKODE,ANS,IERR)
          FJ3=0.5D0*ERFX*T+COE*ANS*RT
        ELSE
          N0=2
          M=1
          IKODE=1
          REL=0.5D-14
          CALL DINERFC(ARG,N0,IKODE,M,REL,Y,NZ)
          FJ3=2.0D0*T*Y(1)
        ENDIF
        CALL INTEGW3(B,C,CAPT,IFORM,REL,ANSW3,IERR)
        T2=COEF*(FJ3-ANSW3)
      ELSE
        CALL INTEGW3(B,C,CAPT,KODE,REL,ANSW3,IERR)
        T2=COEF*ANSW3
      ENDIF
      SUM=SUM+T2
C     CALC T3 OR T3C
      CALL INTEGI21(A,B,C,T,KODE,ANS21,I21ERR,KFORM)
      T3=ANS21/AS
      ANS24=SUM-T3
      RETURN
      END
      PROGRAM J24COMP
C
C     DONALD E. AMOS   DECEMBER, 2002; MAY, 2006
C
C     REF: FOLDER 24
C
C     J24COMP COMPARES THE PROCEDURE FROM SUBROUTINE INTEGJ24
C     AND DGAUS8 QUADRATURE EVALUATIONS OF
C
C          J24=INT ON (0,T) OF W*EXP(A^2*W+2*A*B)*ERFC(X)
C                      X=A*RT+B/RT, RT=SQRT(W)
C
C     AND A AND B ARE NON-NGATIVE.
C
C     OUTPUT IS WRITTEN TO FILE J24COMP.TXT
C-----------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      OPEN(UNIT=7,FILE="J24COMP.TXT")
C
      WRITE (7,110)
 110  FORMAT('      A          B           T          QUAD         REL ERR
     &IERR')
```

```
      DO 10 IT=1,11,2
        T=DBLE(FLOAT(IT))/5.0D0
        DO 20 IA=1,11,2
          A=DBLE(FLOAT(IA-1))/5.0D0
          DO 30 IB=1,11,2
            B=DBLE(FLOAT(IB-1))/2.0D0
            CALL INTEGJ24(A,B,T,ANS2,J2ERR)
            IF((J2ERR.NE.0)) THEN
              WRITE (7,405) A,B,T,J2ERR
 405          FORMAT( 3D11.4,1X,'IN INTEGJ24, J2ERR=',I2)
              GOTO 30
            ENDIF
            CALL J24QUAD(A,B,T,QANS,IERR)
            ERR=DABS(ANS2-QANS)
            IF (QANS.NE.0.0D0) THEN
              ERR=ERR/DABS(QANS)
            ENDIF
            WRITE (7,200) A,B,T,QANS,ERR,IERR
 200        FORMAT(3D11.4,D12.5,D12.4,I4)
   30     CONTINUE
   20   CONTINUE
   10 CONTINUE
      END
      DOUBLE PRECISION FUNCTION DQJ24(X)
C
C     DQJ24 COMPUTES THE INTEGRAND FOR THE QUADRATURE COMPARISON
C     OF J24 WITH INTEGJ24
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/ CFJ24/ CA,CB
C     A CHANGE OF VARIABLE TAU=X*X REMOVES THE SQRT FROM THE INTEGRAND
C     THE UPPER LIMIT ON THE INTEGRAL IS THEN DSQRT(T)
      ARG=CA*X+CB/X
      KODE=3
      NZ=0
      ERFX=DRERF(ARG,KODE,NZ)
      X2=X*X
      ARG=CB*CB/X2
      ERFX=ERFX*DEXP(-ARG)
      DQJ24=2.0D0*ERFX*X2*X
      RETURN
      END
      SUBROUTINE J24QUAD(A,B,T,QANS24,IERR)
C     QUADRATURE FOR THE J24 INTEGRAL. A CHANGE OF VARIABLES TAU=X*X
C     IS USED. THE UPPER LIMIT ON THE X INTEGRAL IS SQRT(T)=RT.
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/ CFJ24/ CA,CB
      DATA TOL /0.5D-14/
      EXTERNAL DQJ24
      IERR=0
      IF(B.EQ.0.0D0) THEN
        XLIM=4.0D0
      ELSE
        XLIM=B/6.0D0
      ENDIF
      IF(T.EQ.0.0D0) THEN
        QANS24=0.0D0
        IERR=1
        RETURN
      ENDIF
      CA=A
      CB=B
      RT=DSQRT(T)
      JD=10
      INIT=-JD
      X1=0.0D0
      REL=TOL
```

```
      SS=0.0D0
      IF (RT.LE.XLIM) THEN
        SIG=RT/DBLE(FLOAT(JD))
        CALL DQUAD8(DQJ24,INIT,X1,SIG,REL,X2,QANS,IERR)
        SS=SS+QANS
      ELSE
        SIG=XLIM/DBLE(FLOAT(JD))
        CALL DQUAD8(DQJ24,INIT,X1,SIG,REL,X2,QANS,IERR)
        SS=SS+QANS
        INIT=-JD
        X1=X2
        SIG=(RT-X2)/DBLE(FLOAT(JD))
C       TEST FOR RT TOO CLOSE TO X2 FOR AN ACCURATE QUADRATURE
C       (SEE THE ERROR MESSAGE AND TEST IN DGAUS8)
        XMID=X1+0.5D0*SIG
        FMID=DQJ24(XMID)
        IF(DABS(FMID*SIG).LT.2.0D0*TOL*DABS(SS)) GOTO 20
        REL=TOL
        CALL DQUAD8(DQJ24,INIT,X1,SIG,REL,X2,QANS,KERR)
        IF(KERR.NE.1) THEN
          IERR=KERR
        ENDIF
        SS=SS+QANS
  20    CONTINUE
      ENDIF
      QANS24=SS
      RETURN
      END
      SUBROUTINE INTEGJ24(A,B,T,ANS24,I2ERR)
C-----------------------------------------------------------------------
C     REFERENCE: FOLDER24
C
C     INTEGJ24 COMPUTES THE INTEGRAL
C
C          J24=INT ON (0,T) OF W*U(A,B,W)
C
C     WITH U(A,B,W)=EXP(A*A*W+2*A*B)*ERFC(A*RW+B/RW), RW=SQRT(W)
C
C     AND A, B ARE NON-NGATIVE.
C-----------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION Y(50),EN(1)
      DATA REL / 0.5D-14/
      DATA RTPI /1.772453850905516D0/
      I2ERR=0
      RT=DSQRT(T)
      AS=A*A
      BS=B*B
      CAPT=1.0D0/RT
      ART=A*RT
      IF(ART.LE.1.0D0) THEN
        N0=4
        KODE=1
        N=50
        X=B*CAPT
        CALL DINERFC(X,N0,KODE,N,REL,Y,NZ)
        SUM=Y(1)
        X=ART+ART
        PW=-X
        DO 10 K=2,N
          TRM=PW*Y(K)
          SUM=SUM+TRM
          IF(DABS(TRM).LT.REL*DABS(SUM)) GOTO 20
          PW=-X*PW
 10     CONTINUE
 20     CONTINUE
        TOL=REL
```

```
        VT=DVOFT(A,B,T,TOL,IERR,KFORM)
        ANS24=T*(VT-16.0D0*T*SUM)
        RETURN
      ENDIF
      TOL=REL
      VT=DVOFT(A,B,T,TOL,IERR,KFORM)
      SUM=(T-1.0D0/AS)*VT
      IFORM=3
      ARG=B*CAPT
      N0=1
      KKODE=1
      N=3
      TOL=REL
      CALL DINERFC(ARG,N0,KKODE,N,TOL,Y,NZ)
      IF(IFORM.EQ.3) THEN
        ANS24=SUM-8.0D0*T*RT*Y(3)/A+4.0D0*T*Y(2)/AS
        RETURN
      ENDIF
      IF (IFORM.EQ.1) THEN
        COEF=4.0D0/(3.0D0*A*RTPI)
        TRM=(T*DEXP(-ARG*ARG)-2.0D0*BS*RTPI*Y(1))*RT
        SUM=SUM-COEF*TRM
        COEF=4.0D0*(1.0D0/AS+(B+B)/A)*T
        ANS24=SUM+COEF*Y(2)
        RETURN
      ELSE
        COEF=2.0D0/(A*RTPI)
        FNH=2.5D0
        KKODE=1
        M=1
        ARG2=ARG*ARG
        CALL DHEXINT(ARG2, FNH, KKODE, M, TOL, EN, NZ, IERR)
        TRM=T*RT*EN(1)
        SUM=SUM-COEF*TRM
        COEF=4.0D0*(1.0D0/AS+(B+B)/A)*T
        ANS24=SUM+COEF*Y(2)
        RETURN
      ENDIF
      END
      PROGRAM V24COMP
C
C     DONALD E. AMOS   DECEMBER, 2002; MAY, 2006
C
C     REF: FOLDER 24
C
C     V24COMP COMPARES THE PROCEDURE FROM SUBROUTINE INTEGV24
C     AND DGAUS8 QUADRATURE EVALUATIONS OF
C
C          V24=INT ON (0,T) OF V(A,B,W)
C
C     WHERE V(A,B,W)=DVOFT=INT ON (0,T) OF U(A,B,W)    WITH
C           U(A,B,T)=EXP(A*A*T+2*A*B)*ERFC(A*RT+B/RT), RT=SQRT(T)
C
C     AND A, B ARE NON-NGATIVE.
C
C     OUTPUT IS WRITTEN TO FILE V24COMP.TXT
C-----------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      OPEN(UNIT=7,FILE="V24COMP.TXT")
C
      WRITE (7,110)
 110  FORMAT('     A         B          T         QUAD        REL ERR
     &IERR')
      DO 10 IT=1,21,2
        T=DBLE(FLOAT(IT))/5.0D0
        DO 20 IA=1,21,2
          A=DBLE(FLOAT(IA-1))/5.0D0
```

```
          DO 30 IB=1,21,2
            B=DBLE(FLOAT(IB-1))/5.0D0
            CALL INTEGV24(A,B,T,ANS2,J2ERR)
            IF((J2ERR.NE.0)) THEN
              WRITE (7,405) A,B,T,J2ERR
 405          FORMAT( 3D11.4,1X,'IN INTEGV24, J2ERR=',I2)
              GOTO 30
            ENDIF
            CALL V24QUAD(A,B,T,QANS,IERR)
            ERR=DABS(ANS2-QANS)
            IF (QANS.NE.0.0D0) THEN
              ERR=ERR/DABS(QANS)
            ENDIF
            WRITE (7,200) A,B,T,QANS,ERR,IERR
 200        FORMAT(3D11.4,D12.5,D12.4,I4)
   30     CONTINUE
   20   CONTINUE
   10 CONTINUE
      END
      DOUBLE PRECISION FUNCTION DQV24(X)
C
C     DQV24 COMPUTES THE INTEGRAND FOR THE QUADRATURE COMPARISON
C     OF V24 WITH INTEGV24
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/ CFV24/ CA,CB
C     A CHANGE OF VARIABLE TAU=X*X REMOVES THE SQRT FROM THE INTEGRAND
C     THE UPPER LIMIT ON THE INTEGRAL IS THEN DSQRT(T)
      DATA REL /0.5D-14/
      CCA=CA
      CCB=CB
      X2=X*X
      DQV24=2.0D0*X*DVOFT(CCA,CCB,X2,REL,IERR,KFORM)
      RETURN
      END
      SUBROUTINE V24QUAD(A,B,T,QANS24,IERR)
C     QUADRATURE FOR THE V24 INTEGRAL. A CHANGE OF VARIABLES TAU=X*X
C     IS USED. THE UPPER LIMIT ON THE X INTEGRAL IS SQRT(T)=RT.
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/ CFV24/ CA,CB
      DATA TOL /0.50D-14/
      EXTERNAL DQV24
      IERR=0
      IF(B.EQ.0.0D0) THEN
        XLIM=4.0D0
      ELSE
        XLIM=B/6.0D0
      ENDIF
      IF(T.EQ.0.0D0) THEN
        QANS24=0.0D0
        IERR=1
        RETURN
      ENDIF
      CA=A
      CB=B
      RT=DSQRT(T)
      JD=10
      INIT=-JD
      X1=0.0D0
      REL=TOL
      SS=0.0D0
      IF (RT.LE.XLIM) THEN
        SIG=RT/DBLE(FLOAT(JD))
        CALL DQUAD8(DQV24,INIT,X1,SIG,REL,X2,QANS,IERR)
        SS=SS+QANS
      ELSE
        SIG=XLIM/DBLE(FLOAT(JD))
```

```
        CALL DQUAD8(DQV24,INIT,X1,SIG,REL,X2,QANS,IERR)
        SS=SS+QANS
        INIT=-JD
        X1=X2
        SIG=(RT-X2)/DBLE(FLOAT(JD))
C       TEST FOR RT TOO CLOSE TO X2 FOR AN ACCURATE QUADRATURE
C       (SEE THE ERROR MESSAGE AND TEST IN DGAUS8)
        XMID=X1+0.5D0*SIG
        FMID=DQV24(XMID)
        IF(DABS(FMID*SIG).LT.2.0D0*TOL*DABS(SS)) GOTO 20
        REL=TOL
        CALL DQUAD8(DQV24,INIT,X1,SIG,REL,X2,QANS,KERR)
        IF(KERR.NE.1) THEN
          IERR=KERR
        ENDIF
        SS=SS+QANS
  20    CONTINUE
      ENDIF
      QANS24=SS
      RETURN
      END
      SUBROUTINE INTEGV24(A,B,T,ANS24,I2ERR)
C---------------------------------------------------------------------
C     REFERENCE: FOLDER24
C
C     INTEGV24 COMPUTES THE INTEGRAL
C
C          V24=INT ON (0,T) OF V(A,B,W)
C
C     WHERE V(A,B,W)=DVOFT=INT ON (0,T) OF U(A,B,W)    WITH
C           U(A,B,W)=EXP(A*A*W+2*A*B)*ERFC(A*RW+B/RW), RW=SQRT(W)
C
C     AND A, B ARE NON-NGATIVE.
C---------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION Y(50),EN(1)
      DATA RTPI /1.772453850905516D0/
      I2ERR=0
      RT=DSQRT(T)
      AS=A*A
      BS=B*B
      CAPT=1.0D0/RT
      REL=0.5D-14
      ART=A*RT
      IF(ART.LE.1.0D0) THEN
        N0=4
        KODE=1
        N=50
        X=B*CAPT
        CALL DINERFC(X,N0,KODE,N,REL,Y,NZ)
        SUM=Y(1)
        X=ART+ART
        PW=-X
        DO 10 K=2,N
          TRM=PW*Y(K)
          SUM=SUM+TRM
          IF(DABS(TRM).LT.REL*DABS(SUM)) GOTO 20
          PW=-X*PW
 10     CONTINUE
 20     CONTINUE
        ANS24=16.0D0*T*T*SUM
        RETURN
      ENDIF
      VT=DVOFT(A,B,T,REL,IERR,KFORM)
      SUM=VT/AS
      IFORM=3
      ARG=B*CAPT
```

```
      N0=1
      KKODE=1
      N=3
      REL=0.5D-14
      CALL DINERFC(ARG,N0,KKODE,N,REL,Y,NZ)
      IF (IFORM.EQ.3) THEN
        ANS24=SUM+8.0D0*T*RT*Y(3)/A-4.0D0*T*Y(2)/AS
        RETURN
      ENDIF
      IF (IFORM.EQ.1) THEN
        COEF=4.0D0/(3.0D0*A*RTPI)
        TRM=(T*DEXP(-ARG*ARG)-2.0D0*BS*RTPI*Y(1))*RT
        SUM=SUM+COEF*TRM
        COEF=4.0D0*(1.0D0/AS+(B+B)/A)*T
        ANS24=SUM-COEF*Y(2)
        RETURN
      ENDIF
      IF (IFORM.EQ.2) THEN
        COEF=2.0D0/(A*RTPI)
        FNH=2.5D0
        KKODE=1
        M=1
        ARG2=ARG*ARG
        CALL DHEXINT(ARG2, FNH, KKODE, M, REL, EN, NZ, IERR)
        TRM=T*RT*EN(1)
        SUM=SUM+COEF*TRM
        COEF=4.0D0*(1.0D0/AS+(B+B)/A)*T
        ANS24=SUM-COEF*Y(2)
        RETURN
      ENDIF
      END
      PROGRAM I25COMP
C
C     DONALD E. AMOS   JANUARY, 2003;  MAY, 2006
C
C     REF: FOLDER 25
C
C     I25COMP COMPARES THE NON-SYMMETRIC FORMULA CODED IN INTEGI25
C     AND DGAUS8 QUADRATURE EVALUATIONS OF
C
C          I25=INT ON (0,T) OF U(A,B,W)*U(C,D,W)
C
C     WITH U(A,B,T)=EXP(A*A*T+2*A*B)*ERFC(A*RT+B/RT), RT=SQRT(T)
C
C     AND A, B ARE NON-NGATIVE.
C
C     OUTPUT IS WRITTEN TO FILE I25COMP.TXT
C-----------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      OPEN(UNIT=7,FILE="I25COMP.TXT")
      WRITE (7,110)
 110  FORMAT('      A         B         C         D         T       QUAD
     &    REL ERR  IERR')
      DO 10 IT=1,11,2
        T=DBLE(FLOAT(IT))/5.0D0
        PRINT *,'IT = ',IT
        DO 20 IA=1,5
          A=DBLE(FLOAT(IA))/2.0D0
C         PRINT *,' IA = ',IA

          DO 30 IB=1,5
            B=DBLE(FLOAT(IB))/2.0D0
C         PRINT *,'   IB = ',IB

            DO 40 IC=1,5
              C=DBLE(FLOAT(IC))/2.0D0
C         PRINT *,'     IC = ',IC
```

```
              DO 50 ID=1,5
                D=DBLE(FLOAT(ID))/2.0D0
                CALL INTEGI25(A,B,C,D,T,ANS25,I2ERR)
C        PRINT *,'       ID = ',ID
                IF((I2ERR.NE.0)) THEN
                  WRITE (7,405) A,B,C,D,T,I2ERR
 405              FORMAT( 5D11.4,1X,'IN INTEGI25, I2ERR=',I2)
                  GOTO 50
                ENDIF
                CALL I25QUAD(A,B,C,D,T,QANS,IERR)
                ERR=DABS(ANS25-QANS)
                IF (QANS.NE.0.0D0) THEN
                  ERR=ERR/DABS(QANS)
                ENDIF
                WRITE (7,200) A,B,C,D,T,QANS,ERR,IERR
 200            FORMAT(5D10.3,D10.3,D10.3,I4)
   50         CONTINUE
   40       CONTINUE
   30     CONTINUE
   20   CONTINUE
   10 CONTINUE
      END
      DOUBLE PRECISION FUNCTION DQI25(X)
C
C     DQI25 COMPUTES THE INTEGRAND FOR THE QUADRATURE COMPARISON
C     OF I25 WITH INTEGI25
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/ CFI25/ CA,CB,CC,CD
      ARG=CA*X+CB/X
      KODE=3
      NZ=0
      ERFX=DRERF(ARG,KODE,NZ)
      X2=X*X
      ARG=CB*CB/X2
      ERFAB=ERFX*DEXP(-ARG)
      ARG=CC*X+CD/X
      KODE=3
      NZ=0
      ERFX=DRERF(ARG,KODE,NZ)
      ARG=CD*CD/X2
      ERFCD=ERFX*DEXP(-ARG)
      DQI25=2.0D0*ERFAB*ERFCD*X
      RETURN
      END
      SUBROUTINE I25QUAD(A,B,C,D,T,QANS25,IERR)
C     QUADRATURE FOR THE I25 INTEGRAL. A CHANGE OF VARIABLES TAU=X*X
C     IS USED. THE UPPER LIMIT ON THE X INTEGRAL IS SQRT(T)=RT.
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON /CFI25/ CA,CB,CC,CD
      DATA TOL /0.5D-14/
      EXTERNAL DQI25
      IERR=0
      IF(T.EQ.0.0D0) THEN
        QANS25=0.0D0
        IERR=1
        RETURN
      ENDIF
      XLIM=DSQRT(B*B+D*D)/6.0D0
      CA=A
      CB=B
      CC=C
      CD=D
      RT=DSQRT(T)
      JD=10
      INIT=-JD
```

```
      X1=0.0D0
      REL=TOL
      SS=0.0D0
      IF (RT.LE.XLIM) THEN
        SIG=RT/DBLE(FLOAT(JD))
        CALL DQUAD8(DQI25,INIT,X1,SIG,REL,X2,QANS,IERR)
        SS=SS+QANS
      ELSE
        SIG=XLIM/DBLE(FLOAT(JD))
        CALL DQUAD8(DQI25,INIT,X1,SIG,REL,X2,QANS,IERR)
        SS=SS+QANS
        INIT=-JD
        X1=X2
        SIG=(RT-X2)/DBLE(FLOAT(JD))
C       TEST FOR RT TOO CLOSE TO X2 FOR AN ACCURATE QUADRATURE
C       (SEE THE ERROR MESSAGE AND TEST IN DGAUS8)
        XMID=X1+0.5D0*SIG
        FMID=DQI25(XMID)
        IF(DABS(FMID*SIG).LT.2.0D0*TOL*DABS(SS)) GOTO 20
        REL=TOL
        CALL DQUAD8(DQI25,INIT,X1,SIG,REL,X2,QANS,KERR)
        IF(KERR.NE.1) THEN
          IERR=KERR
        ENDIF
        SS=SS+QANS
  20    CONTINUE
      ENDIF
      QANS25=SS
      RETURN
      END
      SUBROUTINE INTEGI25(A,B,C,D,T,ANS25,I2ERR)
C
C     DONALD E. AMOS, JANUARY, 2003
C-----------------------------------------------------------------------
C     A DOUBLE PRECISION ROUTINE
C-----------------------------------------------------------------------
C
C     REF: FOLDER 25
C
C     INTEGI25 COMPUTES
C
C        ANS25=INT ON (0,T) OF U(A,B,W)*U(C,D,W)
C
C        U(A,B,W)= EXP(A*A*W+2*A*B)*ERFC(A*RT+B/RT), RT=SQRT(W)
C
C     WHERE A, B, C, D  ARE NON-NEGATIVE.
C
C     ERROR FLAG:
C           I2ERR= 0 NORMAL RETURN
C                =-1 CONVERGENCE NOT SEEN IS SOME SERIES
C                = 4 INPUT VARIABLE OUT OF RANGE
C                = 5 EXPONENTIAL UNDERFLOW, (B^2+D^2)/T TOO LARGE
C                =   ANY OTHER VALUE IS AN ERROR FROM DGAUS8
C
C-----------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION EN(2)
      DATA RTPI /1.772453850905516D0/
      I2ERR=0
      IF(A.LT.0.0D0)THEN
         I2ERR=4
         ANS25=0.0D0
         RETURN
      ENDIF
      IF(B.LT.0.0D0)THEN
         I2ERR=4
         ANS25=0.0D0
```

```
            RETURN
         ENDIF
         IF(C.LT.0.0D0)THEN
            I2ERR=4
            ANS25=0.0D0
            RETURN
         ENDIF
         IF(D.LT.0.0D0)THEN
            I2ERR=4
            ANS25=0.0D0
            RETURN
         ENDIF
         IF(T.LT.0.0D0)THEN
            I2ERR=4
            ANS25=0.0D0
            RETURN
         ENDIF
         IF(T.EQ.0.0D0)THEN
            ANS25=0.0D0
            RETURN
         ENDIF
         OPT=1
C        SINCE THE INTEGRAL IS SYMMETRIC IN THE PAIRS (A,B), (C,D) WE CAN
C        EXCHANGE THE PAIRS IF WE WANT THE A PARAMETER TO BE GREATER THAN
C        THE C PARAMETER. THIS IS WHAT OPT=1 DOES.
C        OPT=1, RUNCODE WITH AA=MAX(A,C), CC=MIN(A,C)
C        OPT=2, RUNCODE WITH NO CHANGE IN PARAMETERS
         IF(OPT.EQ.1) THEN
           IF(C.GT.A)THEN
             AA=C
             BB=D
             CC=A
             DD=B
           ELSE
             AA=A
             BB=B
             CC=C
             DD=D
           ENDIF
         ELSE
           AA=A
           BB=B
           CC=C
           DD=D
         ENDIF
         REL=0.5D-14
         RT=DSQRT(T)
         CAPT=1.0D0/RT
         AS=AA*AA
         BS=BB*BB
         CS=CC*CC
         DS=DD*DD
         CAPX=(BS+DS)/T
         IF(CAPX.GT.667.0D0) THEN
           ANS25=0.0D0
           I2ERR=5
           RETURN
         ENDIF
         VABT=DVOFT(AA,BB,T,REL,IVERR,IFORM)
         ARG=CC*RT+DD/RT
         KODE=3
         NZ=0
         ERFU=DRERF(ARG,KODE,NZ)
         ARG=DS/T
         UCDT=ERFU*DEXP(-ARG)
         SS=AS*VABT*UCDT
         N=1
```

```
      KKODE=1
      M=2
      TOL=REL
      NZ=0
      IERR=0
      CALL DEXINT(CAPX, N, KKODE, M, TOL, EN, NZ, IERR)
      CAB=1.0D0+(BB+BB)*AA
      KODE=2
      CALL INTEGI1(DD,BB,CAPT,KODE,REL,ANSI1C,IERR)
      CALL INTEGI22(AA,BB,DD,T,ANSI22,I2ERR,KFORM)
      R1ABDT=AA*T*EN(2)/RTPI-CAB*ANSI1C
      R1ABDT=R1ABDT+R1ABDT+ANSI22
      CALL INTEGI5(DD,BB,CAPT,ANSI5,IERR)
      CALL INTEGJ21(AA,BB,DD,T,ANSJ21,I2ERR,KFORM)
      R2ABDT=AA*EN(1)/RTPI-CAB*ANSI5
      R2ABDT=R2ABDT+R2ABDT+ANSJ21
      SS=SS+(CC*R1ABDT-DD*R2ABDT)/RTPI
      CALL INTEGJ22(CC,DD,BB,T,ANSJ22,I2ERR,KFORM)
      KODE=2
      CALL INTEGI21(CC,DD,BB,T,KODE,ANSI21C,I2ERR,KFORM)
      SS=SS-(AA+AA)*CS*ANSJ22/RTPI+CS*CAB*ANSI21C
      ANS25=SS/(AS+CS)
      RETURN
      END
      PROGRAM I26COMP
C
C     DONALD E. AMOS   APRIL, 2003;  MAY, 2006
C
C     REF: FOLDER 26
C
C     I26COMP COMPARES THE SYMMETRIC FORMULA CODED IN INTEGI26
C     AND DGAUS8 QUADRATURE EVALUATIONS OF
C
C          I26=INT ON (0,T) OF W*U(A,B,W)*U(C,D,W)
C
C     WITH U(A,B,W)=EXP(A*A*W+2*A*B)*ERFC(A*RW+B/RW), RT=SQRT(W)
C
C     AND A, B ARE NON-NGATIVE.
C
C     OUTPUT IS WRITTEN TO FILE I26COMP.TXT
C-----------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      OPEN(UNIT=7,FILE="I26COMP.TXT")
      WRITE (7,110)
 110  FORMAT('      A         B         C         D         T       QUAD
     &    REL ERR  IERR')
      DO 10 IT=1,11,2
        T=DBLE(FLOAT(IT))/5.0D0
        PRINT *,'IT = ',IT
        DO 20 IA=1,5
          A=DBLE(FLOAT(IA))/2.0D0
C        PRINT *,' IA = ',IA

          DO 30 IB=1,5
            B=DBLE(FLOAT(IB))/2.0D0
C        PRINT *,'   IB = ',IB

            DO 40 IC=1,5
              C=DBLE(FLOAT(IC))/2.0D0
C        PRINT *,'     IC = ',IC

              DO 50 ID=1,5
                D=DBLE(FLOAT(ID))/2.0D0
                CALL INTEGI26(A,B,C,D,T,ANS26,I2ERR)
C        PRINT *,'       ID = ',ID
                IF((I2ERR.NE.0)) THEN
                  WRITE (7,405) A,B,C,D,T,I2ERR
```

```
 405           FORMAT( 5D11.4,1X,'IN INTEGI26, I2ERR=',I2)
                 GOTO 50
               ENDIF
               CALL I26QUAD(A,B,C,D,T,QANS,IERR)
               ERR=DABS(ANS26-QANS)
               IF (QANS.NE.0.0D0) THEN
                 ERR=ERR/DABS(QANS)
               ENDIF
               WRITE (7,200) A,B,C,D,T,QANS,ERR,IERR
 200           FORMAT(5D10.3,D10.3,D10.3,I4)
   50        CONTINUE
   40      CONTINUE
   30    CONTINUE
   20  CONTINUE
   10 CONTINUE
      END
      DOUBLE PRECISION FUNCTION DQI26(X)
C
C     DQI26 COMPUTES THE INTEGRAND FOR THE QUADRATURE COMPARISON
C     OF I26 WITH INTEGI26
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/ CFI26/ CA,CB,CC,CD
      ARG=CA*X+CB/X
      KODE=3
      NZ=0
      ERFX=DRERF(ARG,KODE,NZ)
      X2=X*X
      ARG=CB*CB/X2
      ERFAB=ERFX*DEXP(-ARG)
      ARG=CC*X+CD/X
      KODE=3
      NZ=0
      ERFX=DRERF(ARG,KODE,NZ)
      ARG=CD*CD/X2
      ERFCD=ERFX*DEXP(-ARG)
      DQI26=2.0D0*ERFAB*ERFCD*X*X2
      RETURN
      END
      SUBROUTINE I26QUAD(A,B,C,D,T,QANS26,IERR)
C     QUADRATURE FOR THE I26 INTEGRAL. A CHANGE OF VARIABLES TAU=X*X
C     IS USED. THE UPPER LIMIT ON THE X INTEGRAL IS SQRT(T)=RT.
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/ CFI26/ CA,CB,CC,CD
      DATA TOL /0.5D-14/
      EXTERNAL DQI26
      IERR=0
      IF(T.EQ.0.0D0) THEN
        QANS26=0.0D0
        IERR=1
        RETURN
      ENDIF
      XLIM=DSQRT(B*B+D*D)/6.0D0
      CA=A
      CB=B
      CC=C
      CD=D
      RT=DSQRT(T)
      JD=10
      INIT=-JD
      X1=0.0D0
      REL=TOL
      SS=0.0D0
      IF (RT.LE.XLIM) THEN
        SIG=RT/DBLE(FLOAT(JD))
        CALL DQUAD8(DQI26,INIT,X1,SIG,REL,X2,QANS,IERR)
        SS=SS+QANS
```

```
      ELSE
        SIG=XLIM/DBLE(FLOAT(JD))
        CALL DQUAD8(DQI26,INIT,X1,SIG,REL,X2,QANS,IERR)
        SS=SS+QANS
        INIT=-JD
        X1=X2
        SIG=(RT-X2)/DBLE(FLOAT(JD))
C       TEST FOR RT TOO CLOSE TO X2 FOR AN ACCURATE QUADRATURE
C       (SEE THE ERROR MESSAGE AND TEST IN DGAUS8)
        XMID=X1+0.5D0*SIG
        FMID=DQI26(XMID)
        IF(DABS(FMID*SIG).LT.2.0D0*TOL*DABS(SS)) GOTO 20
        REL=TOL
        CALL DQUAD8(DQI26,INIT,X1,SIG,REL,X2,QANS,KERR)
        IF(KERR.NE.1) THEN
          IERR=KERR
        ENDIF
        SS=SS+QANS
  20    CONTINUE
      ENDIF
      QANS26=SS
      RETURN
      END
      SUBROUTINE INTEGI26(A,B,C,D,T,ANS26,I26ERR)
C
C     DONALD E. AMOS, APRIL, 2003
C-----------------------------------------------------------------------
C     A DOUBLE PRECISION ROUTINE
C-----------------------------------------------------------------------
C
C     REF: FOLDER 26
C
C     INTEGI26 COMPUTES ANS26 BY THE SYMMETRIC FORMULA FOR
C
C        I26=INT ON (0,T) OF W*U(A,B,W)*U(C,D,W)
C
C        U(A,B,W)= EXP(A*A*W+2*A*B)*ERFC(A*RT+B/RT), RT=SQRT(W)
C
C     WHERE A, B, C, D  ARE NON-NEGATIVE.
C
C     ERROR FLAG:
C          I26ERR= 0 NORMAL RETURN
C                =-1 CONVERGENCE NOT SEEN IS SOME SERIES
C                = 4 INPUT VARIABLE OUT OF RANGE
C                = 5 EXPONENTIAL UNDERFLOW IN SOME SUBROUTINE
C                =   ANY OTHER VALUE IS AN ERROR FROM DGAUS8
C
C-----------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DATA RTPI /1.772453850905516D0/
      I26ERR=0
      IF(A.LT.0.0D0)THEN
         I26ERR=4
         ANS26=0.0D0
         RETURN
      ENDIF
      IF(B.LT.0.0D0)THEN
         I26ERR=4
         ANS26=0.0D0
         RETURN
      ENDIF
      IF(C.LT.0.0D0)THEN
         I26ERR=4
         ANS26=0.0D0
         RETURN
      ENDIF
      IF(D.LT.0.0D0)THEN
```

```
            I26ERR=4
            ANS26=0.0D0
            RETURN
         ENDIF
         IF(T.LT.0.0D0)THEN
            I26ERR=4
            ANS26=0.0D0
            RETURN
         ENDIF
         IF(T.EQ.0.0D0)THEN
            ANS26=0.0D0
            RETURN
         ENDIF
         RT=DSQRT(T)
         AS=A*A
         BS=B*B
         CS=C*C
         DS=D*D
         ARG=A*RT+B/RT
         KODE=3
         NZ=0
         ERFU=DRERF(ARG,KODE,NZ)
         ARG=BS/T
         UABT=ERFU*DEXP(-ARG)
         ARG=C*RT+D/RT
         KODE=3
         NZ=0
         ERFU=DRERF(ARG,KODE,NZ)
         ARG=DS/T
         UCDT=ERFU*DEXP(-ARG)
         SS=T*UABT*UCDT
         CALL INTEGI25(A,B,C,D,T,ANSI25,I2ERR)
         I26ERR=I26ERR+I2ERR
         IF(I26ERR.NE.0) THEN
           ANS26=0.0D0
           RETURN
         ENDIF
         CALL INTEGI22(A,B,D,T,AI22AB,I2ERR,KFORM)
         I26ERR=I26ERR+I2ERR
         IF(I26ERR.NE.0) THEN
           ANS26=0.0D0
           RETURN
         ENDIF
         CALL INTEGJ22(A,B,D,T,AJ22AB,I2ERR,KFORM)
         I26ERR=I26ERR+I2ERR
         IF(I26ERR.NE.0) THEN
           ANS26=0.0D0
           RETURN
         ENDIF
         CALL INTEGI22(C,D,B,T,AI22CD,I2ERR,KFORM)
         I26ERR=I26ERR+I2ERR
         IF(I26ERR.NE.0) THEN
           ANS26=0.0D0
           RETURN
         ENDIF
         CALL INTEGJ22(C,D,B,T,AJ22CD,I2ERR,KFORM)
         I26ERR=I26ERR+I2ERR
         IF(I26ERR.NE.0) THEN
           ANS26=0.0D0
           RETURN
         ENDIF
         SS=SS-ANSI25+(C*AJ22AB-D*AI22AB+A*AJ22CD-B*AI22CD)/RTPI
         ANS26=SS/(AS+CS)
         RETURN
         END
         SUBROUTINE INTEGI25(A,B,C,D,T,ANS25,I2ERR)
C
```

```
C     DONALD E. AMOS, JANUARY, 2003
C-----------------------------------------------------------------------
C     A DOUBLE PRECISION ROUTINE
C-----------------------------------------------------------------------
C
C     REF: FOLDER 25
C
C     INTEGI25 COMPUTES
C
C        ANS25=INT ON (0,T) OF U(A,B,W)*U(C,D,W)
C
C        U(A,B,W)= EXP(A*A*W+2*A*B)*ERFC(A*RT+B/RT), RT=SQRT(W)
C
C     WHERE A, B, C, D  ARE NON-NEGATIVE.
C
C     ERROR FLAG:
C           I2ERR= 0 NORMAL RETURN
C                =-1 CONVERGENCE NOT SEEN IS SOME SERIES
C                = 4 INPUT VARIABLE OUT OF RANGE
C                = 5 EXPONENTIAL UNDERFLOW, (B^2+D^2)/T TOO LARGE
C                =   ANY OTHER VALUE IS AN ERROR FROM DGAUS8
C
C-----------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION EN(2)
      DATA RTPI /1.772453850905516D0/
      I2ERR=0
      IF(A.LT.0.0D0)THEN
         I2ERR=4
         ANS25=0.0D0
         RETURN
      ENDIF
      IF(B.LT.0.0D0)THEN
         I2ERR=4
         ANS25=0.0D0
         RETURN
      ENDIF
      IF(C.LT.0.0D0)THEN
         I2ERR=4
         ANS25=0.0D0
         RETURN
      ENDIF
      IF(D.LT.0.0D0)THEN
         I2ERR=4
         ANS25=0.0D0
         RETURN
      ENDIF
      IF(T.LT.0.0D0)THEN
         I2ERR=4
         ANS25=0.0D0
         RETURN
      ENDIF
      IF(T.EQ.0.0D0)THEN
         ANS25=0.0D0
         RETURN
      ENDIF
      OPT=1
C     SINCE THE INTEGRAL IS SYMMETRIC IN THE PAIRS (A,B), (C,D) WE CAN
C     EXCHANGE THE PAIRS IF WE WANT THE A PARAMETER TO BE GREATER THAN
C     THE C PARAMETER. THIS IS WHAT OPT=1 DOES.
C     OPT=1, RUNCODE WITH AA=MAX(A,C), CC=MIN(A,C)
C     OPT=2, RUNCODE WITH NO CHANGE IN PARAMETERS
      IF(OPT.EQ.1) THEN
        IF(C.GT.A)THEN
          AA=C
          BB=D
          CC=A
```

```
            DD=B
          ELSE
            AA=A
            BB=B
            CC=C
            DD=D
          ENDIF
        ELSE
          AA=A
          BB=B
          CC=C
          DD=D
        ENDIF
        REL=0.5D-14
        RT=DSQRT(T)
        CAPT=1.0D0/RT
        AS=AA*AA
        BS=BB*BB
        CS=CC*CC
        DS=DD*DD
        CAPX=(BS+DS)/T
        IF(CAPX.GT.667.0D0) THEN
          ANS25=0.0D0
          I2ERR=5
          RETURN
        ENDIF
        VABT=DVOFT(AA,BB,T,REL,IVERR,IFORM)
        ARG=CC*RT+DD/RT
        KODE=3
        NZ=0
        ERFU=DRERF(ARG,KODE,NZ)
        ARG=DS/T
        UCDT=ERFU*DEXP(-ARG)
        SS=AS*VABT*UCDT
        N=1
        KKODE=1
        M=2
        TOL=REL
        NZ=0
        IERR=0
        CALL DEXINT(CAPX, N, KKODE, M, TOL, EN, NZ, IERR)
        CAB=1.0D0+(BB+BB)*AA
        KODE=2
        CALL INTEGI1(DD,BB,CAPT,KODE,REL,ANSI1C,IERR)
        CALL INTEGI22(AA,BB,DD,T,ANSI22,I2ERR,KFORM)
        R1ABDT=AA*T*EN(2)/RTPI-CAB*ANSI1C
        R1ABDT=R1ABDT+R1ABDT+ANSI22
        CALL INTEGI5(DD,BB,CAPT,ANSI5,IERR)
        CALL INTEGJ21(AA,BB,DD,T,ANSJ21,I2ERR,KFORM)
        R2ABDT=AA*EN(1)/RTPI-CAB*ANSI5
        R2ABDT=R2ABDT+R2ABDT+ANSJ21
        SS=SS+(CC*R1ABDT-DD*R2ABDT)/RTPI
        CALL INTEGJ22(CC,DD,BB,T,ANSJ22,I2ERR,KFORM)
        KODE=2
        CALL INTEGI21(CC,DD,BB,T,KODE,ANSI21C,I2ERR,KFORM)
        SS=SS-(AA+AA)*CS*ANSJ22/RTPI+CS*CAB*ANSI21C
        ANS25=SS/(AS+CS)
        RETURN
        END
        PROGRAM I26ACOMP
C
C       DONALD E. AMOS   APRIL, 2003;  MAY, 2006
C
C       REF: FOLDER 26
C
C       I26ACOMP COMPARES THE SYMMETRIC FORMULA CODED IN INTEGI26A
C       AND DGAUS8 QUADRATURE EVALUATIONS OF
```

```
C
C           I26A=INT ON (0,T) OF W*U(A,B,W)*U(C,D,W)
C
C     WITH U(A,B,W)=EXP(A*A*W+2*A*B)*ERFC(A*RW+B/RW), RT=SQRT(W)
C
C     AND A, B ARE NON-NGATIVE.
C
C     OUTPUT IS WRITTEN TO FILE I26ACOMP.TXT
C-----------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      OPEN(UNIT=7,FILE="I26ACOMP.TXT")
      WRITE (7,110)
 110  FORMAT('      A         B         C         D         T       QUAD
     &    REL ERR  IERR')
      DO 10 IT=1,11,2
        T=DBLE(FLOAT(IT))/5.0D0
        PRINT *,'IT = ',IT
        DO 20 IA=1,5
          A=DBLE(FLOAT(IA))/2.0D0
C        PRINT *,' IA = ',IA

          DO 30 IB=1,5
            B=DBLE(FLOAT(IB))/2.0D0
C        PRINT *,'   IB = ',IB

            DO 40 IC=1,5
              C=DBLE(FLOAT(IC))/2.0D0
C        PRINT *,'     IC = ',IC

              DO 50 ID=1,5
                D=DBLE(FLOAT(ID))/2.0D0
                CALL INTEG26A(A,B,C,D,T,ANS26,I2ERR)
C        PRINT *,'       ID = ',ID
                IF((I2ERR.NE.0)) THEN
                  WRITE (7,405) A,B,C,D,T,I2ERR
 405            FORMAT( 5D11.4,1X,'IN INTEGI26A, I2ERR=',I2)
                  GOTO 50
                ENDIF
                CALL I26AQUAD(A,B,C,D,T,QANS,IERR)
                ERR=DABS(ANS26-QANS)
                IF (QANS.NE.0.0D0) THEN
                  ERR=ERR/DABS(QANS)
                ENDIF
                WRITE (7,200) A,B,C,D,T,QANS,ERR,IERR
 200            FORMAT(5D10.3,D10.3,D10.3,I4)
   50         CONTINUE
   40       CONTINUE
   30     CONTINUE
   20   CONTINUE
   10 CONTINUE
      END
      DOUBLE PRECISION FUNCTION DQI26A(X)
C
C     DQI26A COMPUTES THE INTEGRAND FOR THE QUADRATURE COMPARISON
C     OF I26A WITH INTEG26A
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/ CFI26A/ CA,CB,CC,CD
      ARG=CA*X+CB/X
      KODE=3
      NZ=0
      ERFX=DRERF(ARG,KODE,NZ)
      X2=X*X
      ARG=CB*CB/X2
      ERFAB=ERFX*DEXP(-ARG)
      ARG=CC*X+CD/X
      KODE=3
```

```
      NZ=0
      ERFX=DRERF(ARG,KODE,NZ)
      ARG=CD*CD/X2
      ERFCD=ERFX*DEXP(-ARG)
      DQI26A=2.0D0*ERFAB*ERFCD*X*X2
      RETURN
      END
      SUBROUTINE I26AQUAD(A,B,C,D,T,QANS26,IERR)
C     QUADRATURE FOR THE I26A INTEGRAL. A CHANGE OF VARIABLES TAU=X*X
C     IS USED. THE UPPER LIMIT ON THE X INTEGRAL IS SQRT(T)=RT.
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/ CFI26A/ CA,CB,CC,CD
      DATA TOL /0.5D-14/
      EXTERNAL DQI26A
      IERR=0
      IF(T.EQ.0.0D0) THEN
        QANS26=0.0D0
        IERR=1
        RETURN
      ENDIF
      XLIM=DSQRT(B*B+D*D)/6.0D0
      CA=A
      CB=B
      CC=C
      CD=D
      RT=DSQRT(T)
      JD=10
      INIT=-JD
      X1=0.0D0
      REL=TOL
      SS=0.0D0
      IF (RT.LE.XLIM) THEN
        SIG=RT/DBLE(FLOAT(JD))
        CALL DQUAD8(DQI26A,INIT,X1,SIG,REL,X2,QANS,IERR)
        SS=SS+QANS
      ELSE
        SIG=XLIM/DBLE(FLOAT(JD))
        CALL DQUAD8(DQI26A,INIT,X1,SIG,REL,X2,QANS,IERR)
        SS=SS+QANS
        INIT=-JD
        X1=X2
        SIG=(RT-X2)/DBLE(FLOAT(JD))
C       TEST FOR RT TOO CLOSE TO X2 FOR AN ACCURATE QUADRATURE
C       (SEE THE ERROR MESSAGE AND TEST IN DGAUS8)
        XMID=X1+0.5D0*SIG
        FMID=DQI26A(XMID)
        IF(DABS(FMID*SIG).LT.2.0D0*TOL*DABS(SS)) GOTO 20
        REL=TOL
        CALL DQUAD8(DQI26A,INIT,X1,SIG,REL,X2,QANS,KERR)
        IF(KERR.NE.1) THEN
          IERR=KERR
        ENDIF
        SS=SS+QANS
  20    CONTINUE
      ENDIF
      QANS26=SS
      RETURN
      END
      SUBROUTINE INTEG26A(A,B,C,D,T,ANS26A,I26AERR)
C
C     DONALD E. AMOS, APRIL, 2003
C----------------------------------------------------------------------
C     A DOUBLE PRECISION ROUTINE
C----------------------------------------------------------------------
C
C     REF: FOLDER 26
C
```

```
C     INTEG26A COMPUTES THE NON-SYMMETRIC FORM OF I26 IN FOLDER 26 FOR
C
C        I26=INT ON (0,T) OF W*U(A,B,W)*U(C,D,W)
C
C        U(A,B,W)= EXP(A*A*W+2*A*B)*ERFC(A*RT+B/RT), RT=SQRT(W)
C
C     WHERE A, B, C, D  ARE NON-NEGATIVE.
C
C     ERROR FLAG:
C          I26AERR= 0 NORMAL RETURN
C                 =-1 CONVERGENCE NOT SEEN IS SOME SERIES
C                 = 4 INPUT VARIABLE OUT OF RANGE
C                 = 5 EXPONENTIAL UNDERFLOW IN SOME SUBROUTINE
C                 =   ANY OTHER VALUE IS AN ERROR FROM DGAUS8
C
C------------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION EN(2)
      DATA RTPI /1.772453850905516D0/
      I26AERR=0
      IF(A.LT.0.0D0)THEN
         I26AERR=4
         ANS26A=0.0D0
         RETURN
      ENDIF
      IF(B.LT.0.0D0)THEN
         I26AERR=4
         ANS26A=0.0D0
         RETURN
      ENDIF
      IF(C.LT.0.0D0)THEN
         I26AERR=4
         ANS26A=0.0D0
         RETURN
      ENDIF
      IF(D.LT.0.0D0)THEN
         I26AERR=4
         ANS26A=0.0D0
         RETURN
      ENDIF
      IF(T.LT.0.0D0)THEN
         I26AERR=4
         ANS26A=0.0D0
         RETURN
      ENDIF
      IF(T.EQ.0.0D0)THEN
         ANS26A=0.0D0
         RETURN
      ENDIF
      OPT=2
C     SINCE THE INTEGRAL IS SYMMETRIC IN THE PAIRS (A,B), (C,D) WE CAN
C     EXCHANGE THE PAIRS IF WE WANT THE A PARAMETER TO BE GREATER THAN
C     THE C PARAMETER. THIS IS WHAT OPT=1 DOES.
C     OPT=1, RUNCODE WITH AA=MAX(A,C), CC=MIN(A,C)
C     OPT=2, RUNCODE WITH NO CHANGE IN PARAMETERS
      IF(OPT.EQ.1) THEN
        IF(C.GT.A)THEN
          AA=C
          BB=D
          CC=A
          DD=B
        ELSE
          AA=A
          BB=B
          CC=C
          DD=D
        ENDIF
```

```
      ELSE
        AA=A
        BB=B
        CC=C
        DD=D
      ENDIF
      REL=0.5D-14
      RT=DSQRT(T)
      CAPT=1.0D0/RT
      AS=AA*AA
      BS=BB*BB
      CS=CC*CC
      DS=DD*DD
      CAPX=(BS+DS)/T
      IF(CAPX.GT.667.0D0) THEN
        ANS26A=0.0D0
        I2ERR=5
        RETURN
      ENDIF
      VABT=DVOFT(AA,BB,T,REL,IVERR,IFORM)
      ARG=CC*RT+DD/RT
      KODE=3
      NZ=0
      ERFU=DRERF(ARG,KODE,NZ)
      ARG=DS/T
      UCDT=ERFU*DEXP(-ARG)
      SS=AS*T*VABT*UCDT
      N=2
      KKODE=1
      M=2
      TOL=REL
      NZ=0
      IERR=0
      CALL DEXINT(CAPX, N, KKODE, M, TOL, EN, NZ, IERR)
      CAB=1.0D0+2.0D0*AA*BB
      CABP=CAB+CAB
      T2=T*T
      TRT=T*RT
      EX=DEXP(-CAPX)
      ARG=BB/RT
      KODE=3
      ERFB=DRERF(ARG,KODE,NZ)
      KODE=2
      CALL INTEGI1(DD,BB,CAPT,KODE,REL,ANSI1DB,IERR)
      R12DBT=(EX*ERFB*TRT-BB*T*EN(1)/RTPI-(DS+DS)*ANSI1DB)/3.0D0
      CALL INTEGJ22(AA,BB,DD,T,ANSJ22,J22ERR,KFORM)
      R1ABDT=(AA+AA)*T2*EN(2)/RTPI-CABP*R12DBT+ANSJ22
      CALL INTEGI22(AA,BB,DD,T,ANSI22,I22ERR,KFORM)
      R2ABDT=(AA+AA)*T*EN(1)/RTPI-CABP*ANSI1DB+ANSI22
      SS=SS+(CC*R1ABDT-DD*R2ABDT)/RTPI
      CALL INTEGI25(AA,BB,CC,DD,T,ANSI25,I2ERR)
      CALL INTEGJ22(CC,DD,BB,T,ANSJ22,J22ERR,KFORM)
      KODE=2
      CALL INTEGI21(CC,DD,BB,T,KODE,ANSI21,I21ERR,KFORM)
      R3ABCDT=(AA+AA)*ANSJ22/RTPI-CAB*ANSI21+ANSI25
      SS=SS-R3ABCDT
      CCD=1.0D0+2.0D0*CC*DD
      CCDP=CCD+CCD
      KODE=2
      CALL INTEGI1(BB,DD,CAPT,KODE,REL,ANSI1BD,IERR)
      CALL INTEGI22(CC,DD,BB,T,ANSI22,I22ERR,KFORM)
      R2CDBT=(T+T)*CC*EN(1)/RTPI-CCDP*ANSI1BD+ANSI22
      ARG=DD/RT
      KODE=3
      ERFD=DRERF(ARG,KODE,NZ)
      R12BDT=(EX*ERFD*TRT-DD*T*EN(1)/RTPI-(BS+BS)*ANSI1BD)/3.0D0
      R413CDBT=(T2+T2)*CC*EN(2)/RTPI-CCDP*R12BDT+ANSJ22
```

```
      ARG=BS/T
      EX=DEXP(-ARG)
      VCDT=DVOFT(CC,DD,T,REL,IVERR,IFORM)
      R41CDBT=CS*TRT*EX*VCDT-BS*R2CDBT-1.5D0*R413CDBT
      KODE=2
      CALL INTEGW3(BB,DD,CAPT,KODE,REL,ANSIW3,IERR)
      R423CDBT=4.0D0*CC*R12DBT/RTPI-CCDP*ANSIW3+ANSI21
      R42CDBT=CS*T*ERFB*EX*VCDT-BB*R2CDBT/RTPI-R423CDBT
      R4ABCDT=(AA+AA)*R41CDBT/RTPI-CAB*R42CDBT
      ANS26A=(SS-R4ABCDT)/(AS+CS)
      RETURN
      END
      SUBROUTINE INTEGI25(A,B,C,D,T,ANS25,I2ERR)
C
C     DONALD E. AMOS, JANUARY, 2003
C-----------------------------------------------------------------------
C     A DOUBLE PRECISION ROUTINE
C-----------------------------------------------------------------------
C
C     REF: FOLDER 25
C
C     INTEGI25 COMPUTES
C
C        ANS25=INT ON (0,T) OF U(A,B,W)*U(C,D,W)
C
C        U(A,B,W)= EXP(A*A*W+2*A*B)*ERFC(A*RT+B/RT), RT=SQRT(W)
C
C     WHERE A, B, C, D  ARE NON-NEGATIVE.
C
C     ERROR FLAG:
C           I2ERR= 0 NORMAL RETURN
C                =-1 CONVERGENCE NOT SEEN IS SOME SERIES
C                = 4 INPUT VARIABLE OUT OF RANGE
C                = 5 EXPONENTIAL UNDERFLOW, (B^2+D^2)/T TOO LARGE
C                =   ANY OTHER VALUE IS AN ERROR FROM DGAUS8
C
C-----------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION EN(2)
      DATA RTPI /1.772453850905516D0/
      I2ERR=0
      IF(A.LT.0.0D0)THEN
         I2ERR=4
         ANS25=0.0D0
         RETURN
      ENDIF
      IF(B.LT.0.0D0)THEN
         I2ERR=4
         ANS25=0.0D0
         RETURN
      ENDIF
      IF(C.LT.0.0D0)THEN
         I2ERR=4
         ANS25=0.0D0
         RETURN
      ENDIF
      IF(D.LT.0.0D0)THEN
         I2ERR=4
         ANS25=0.0D0
         RETURN
      ENDIF
      IF(T.LT.0.0D0)THEN
         I2ERR=4
         ANS25=0.0D0
         RETURN
      ENDIF
      IF(T.EQ.0.0D0)THEN
```

```
            ANS25=0.0D0
            RETURN
      ENDIF
      OPT=1
C     SINCE THE INTEGRAL IS SYMMETRIC IN THE PAIRS (A,B), (C,D) WE CAN
C     EXCHANGE THE PAIRS IF WE WANT THE A PARAMETER TO BE GREATER THAN
C     THE C PARAMETER. THIS IS WHAT OPT=1 DOES.
C     OPT=1, RUNCODE WITH AA=MAX(A,C), CC=MIN(A,C)
C     OPT=2, RUNCODE WITH NO CHANGE IN PARAMETERS
      IF(OPT.EQ.1) THEN
        IF(C.GT.A)THEN
          AA=C
          BB=D
          CC=A
          DD=B
        ELSE
          AA=A
          BB=B
          CC=C
          DD=D
        ENDIF
      ELSE
        AA=A
        BB=B
        CC=C
        DD=D
      ENDIF
      REL=0.5D-14
      RT=DSQRT(T)
      CAPT=1.0D0/RT
      AS=AA*AA
      BS=BB*BB
      CS=CC*CC
      DS=DD*DD
      CAPX=(BS+DS)/T
      IF(CAPX.GT.667.0D0) THEN
        ANS25=0.0D0
        I2ERR=5
        RETURN
      ENDIF
      VABT=DVOFT(AA,BB,T,REL,IVERR,IFORM)
      ARG=CC*RT+DD/RT
      KODE=3
      NZ=0
      ERFU=DRERF(ARG,KODE,NZ)
      ARG=DS/T
      UCDT=ERFU*DEXP(-ARG)
      SS=AS*VABT*UCDT
      N=1
      KKODE=1
      M=2
      TOL=REL
      NZ=0
      IERR=0
      CALL DEXINT(CAPX, N, KKODE, M, TOL, EN, NZ, IERR)
      CAB=1.0D0+(BB+BB)*AA
      KODE=2
      CALL INTEGI1(DD,BB,CAPT,KODE,REL,ANSI1C,IERR)
      CALL INTEGI22(AA,BB,DD,T,ANSI22,I2ERR,KFORM)
      R1ABDT=AA*T*EN(2)/RTPI-CAB*ANSI1C
      R1ABDT=R1ABDT+R1ABDT+ANSI22
      CALL INTEGI5(DD,BB,CAPT,ANSI5,IERR)
      CALL INTEGJ21(AA,BB,DD,T,ANSJ21,I2ERR,KFORM)
      R2ABDT=AA*EN(1)/RTPI-CAB*ANSI5
      R2ABDT=R2ABDT+R2ABDT+ANSJ21
      SS=SS+(CC*R1ABDT-DD*R2ABDT)/RTPI
      CALL INTEGJ22(CC,DD,BB,T,ANSJ22,I2ERR,KFORM)
```

```
      KODE=2
      CALL INTEGI21(CC,DD,BB,T,KODE,ANSI21C,I2ERR,KFORM)
      SS=SS-(AA+AA)*CS*ANSJ22/RTPI+CS*CAB*ANSI21C
      ANS25=SS/(AS+CS)
      RETURN
      END
      PROGRAM DGSCOMP
C
C     WRITTEN BY DONALD E. AMOS, SEPTEMBER, 2005.
C
C     REFERENCE: FOLDERS 6 AND 16
C
C     PROGRAM DGSCOMP IMPLEMENTS THE ACCELERATION OF THE SERIES FOR
C
C              G(X)= INT ON (X, INF) OF ERFC(U)/U
C
C     IN TERMS OF THE EXPONENTIAL INTEGRAL AND COMPARES THE RESULTS WITH
C     THE IMPLEMENTATION OF THE CHEBYSHEV EXPANSION IN FUNCTION
C     DGERFC(X,KODE,REL IERR)), BOTH OF WHICH ARE DOCUMENTED IN
C     FOLDER 16. DGERFC RETURNS RELATIVE ERRORS ON THE ORDER OF
C     O(10**(-13)) OR BETTER OVER THE FULL EXPONENT RANGE.
C
C     AS NOTED IN THE NUMERICAL COMMENTS ON THE ACCELERATED SERIES, ONE
C     CAN EXPECT GOOD RELATIVE ERRORS FOR X < XCRITICAL=DSQRT(N/e) WITH
C     DEGENERATION OF RELATIVE ACCURACY (SIGNIFICANT DIGITS) FOR VALUES
C     OF X LARGER THAN XCRITICAL. HERE N IS THE NUMBER OF APPLICATIONS
C     OF THE RECURRENCE TO OBTAIN THE ACCELERATED SERIES. THE S AND W
C     SEQUENCES ARE STORED IN DATA STATEMENTS AS SN(*) AND WN(*).
C
C     THE RESULTS OF THE COMPUTATION ARE STORED IN FILE DGSCOMP.TXT.
C-----------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION WN(10),EN(301),SN(10),S0(10),W(10)
      DATA (SN(I), I=1,10)/
     & 0.43551721806072D+01, 0.50323516580300D+00, 0.89336136051503D-01,
     & 0.14947622042659D-01, 0.22306483497165D-02, 0.29554277395613D-03,
     & 0.34977497160467D-04, 0.37284500519472D-05, 0.36090015081122D-06,
     & 0.31959542319209D-07/
      DATA (WN(I), I=1,10)/
     & 0.35517218060720D+00, 0.58790721358557D-01, 0.93361360515026D-02,
     & 0.13421798657880D-02, 0.17303518099223D-03, 0.20060680292218D-04,
     & 0.21043873642805D-05, 0.20111319127703D-06, 0.17625586083365D-07,
     & 0.14251165862602D-08/
      DATA RTPI/1.772453850905516D0/
      DATA TPI/6.28318530717958648D0/
      OPEN(UNIT=7,FILE="DGSCOMP.TXT")
C     N = NUMBER OF TERMS IN THE COEFFICIENTS OF EXP(-X**2) AND IERFC(X)
C       = NUMBER OF APPLICATIONS OF THE RECURRENCE TO ACCELERATE THE
C         CONVERGENCE OF THE SERIES.
      N=6
      P=1.0D0
      AJ=1.0D0
      DO 10 J=1,N
        P=P*AJ
        AJ=AJ+1.0D0
 10   CONTINUE
C     P= N FACTORIAL, AJ=N+1
      FACN=P
      FN=AJ-1.0D0
      FACJ=1.0D0
      FJ=1.0D0
      AJ=1.5D0
      S0(1)=4.0D0
      DO 11 J=2,N
        S0(J)=1.0D0/(FACJ*AJ*AJ)
        AJ=AJ+1.0D0
        FJ=FJ+1.0D0
```

```
        FACJ=FACJ*FJ
 11   CONTINUE
      WRITE(7,103)
 103  FORMAT(10X,"VERIFYING COMPUTATION OF WN(*)"/
     &'    J',12X,'WN(J)',11X,'REL ERR' )
      DO 12 J=1,N
        W(J)=SN(J)-S0(J)
        ERR=(W(J)-WN(J))/WN(J)
        WRITE(7,102) J,WN(J),ERR
 102    FORMAT(I5,D25.15,D12.4)
 12   CONTINUE
      XC=FN/DEXP(1.0D0)
      XC=DSQRT(XC)
      NXCRIT=INT(SNGL(XC*100.0D0))
      XCRIT=DBLE(FLOAT(NXCRIT))/100.0D0
      WRITE(7,101)XCRIT
 101  FORMAT(/"      XCRITICAL IS APPROX = ",D15.2//
     &"  N       X         G(X)      GSER(X)     REL ERR     ABS ERR   K
     &TERMS")
      DO 15 IX=1,25
C       GENERATE THE X VALUES TO BE USED IN THE COMPARISON
        X=DBLE(FLOAT(IX)/5.0D0)
C       GENERATE THE COEFFICIENT SUMS OF EXP(-X**2) AND IERFC(X)
        X2=X*X
        SS=0.0D0
        SW=0.0D0
        P=1.0D0
        DO 16 J=1,N
          SS=SS+P*S0(J)
          SW=SW+P*WN(J)
          P=-P*X2
 16     CONTINUE
C       P=(-X2)**N
        KODE=1
        CALL DIERFC(X,KODE,ANS,IERR)
C       SW=COEFFICIENT OF DEXP(-X**2)
C       SS*RTPI=COEFFICIENT OF IERFC(X)
        TRM1=DEXP(-X2)*SW
        TRM2=SS*RTPI*ANS
        SUM=TRM1+TRM2
        KODE=2
C       COMPUTE EXPONENTIAL INTEGRAL SUM
        FNH=1.5D0
        KODE=1
        M=200
        TOL=1.0D-13
C       M EXPONENTIAL INTEGRALS OF HALF ODD INTEGER ORDER IN EN(*)
        CALL DHEXINT(X2, FNH, KODE, M, TOL, EN, NZ, IERR)
        SS=0.0D0
        BK=0.5D0
        FHK=BK/FACN
        TOL=1.0D-12
        AK=FN+0.50D0
        FNK=FN
        DO 30 K=1,M
          SNK=FHK/(AK*AK)
          TRM=SNK*EN(K)
          SS=SS+TRM
          IF(DABS(TRM).LT.TOL*DABS(SS)) GOTO 31
          FNK=FNK+1.0D0
          BK=BK+1.0D0
          FHK=FHK*BK/FNK
          AK=AK+1.0D0
 30     CONTINUE
        WRITE(7,200)
 200    FORMAT(2X,"FELL THRU E(K+0.5,X**2) LOOP")
 31     CONTINUE
```

```
C       COMPUTE H(X)
        HX=SUM+P*SS
C       COMPUTE 0.5*E(1,X^2)
        NN=1
        KODE=1
        M=1
        TOL=1.0D-13
C       EXPONENTIAL INTEGRAL OF ORDER ONE IN EN(1)
        CALL DEXINT(X2, NN, KODE, M, TOL, EN, NZ, IERR)
        GSER=0.5D0*EN(1)-HX/TPI
        KODE=1
        REL=TOL
C       Y=G(X) FROM THE CHEBYSHEV EXPANSION
        Y=DGERFC(X,KODE,REL,IERR)
        ER=DABS(Y-GSER)
        IF(DABS(Y).GT.0.0D0) THEN
          ERR=ER/DABS(Y)
        ENDIF
        WRITE(7,300) N,X,Y,GSER,ERR,ER,K
 300    FORMAT(I3,5D12.4,I5)
 15   CONTINUE
      END
```

```
      PROGRAM SERSTEST
C     INITSX    INITIALIZES A SERIES TO A CONSTANT VALUE
C     PXSX      MULTIPLIES  A SERIES BY X**K
C     QXSX      DIVIDES     A SERIES BY X**K
C     DKSX      TAKES K DERIVATIVES (K<10 IF ZERO POWER HAS INDEX 10)
C     INTSX     INTEGRATES  A SERIES K TIMES ON (0,X) (FOR NON-NEGATIVE
C               EXPONENTS ONLY WITH THE ZERO POWER AT INDEX 10)
C     ADDSX     ADDS        TWO SERIES
C     SUBSX     SUBTRACTS   TWO SERIES
C     CMSX      MULTIPLIES  A SERIES BY A CONSTANT
C     CDSX      DIVIDES     A SERIES BY A CONSTANT
C     COPYSX    COPIES      A SERIES ARRAY TO ANOTHER ARRAY
C     WRTSX     WRITES OUT  AN ARRAY IN THE FORM OF A DATA STATEMENT
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION F(60),G(60),H(60)
      OPEN(UNIT=7,FILE="SERSTEST.TXT")
      CON=1.0D0
      CALL INITSX(F,CON)
      WRITE (7,100)
 100  FORMAT('INIT')
      CALL WRTSX(F,7)
C
      CALL PXSX(F,H,3)
      WRITE (7,200)
 200  FORMAT('PXSX K=3')
      CALL WRTSX(H,7)
C
      CALL QXSX(F,G,4)
      WRITE (7,300)
 300  FORMAT('QXSX K=4')
      CALL WRTSX(G,7)
C
      CALL DKSX(F,H,2)
      WRITE (7,400)
 400  FORMAT('DKSX K=2')
      CALL WRTSX(H,7)
C
      CALL INTSX(F,H,2)
      WRITE (7,500)
 500  FORMAT('INTSX K=2')
      CALL WRTSX(H,7)
C
      CALL INITSX(G,3.0D0)
      CALL ADDSX(F,G,H)
      WRITE (7,600)
 600  FORMAT('ADDSX ANS=4')
      CALL WRTSX(H,7)
C
      CALL SUBSX(F,G,H)
      WRITE (7,700)
 700  FORMAT('SUBSX ANS=-2')
      CALL WRTSX(H,7)
C
      CALL CMSX(F,H,5.0D0)
      WRITE (7,800)
 800  FORMAT('CMSX ANS=5')
      CALL WRTSX(H,7)
C
      CALL CDSX(F,H,4.0D0)
      WRITE (7,900)
 900  FORMAT('CDSX ANS=1/4')
      CALL WRTSX(H,7)
      END
      SUBROUTINE INFO
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
```

```
C      THE SUBROUTINES IN THIS FILE MANIPULATE DOUBLE PRECISION VECTORS
C      (DIMENSIONED BY 60) WHOSE VALUES ARE INTERPRETED AS COEFFIENTS OF
C      POWER SERIES. IF THE CONSTANT TERM IS SET AT SOME INDEX GREATER
C      THAN 1 (SAY 10), THE USUAL OPERATIONS (MULTIPLICATION BY X**K,
C      MULTIPLICATION CONSTANTS, DIVISION BY X**K, ETC.) WILL WORK FOR
C      NEGATIVE INDICES AS WELL UP TO POWERS OF -9. IF THIS INDEX IS 10,
C      THEN (INDEX-10) IS THE POWER ASSOCIATED WITH THAT COEFFICIENT. THE
C      LIST IS:
C
C                ALL ARRAYS (VECTORS) ARE DIMENSIONED BY 60
C      INITSX    INITIALIZES A SERIES TO A CONSTANT VALUE
C      PXSX      MULTIPLIES  A SERIES BY X**K
C      QXSX      DIVIDES     A SERIES BY X**K
C      DKSX      TAKES K DERIVATIVES (K<10 IF ZERO POWER HAS INDEX 10)
C      INTSX     INTEGRATES  A SERIES K TIMES ON (0,X) (FOR NON-NEGATIVE
C                EXPONENTS ONLY WITH THE ZERO POWER AT INDEX 10)
C      ADDSX     ADDS        TWO SERIES
C      SUBSX     SUBTRACTS   TWO SERIES
C      CMSX      MULTIPLIES  A SERIES BY A CONSTANT
C      CDSX      DIVIDES     A SERIES BY A CONSTANT
C      COPYSX    COPIES      A SERIES ARRAY TO ANOTHER ARRAY
C      WRTSX     WRITES OUT  AN ARRAY IN THE FORM OF A DATA STATEMENT
C
       RETURN
       END
       SUBROUTINE INITSX(TEMP,CON)
C      INTITIALIZE TEMP VECTOR TO A COMMON VALUE, TEMP=CON
       DIMENSION TEMP(60)
       IMPLICIT DOUBLE PRECISION (A-H,O-Z)
       DO 10 J=1,60
         TEMP(J)=CON
  10   CONTINUE
       RETURN
       END
       SUBROUTINE PXSX(TEMP,ANS,K)
C      MULTIPLY SERIES BY X**K
       DIMENSION TEMP(60),ANS1(60),ANS(60)
       IMPLICIT DOUBLE PRECISION (A-H,O-Z)
       DO 5 J=1,60
         ANS1(J)=0.0D0
  5    CONTINUE
       JL=60-K
       DO 10 J=1,JL
         ANS1(J+K)=TEMP(J)
  10   CONTINUE
       CALL COPYSX(ANS1,ANS)
       RETURN
       END
       SUBROUTINE QXSX(TEMP,ANS,K)
C      DIVIDE SERIES BY X**K
       DIMENSION TEMP(60),ANS1(60)
       IMPLICIT DOUBLE PRECISION (A-H,O-Z)
       DO 5 J=1,60
         ANS1(J)=0.0D0
  5    CONTINUE
       DO 10 J=1,60-K
         ANS1(J)=TEMP(J+K)
  10   CONTINUE
       CALL COPYSX(ANS1,ANS)
       RETURN
       END
       SUBROUTINE DKSX(TEMP,ANS,K)
C      TAKE K DERIVATIVES OF SERIES
       DIMENSION TEMP(60),ANS1(60),ANS(60)
       IMPLICIT DOUBLE PRECISION (A-H,O-Z)
       DO 5 J=1,60
         ANS1(J)=TEMP(J)
```

```
  5   CONTINUE
      DO 7 M=1,K
        AK=-8.0D0
        DO 10 J=2,60
          ANS1(J-1)=ANS1(J)*AK
          AK=AK+1.0D0
  10    CONTINUE
        ANS1(60)=0.0D0
  7   CONTINUE
      CALL COPYSX(ANS1,ANS)
      RETURN
      END
      SUBROUTINE INTSX(TEMP,ANS,K)
C     INTEGRATES A SERIES WITH NON-NEGATIVE POWERS ON (0,X). THE
C     INDEX FOR THE ZERO POWER TERM IS 10. ORDER OF DERIVATIVES IS
C     (INDEX-10) (50 TERMS MAX FOR ARRAYS DIMENSIONED BY 60).
      DIMENSION TEMP(60),ANS1(60),ANS(60)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DO 5 J=1,60
        ANS1(J)=TEMP(J)
  5   CONTINUE
      DO 7 M=1,K
        JJ=59
        AK=50.0D0
        DO 10 J=10,59
          ANS1(JJ+1)=ANS1(JJ)/AK
          JJ=JJ-1
          AK=AK-1.0D0
  10    CONTINUE
        ANS1(10)=0.0D0
  7   CONTINUE
      CALL COPYSX(ANS1,ANS)
      RETURN
      END
      SUBROUTINE CMSX(TEMP,ANS,CON)
C     MULTIPY SERIES BY CONSTANT
      DIMENSION TEMP(60),ANS(60)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DO 10 J=1,60
        ANS(J)=CON*TEMP(J)
  10  CONTINUE
      RETURN
      END
      SUBROUTINE CDSX(TEMP,ANS,CON)
C     DIVIDE SERIES BY CONSTANT
      DIMENSION TEMP(60),ANS(60)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DO 10 J=1,60
        ANS(J)=TEMP(J)/CON
  10  CONTINUE
      RETURN
      END
      SUBROUTINE ADDSX(TEMP1,TEMP2,ANS)
C     ADD TWO SERIES ANS=TEMP1 + TEMP2
      DIMENSION TEMP1(60),TEMP2(60),ANS1(60),ANS(60)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DO 10 J=1,60
        ANS1(J)=TEMP1(J)+TEMP2(J)
  10  CONTINUE
      CALL COPYSX(ANS1,ANS)
      RETURN
      END
      SUBROUTINE SUBSX(TEMP1,TEMP2,ANS)
C     SUBTRACT TWO SERIES ANS=TEMP1-TEMP2
      DIMENSION TEMP1(60),TEMP2(60),ANS1(60),ANS(60)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DO 10 J=1,60
```

```
        ANS1(J)=TEMP1(J)-TEMP2(J)
   10 CONTINUE
      CALL COPYSX(ANS1,ANS)
      RETURN
      END
      SUBROUTINE COPYSX(TEMP1,TEMP2)
C     COPY SERIES TEMP1 TO TEMP2
      DIMENSION TEMP1(60),TEMP2(60)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DO 10 J=1,60
        TEMP2(J)=TEMP1(J)
   10 CONTINUE
      RETURN
      END
      SUBROUTINE WRTSX(TEMP,NUNIT)
C     WRITE TEMP TO UNIT NUMBER NUNIT IN THE FORM OF A DATA
C     STATEMENT. THE MAIN PROGRAM MUST HAVE THE OPEN STATEMENT
C                OPEN(NUNIT,FILE='FILENAME')
C     WHICH ASSOCIATES THE UNIT NUMBER WITH THE OUTPUT FILE.
C
      DIMENSION TEMP(60)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      WRITE (NUNIT,100) (TEMP(J),J=1,60)
  100 FORMAT('     C',D21.15,',',D21.15,',',D21.15,',')
      RETURN
      END
***********************************************************************
      PROGRAM MPCHEBY
C
C     WRITTEN BY D.E.AMOS, DECEMBER,1972; MODIFIED MAY, 1990.
C
C     REFERENCE SANDIA NATIONAL LABORATORIES REPORT SC-DR-72 0917
C
C     ABSTRACT
C
C     PROGRAM MPCHEBY CALCULATES THE CHEBYSHEV COEFFICIENTS FOR A
C     MULTIPLE PRECISION FUNCTION MPGOFX(X) ON AN INTERVAL [A,B].
C
C     FINAL EXPANSION IS  S = 0.5*A(1)+SUM( A(J)*T(J,XX), J=2,N) WHERE
C
C          XX=(2*X-(A+B))/(B-A)   FOR    A.LE.X.LE.B
C                                 OR
C          XX=2*A/X-1.0           FOR    A.LE.X.LT.INFINITY,
C
C     T(J,XX) ARE CHEBYSHEV POLYNOMIALS AND A(J) ARE THE COEFFICIENTS
C     GENERATED BY THIS PROGRAM AND STORED IN A DATA STATEMENT IN A
C     FILE CALLED MPCHEBY.DAT.
C
C     DESCRIPTION OF ARGUMENTS
C
C     INPUT
C       A, B    = INTERVAL [A,B] END POINTS. IF A=B, THE EXPANSION
C                 IS TAKEN ON A TO INFINITY USING T(J,2*A/X-1)
C       N       = THE NUMBER OF COEFFICIENTS DESIRED (USUALLY N < 26)
C       NCHEBY  = NUMBER OF CHEBYSHEV NODES FOR THE APPROXIMATION
C                 NCHEBY IS MUCH LARGER THAN N  (USUALLY NCHEBY > 40)
C       IERR    = 1 GIVES A RELATIVE ERROR PRINT
C               = 0 GIVES AN ABSOLUTE ERROR PRINT
C       DNAME   = NAME OF THE COEFFICIENT ARRAY IN THE DATA STATEMENT
C       MPGOFX  = MULTIPLE PRECISION SUBROUTINE FOR THE FUNCTION BEING
C                 APPROXIMATED. SUBROUTINE MPGOFX(MX,MGOFX)
C     OUTPUT
C       FILE MPCHEBY.FIT SHOWING THE COEFFICIENTS AND ERRORS
C       FILE MPCHEBY.DAT CONTAINING THE DATA STATEMENT FOR THE
C                     COEFFICIENTS 0.5*A(1) THRU A(N)
C       FILE MPCHEBY.DAT IS CREATED AND USED AS A SCRATCH FILE
C
```

```
C     PROGRAM MPCHEBY USES MPGOFX AND DATAS PLUS ANY OTHERS RELATED TO
C     THE COMPUTATION OF GOFX
C-----------------------------------------------------------------------
      CHARACTER*2 NK,NAB
      CHARACTER*10 ILAB(2)
      CHARACTER*12 FNAME,DNAME,FWORK
C      CHARACTER*25 LINEA
      CHARACTER*80 LINE,LINEA
      CHARACTER*160 LTEMP,LBLANK
      DOUBLE PRECISION T,PI,DN,THETA,VK,VONE,TK,TONE,APB,BMA,X
      DOUBLE PRECISION Z,TZ,B1,B2,F,AA,BB,AR,TEMP,TEMP1
C
C-----------------------------------------------------------------------
      DOUBLE PRECISION DTOL,DH
      DOUBLE PRECISION DPMAX
C     SINGLE PRECISION SPMAX

      PARAMETER ( MXNDIG=256 , NBITS=32 ,
     *          LPACK  = (MXNDIG+1)/2 + 1 , LUNPCK = (6*MXNDIG)/5 + 20,
     *          LMWA   = 2*LUNPCK         , LJSUMS = 8*LUNPCK,
     *          LMBUFF = ((LUNPCK+3)*(NBITS-1)*301)/2000 + 6  )
C
C
      DIMENSION MTEMP1(200),MTEMP2(200),MAPB(200),MBMA(200),MAA(200),
     &MBB(200),MVK(200),MVONE(200),MTK(200),MTHETA(200),MZ(200),
     &MTZ(200),MB1(200),MB2(200)
      DIMENSION T(201),F(201),X(201),AR(201)
      DIMENSION MT(200,100),MF(200,100),MX(200,100),MAR(200,100),
     &MTEMP(200)

C
      COMMON /FMUSER/ NDIG,JBASE,JFORM1,JFORM2,KRAD,
     *                KW,NTRACE,LVLTRC,KFLAG,KWARN,KROUND
C
      COMMON /FM/ MWA(LMWA),NCALL,MXEXP,MXEXP2,KARGSW,KEXPUN,KEXPOV,
     *            KUNKNO,IUNKNO,RUNKNO,MXBASE,MXNDG2,KSTACK(19),
     *            MAXINT,SPMAX,DPMAX
C
      COMMON /FMSAVE/ NDIGPI,NJBPI,NDIGE,NJBE,NDIGLB,NJBLB,NDIGLI,NJBLI,
     *                MPISAV(LUNPCK),MESAV(LUNPCK),MLBSAV(LUNPCK),
     *                MLN1(LUNPCK),MLN2(LUNPCK),MLN3(LUNPCK),
     *                MLN4(LUNPCK)
C
      COMMON /CINIT/INIT
      COMMON /CWORK/MPI(200),MRTPI(200),MONE(200),MTWO(200),MTRTPI(200),
     *MZERO(200),MTOL(200),MDH(200),MNINE(200),NCDIG
      COMMON /GAUSS/ MX1(200),MX2(200),MX3(200),MX4(200),MW1(200),
     &MW2(200),MW3(200),MW4(200)
C
      DATA (ILAB(I),I=1,2)/' RELATIVE ',' ABSOLUTE '/

C-----------------------------------------------------------------------
      PRINT 950
  950 FORMAT(' APPROXIMATION INTERVAL A, B= ')
      READ *,AA,BB
      PRINT 951
  951 FORMAT(' NUMBER OF CHEBYSHEV COEFFICIENTS DESIRED (N~25), N= ')
      READ *,NAR
      PRINT 952
  952 FORMAT(' NUMBER OF CHEBYSHEV NODES (NCHEBY~40), NCHEBY= ')
      READ *,NCHEBY
      PRINT 953
  953 FORMAT(' NAME OF THE COEFFICIENTS IN THE DATA STATEMENT, DNAME= ')
      READ 900,DNAME
  900 FORMAT(A8)
      PRINT 954
  954 FORMAT(' 1 FOR RELATIVE ERROR; 2 FOR ABSOLUTE ERROR, IERR= ')
```

```
      READ *,IERR
      IF (IERR.NE.1 .AND. IERR.NE.2) THEN
        IERR=2
      ENDIF
C-----------------------------------------------------------------------
C     END INPUT
C-----------------------------------------------------------------------
      INIT=0
      IF (INIT.EQ.0) THEN
        NPREC=65
        CALL FPSET(NPREC)
C       FOR JBASE=10000, NDIG=NPREC/4+2
        NDIG=INT(NPREC/4)+5
        NCDIG=NDIG
        K=0
        CALL FPI2M(K,MZERO)
        K=1
        CALL FPI2M(K,MONE)
        K=2
        CALL FPI2M(K,MTWO)
        K=9
        CALL FPI2M(K,MNINE)
        CALL FPPI(MPI)
        CALL FPM2DP(MPI,PI)
        CALL FPSQRT(MPI,MRTPI)
C       MRTPI=SQRT(PI)
        CALL FPDIV(MONE,MRTPI,MTEMP)
        CALL FPADD(MTEMP,MTEMP,MTRTPI)
C       MTRTPI=2/RTPI
        DTOL=0.50D-19
        CALL FPDP2M(DTOL,MTOL)
        I=2
        CALL FPDIVI(MONE,I,MDH)
C       INIT=1
C       INIT IS SET TO 1 AFTER THE INITIALIZATION OF MQUAD8
      ENDIF
      N1 = NCHEBY-1
      N2 = NCHEBY-1
      APB=AA+BB
      BMA=BB-AA
      CALL FPDP2M(AA,MAA)
      CALL FPDP2M(BB,MBB)
      CALL FPADD(MAA,MBB,MAPB)
      CALL FPSUB(MBB,MAA,MBMA)
      C1= SNGL(APB)
      C2= SNGL(BMA)
      IF(BB.EQ.AA) C1=SNGL(AA+AA)
      OPEN(10,FILE='MPCHEBY.FIT',STATUS='UNKNOWN')
      DO 500 N=N1,N2
        IF (NAR.GT.N) THEN
          NAR=N+1
        ENDIF
        NP1=N+1
        DN=DBLE(FLOAT(N))
        THETA=PI/DN
        CALL FPDIVI(MPI,N,MTHETA)
        T(1)=1.0D0
        CALL COPYTO(MONE,1,MT)
        T(2)=DCOS(THETA)
        CALL FPCOS(MTHETA,MTEMP)
        CALL COPYTO(MTEMP,2,MT)
        CALL FPEQU(MTEMP,MTONE,NDIG,NDIG)
        TONE=T(2)
        VONE=DSIN(THETA)
        CALL FPSIN(MTHETA,MVONE)
        VK = VONE
        CALL FPEQU(MVONE,MVK,NDIG,NDIG)
```

```
C-----------------------------------------------------------------
C     GET CHEBYSHEV NODES ON [A,B] IN X ARRAY
C     TRIG IDENTITIES SIN((K+1)*X) = SIN(K*X)*COS(X)+COS(K*X)*SIN(X)
C                     COS((K+1)*X) = COS(K*X)*COS(X)-SIN(K*X)*SIN(X)
C     FOR CHEBYSHEV NODES T(K)=COS((K-1)*PI/N), K=1,N+1 BY RECURRENCE
C-----------------------------------------------------------------
        DO 5 K=2,N
          TK=T(K)
          CALL COPYFR(MT,K,MTK)
          T(K+1)=TONE*TK-VK*VONE
          CALL FPMPY(MTONE,MTK,MTEMP)
          CALL FPMPY(MVK,MVONE,MTEMP1)
          CALL FPSUB(MTEMP,MTEMP1,MTEMP)
          CALL COPYTO(MTEMP,K+1,MT)
          VK=VK*TONE+TK*VONE
          CALL FPMPY(MVK,MTONE,MTEMP)
          CALL FPMPY(MTK,MVONE,MTEMP1)
          CALL FPADD(MTEMP,MTEMP1,MVK)
          X(K)=(TK*BMA+APB)/2.0D0
          CALL FPMPY(MTK,MBMA,MTEMP)
          CALL FPADD(MTEMP,MAPB,MTEMP)
          CALL FPDIVI(MTEMP,2,MTEMP)
          CALL COPYTO(MTEMP,K,MX)
          CALL FPM2DP(MTEMP,TEMP)
          PRINT *,'K, X(K) ',K,TEMP
C          IF(AA.NE.BB) GO TO 5
C          IF(TK.NE.-1.0D0) GO TO 10
C          X(K)=1.0D+16
C          GO TO 5
C   10     CONTINUE
C          X(K)=(AA+AA)/(1.0D0+TK)
    5   CONTINUE
        X(1)=(T(1)*BMA+APB)/2.0D0
        CALL COPYFR(MT,1,MTEMP)
        CALL FPMPY(MTEMP,MBMA,MTEMP)
        CALL FPADD(MTEMP,MAPB,MTEMP)
        CALL FPDIVI(MTEMP,2,MTEMP)
        CALL COPYTO(MTEMP,1,MX)
        X(NP1)=(T(NP1)*BMA+APB)/2.0D0
        CALL COPYFR(MT,NP1,MTEMP)
        CALL FPMPY(MTEMP,MBMA,MTEMP)
        CALL FPADD(MTEMP,MAPB,MTEMP)
        CALL FPDIVI(MTEMP,2,MTEMP)
        CALL COPYTO(MTEMP,NP1,MX)
C        IF(AA.NE.BB) GO TO 20
C        X(1)=(AA+AA)/(1.0D0+T(1))
C        X(NP1)=1.0D+16
C   20   CONTINUE
C-----------------------------------------------------------------
        WRITE(10, 52)NP1,AA,BB
   52 FORMAT(' NODES='I5/' A    ='D23.14/' B    ='D23.14/)
        WRITE(10, 51)(X(I),I=1,NP1)
   51 FORMAT(' CHEBYSHEV NODES  X(J)'/(1X,3D23.14))
C-----------------------------------------------------------------
C     FUNCTION EVALUATION AT CHEBYSHEV NODES X(I), I=1,NP1
C-----------------------------------------------------------------
        DO 7 I=1,NP1
          CALL COPYFR(MX,I,MTEMP)
          CALL FPM2DP(MTEMP,TEMP1)
          CALL MPGOFX(MTEMP,MTEMP1)
          CALL COPYTO(MTEMP1,I,MF)
          CALL FPM2DP(MTEMP1,TEMP)
          F(I)=TEMP
          PRINT *,'I,X(I),F(I)',I,TEMP1,TEMP
    7   CONTINUE
        F(NP1)=TEMP/2.0D0
        CALL FPDIVI(MTEMP1,2,MTEMP)
```

```
        CALL COPYTO(MTEMP,NP1,MF)
C-----------------------------------------------------------------------
C     END OF FUNCTION EVALUATION
C-----------------------------------------------------------------------
        DO 100 IR=1,NP1
C          Z=T(IR)
          CALL COPYFR(MT,IR,MZ)
C          TZ=Z+Z
          CALL FPMPYI(MZ,2,MTZ)
C          B2=0.0D0
          CALL FPEQU(MZERO,MB2,NDIG,NDIG)
C          B1= 0.0D0
          CALL FPEQU(MZERO,MB1,NDIG,NDIG)
          DO 15 J=1,N
            K=NP1-J+1
            TEMP=B1
            CALL FPEQU(MB1,MTEMP,NDIG,NDIG)
C            B1=TZ*B1-B2+F(K)
            CALL COPYFR(MF,K,MTEMP1)
            CALL FPMPY(MTZ,MB1,MTEMP2)
            CALL FPSUB(MTEMP2,MB2,MTEMP2)
            CALL FPADD(MTEMP2,MTEMP1,MB1)
C            B2=TEMP
            CALL FPEQU(MTEMP,MB2,NDIG,NDIG)
   15     CONTINUE
C          ANS=Z*B1-B2+F(1)/2.0D0
          CALL COPYFR(MF,1,MTEMP)
          CALL FPDIVI(MTEMP,2,MTEMP)
          CALL FPMPY(MZ,MB1,MTEMP1)
          CALL FPSUB(MTEMP1,MB2,MTEMP1)
          CALL FPADD(MTEMP1,MTEMP,MTEMP)
C          AR(IR)=(ANS+ANS)/DN
          CALL FPMPYI(MTEMP,2,MTEMP)
          CALL FPDIVI(MTEMP,N,MTEMP)
          CALL COPYTO(MTEMP,IR,MAR)
          CALL FPM2DP(MTEMP,AR(IR))
  100   CONTINUE
C-----------------------------------------------------------------------
        WRITE(10,153) (AR(K),K=1,NP1)
  153 FORMAT(' CHEBYSHEV COEFFICIENTS A(J)'/(1X,3D23.14))
C-----------------------------------------------------------------------
        F(NP1)=F(NP1)+F(NP1)
        CALL COPYFR(MF,NP1,MTEMP)
        CALL FPMPYI(MTEMP,2,MTEMP)
        CALL COPYTO(MTEMP,NP1,MF)
C-----------------------------------------------------------------------
        WRITE(10, 54)(F(I),I=1,NP1)
   54 FORMAT(' FUNCTION VALUES AT CHEBYSHEV NODES F(X(J))'/(1X,3D23.14))
C-----------------------------------------------------------------------
        NARM=NAR-1
        DO 30 I=1,NP1
          XX=SNGL(X(I))
          IF(AA.NE.BB) GO TO 26
          SZ=C1/XX-1.0E0
          GO TO 27
   26     CONTINUE
          SZ=(XX+XX-C1)/C2
   27     CONTINUE
          STZ=SZ+SZ
          S2=0.0E0
          S1=0.0E0
          DO 25 J=1,NARM
            K=NAR-J+1
            ST=S1
            SA=AR(K)
            S1=STZ*S1-S2+SA
            S2=ST
```

```
   25      CONTINUE
           SA=AR(1)
           FX=SZ*S1-S2+SA/2.0
           T(I)=DBLE(FX)-F(I)
           IF(IERR.EQ.1) THEN
             IF(F(I).NE.0.0D0) THEN
               T(I)=T(I)/F(I)
             ENDIF
           ENDIF
   30    CONTINUE
C-----------------------------------------------------------------------
         WRITE(10, 53)ILAB(IERR),NAR,AA,BB,(T(I),I=1,NP1)
   53    FORMAT(/' SINGLE PRECISION',A10,'ERRORS AT CHEBYSHEV NODES USING
      &',I5,' TERMS'/' OF THE CHEBYSHEV SUM ON',2D23.14//(1X,3D23.14))
C-----------------------------------------------------------------------
         DO 35 I=1,NP1
           IF(AA.NE.BB) GO TO 36
           Z=(AA+AA)/X(I)-1.0D0
           GO TO 37
   36      CONTINUE
           Z=(X(I)+X(I)-APB)/BMA
   37      CONTINUE
           TZ=Z+Z
           B2=0.0D0
           B1=0.0D0
           DO 40 J=1,NARM
             K=NAR-J+1
             TEMP=B1
             CALL COPYFR(MAR,K,MTEMP)
             CALL FPM2DP(MTEMP,TEMP1)
C             B1=TZ*B1-B2+AR(K)
             B1=TZ*B1-B2+TEMP1
             B2=TEMP
   40      CONTINUE
           CALL COPYFR(MAR,1,MTEMP)
           CALL FPM2DP(MTEMP,TEMP)
C           TEMP=Z*B1-B2+AR(1)/2.0D0
           TEMP=Z*B1-B2+TEMP/2.0D0
           CALL COPYFR(MF,I,MTEMP)
           CALL FPM2DP(MTEMP,TEMP1)
C           T(I)=TEMP-F(I)
           T(I)=TEMP-TEMP1
           IF(IERR.EQ.1) THEN
             IF(F(I).NE.0.0D0) THEN
C                T(I)=T(I)/F(I)
                T(I)=T(I)/TEMP1
             ENDIF
           ENDIF
   35    CONTINUE
C-----------------------------------------------------------------------
         WRITE(10, 55)ILAB(IERR),NAR,AA,BB,(T(I),I=1,NP1)
   55    FORMAT(/' DOUBLE PRECISION',A10,'ERRORS AT CHEBYSHEV NODES USING
      &',I5,' TERMS'/' OF THE CHEBYSHEV SUM ON',2D23.14//(1X,3D23.14))
C-----------------------------------------------------------------------
  500  CONTINUE
       CLOSE(10)
       FNAME='MPCHEBY.DAT'
       FWORK='MPCHEBZ.DAT'
       AR(1)=AR(1)*0.5D0
       CALL COPYFR(MAR,1,MTEMP)
       CALL FPDIVI(MTEMP,2,MTEMP)
       CALL COPYTO(MTEMP,1,MAR)
       OPEN(10,FILE='MPCHEBY.DAT',STATUS='UNKNOWN')
       REWIND(10)
       WRITE(10,531)NP1
 531   FORMAT('NODES= ',I3)
       WRITE(10,530) AA,BB
```

```
 530  FORMAT('A= 'D30.18/'B= ',D30.18/)
      CLOSE(10)
C-----------------------------------------------------------------------
C     WRITE NORMAL DATA STATEMENT WITH 14 DIGITS, 3 ON A LINE
C-----------------------------------------------------------------------
      CALL DATAS(NAR,AR,DNAME,FNAME,FWORK,KERR)
C-----------------------------------------------------------------------
C     WRITE DATA STATEMENT IN 2 COLUMNS OF 18 DIGITS
C-----------------------------------------------------------------------
      DO 515 I=1,10
        IF(DNAME(I:I).NE.' ') GO TO 520
  515 CONTINUE
      PRINT *, 'THE DATA NAME IS BLANK'
      STOP
  520 CONTINUE
      K1=I
      DO 501 I=K1,10
        IF(DNAME(I:I).EQ.' ') GO TO 502
 501  CONTINUE
      I=11
 502  CONTINUE
      K2=I-1
C     READ TO THE BOTTOM OF THE FILE (TO SKIP WHAT IS ALREADY THERE)
      OPEN(15,FILE=FNAME,STATUS='UNKNOWN')
      REWIND(15)
      DO 511 LOOP=1,400
        READ(15,'( )',END=512)
 511  CONTINUE
 512  CONTINUE
      ENDFILE(15)
      BACKSPACE(15)
      OPEN(20,FILE=FWORK,STATUS='UNKNOWN')
      WRITE(20,509) NAR
 509  FORMAT(I2)
      REWIND(20)
      READ(20,510) NAB
 510  FORMAT(A2)
      REWIND (20)
      LINE='      DATA ('//DNAME(K1:K2)//'(I), I=1,'//NAB//')/'
      WRITE(15,506)LINE
 506  FORMAT(A72)
      LBLANK='                                                                        '
      LL=72
      JFORM1=0
      K=1
      NMOD=MOD(NAR,2)
      IF(NMOD.EQ.0) THEN
        NJ=NAR-1
      ELSE
        NJ=NAR-2
      ENDIF
      DO 300 J=1,NJ,2
        WRITE(20,509) K
        REWIND(20)
        READ(20,510) NK
        REWIND(20)
        CALL COPYFR(MAR,J,MTEMP)
        LTEMP='    '//NK//'    '
        CALL FPOUT(MTEMP,LINEA,LL)
        I1=NPREC+4
        I2=I1+3
        LINEA=LINEA(1:22)//'D'//LINEA(I1:I2)//',  '
        LTEMP=LTEMP(1:8)//LINEA(1:30)
        CALL COPYFR(MAR,J+1,MTEMP)
        CALL FPOUT(MTEMP,LINEA,LL)
        IF (J.EQ.NJ) THEN
          IF(NMOD.EQ.0) THEN
```

```
            LINEA=LINEA(1:22)//'D'//LINEA(I1:I2)//'/'
          ELSE
            LINEA=LINEA(1:22)//'D'//LINEA(I1:I2)//','
          ENDIF
        ELSE
          LINEA=LINEA(1:22)//'D'//LINEA(I1:I2)//','
        ENDIF
        LTEMP=LTEMP(1:38)//LINEA(1:28)//LBLANK
        LINE=LTEMP(1:72)
        WRITE(15,301)LINE
 301    FORMAT(A72)
        K=K+1
 300  CONTINUE
      IF(NMOD.NE.0) THEN
        WRITE(20,509) K
        REWIND(20)
        READ(20,510) NK
        REWIND(20)
        CALL COPYFR(MAR,NAR,MTEMP)
        LTEMP='     '//NK//'     '
        CALL FPOUT(MTEMP,LINEA,LL)
        LINEA=LINEA(1:22)//'D'//LINEA(64:67)//'/  '
        LTEMP=LTEMP(1:8)//LINEA(1:28)//LBLANK
        LINE=LTEMP(1:72)
        WRITE(15,301)LINE
      ENDIF
      CLOSE(15)
C-----------------------------------------------------------------------
      IF (KERR.NE.0) THEN
        PRINT *,' ERROR IN DATAS, IERR = ',KERR
      ELSE
        PRINT *,'  '
        PRINT *,' APPROXIMATION USING ',NP1,' CHEBYSHEV NODES'
        PRINT *,'  '
        PRINT *,' FILES WRITTEN:'
        PRINT *,'  '
        PRINT *,' MPCHEBY.FIT:    THE COEFFIENTS, NODES, AND ERRORS,'
        PRINT *,' MPCHEBY.DAT:    THE DATA STATEMENT,'
        PRINT *,' MPCHEBZ.DAT:    SCRATCH FILE.'
      ENDIF
      PAUSE
      STOP
      END
      SUBROUTINE COPYTO(MA,K,MB)
C     COPY VECTOR MA to KTH COLUMN OF MATRIX MB
      DIMENSION MB(200,100),MA(200)
      DO 10 J=1,200
        MB(J,K)=MA(J)
 10   CONTINUE
      RETURN
      END
      SUBROUTINE COPYFR(MB,K,MA)
C     COPY from KTH COLUMN OF MATRIX MB TO VECTOR MA
      DIMENSION MB(200,100),MA(200)
      DO 10 J=1,200
        MA(J)=MB(J,K)
 10   CONTINUE
      RETURN
      END
      SUBROUTINE MPGOFXX(MX,MGOFX)
      DIMENSION MX(200),MGOFX(200)
      COMMON /CWORK/MPI(200),MRTPI(200),MONE(200),MTWO(200),MTRTPI(200),
     *MZERO(200),MTOL(200),MDH(200),MNINE(200),NCDIG
      CALL FPDIV(MONE,MX,MGOFX)
      RETURN
      END
      SUBROUTINE MPGOFXY(MT,MFUN)
```

```
C     COMPUTES FOR ERF(X)
      LOGICAL FPCOMP
      DIMENSION MT(200),MT2(200),MTRM(200),MSUM(200),MCI(200),MCNT(200),
     &MTRM1(200),MTRMA(200),MTRMS(200),MTEMP(200),MSUMA(200),MAK(200),
     &MFUN(200),MSUMT(200)
      COMMON /CWORK/MPI(200),MRTPI(200),MONE(200),MTWO(200),MTRTPI(200),
     *MZERO(200),MTOL(200),MDH(200),MNINE(200),NCDIG
C
C     CALCULATE ERF ON [0,2]
C
      NDIG=NCDIG
      CALL FPMPY(MT,MT,MT2)
      CALL FPEQU(MONE,MTRM,NDIG,NDIG)
      CALL FPEQU(MONE,MSUM,NDIG,NDIG)
      J=-1
      IF(FPCOMP(MT,"LT",MNINE)) THEN
        CALL FPEQU(MONE,MCI,NDIG,NDIG)
        CALL FPEQU(MTWO,MCNT,NDIG,NDIG)
        CALL FPADD(MONE,MCNT,MCNT)
C         CALL FPDIV(MONE,MT,MTRM1)
        DO 10 I=1,1000
          CALL FPMPY(MTRM,MT2,MTRM)
          CALL FPDIV(MTRM,MCI,MTRM)
          CALL FPMPYI(MTRM,J,MTRM)
          CALL FPDIV(MTRM,MCNT,MTRMS)
          CALL FPADD(MSUM,MTRMS,MSUM)
          IF(I.LT.5) GOTO 5
          CALL FPMPY(MSUM,MTRTPI,MSUMT)
          CALL FPABS(MTRMS,MTRMA)
          CALL FPABS(MSUMT,MSUMA)
          CALL FPMPY(MTOL,MSUMA,MSUMA)
          IF (FPCOMP(MTRMA,'LT',MSUMA)) GOTO 15
  5       CONTINUE
          CALL FPADD(MCI,MONE,MCI)
          CALL FPADD(MCNT,MTWO,MCNT)
  10    CONTINUE
        PRINT*,"DROP THRU SERIES LOOP"
        PAUSE
  15    CONTINUE
      ELSE
        CALL FPEQU(MDH,MAK,NDIG,NDIG)
        DO 20 I=1,25
          CALL FPMPY(MTRM,MAK,MTRM)
          CALL FPDIV(MTRM,MT2,MTRM)
          CALL FPABS(MTRM,MTRMA)
          IF (FPCOMP(MTRMA,'LE',MTOL)) GOTO 25
          CALL FPMPYI(MTRM,J,MTRM)
          CALL FPADD(MSUM,MTRM,MSUM)
          CALL FPADD(MAK,MONE,MAK)
  20    CONTINUE
        PRINT*,"DROP THRU ASYMPTOTOIC SERIES LOOP"
        PAUSE
  25    CONTINUE
        CALL FPMPYI(MT2,J,MAK)
        CALL FPEXP(MAK,MTRM)
        CALL FPMPY(MSUM,MTRM,MSUM)
        CALL FPMPY(MRTPI,MT,MAK)
        CALL FPDIV(MSUM,MAK,MSUMT)
      ENDIF
      CALL FPEQU(MSUMT,MFUN,NDIG,NDIG)
      RETURN
      END
      SUBROUTINE MPGOFX(MX,MGOFX)
C
C     COMPUTES INT ON (X,INF) OF ERFC(X)/X IN MP ARITHMETIC USING
C     GAUS8 FORMULA ON REPEATED SUBDIVIDED INTERVALS
C
```

```
C      DOUBLE PRECISION DSUM
      LOGICAL FPCOMP
      DIMENSION MSUM(200),MSUMA(200),MANS(200),MANSA(200),MTOLA(200),
     &MTEMP(200),MA(200),MB(200),MX(200),MSUM1(200),MDX(200)
C
      DIMENSION MGOFX(200)
C
      COMMON /CWORK/MPI(200),MRTPI(200),MONE(200),MTWO(200),MTRTPI(200),
     *MZERO(200),MTOL(200),MDH(200),MNINE(200),NCDIG
C
      NDIG=NCDIG
      KL=10
      CALL FPMPYI(MTOL,KL,MTOLA)
      CALL FPEQU(MDH,MDX,NDIG,NDIG)
      CALL FPEQU(MZERO,MSUM1,NDIG,NDIG)
      KL=100
      DO 10 J=1,100
        CALL FPDIVI(MDX,2,MDX)
        KL=KL+KL
        CALL FPEQU(MZERO,MSUM,NDIG,NDIG)
        CALL FPEQU(MX,MB,NDIG,NDIG)
        DO 20 K=1,KL
          CALL FPEQU(MB,MA,NDIG,NDIG)
          CALL FPADD(MA,MDX,MB)
          CALL MQUAD8(MA,MB,MANS)
          CALL FPADD(MSUM,MANS,MSUM)
          IF(J.EQ.1) THEN
            CALL FPABS(MANS,MANSA)
            CALL FPABS(MSUM,MSUMA)
            CALL FPMPY(MSUMA,MTOL,MSUMA)
            IF (FPCOMP(MANSA,'LT',MSUMA)) GOTO 25
          ENDIF
 20     CONTINUE
 25     CONTINUE
        IF(J.EQ.1) THEN
          KL=K
          CALL FPEQU(MSUM,MSUM1,NDIG,NDIG)
          GOTO 10
        ENDIF
        CALL FPSUB(MSUM,MSUM1,MTEMP)
        CALL FPABS(MTEMP,MTEMP)
        CALL FPABS(MSUM,MSUMA)
        CALL FPMPY(MSUMA,MTOLA,MSUMA)
        IF (FPCOMP(MTEMP,'LE',MSUMA)) GOTO 30
        CALL FPEQU(MSUM,MSUM1,NDIG,NDIG)
C        CALL FPM2DP(MSUM,DSUM)
C        PRINT *,KL,DSUM
 10   CONTINUE
 30   CONTINUE
      CALL FPMPY(MX,MX,MTEMP)
      CALL FPEXP(MTEMP,MB)
C      CALL FPMPY(MTEMP,MX,MA)
C      CALL FPMPY(MA,MB,MTEMP)
      CALL FPMPY(MTEMP,MB,MTEMP)
      CALL FPMPY(MTEMP,MSUM,MGOFX)
      RETURN
      END
      SUBROUTINE MQUAD8(MA,MB,MQUAD)
      DIMENSION MA(200),MB(200),MAPB(200),MBMA(200),MTEMP1(200),
     &MTEMP2(200),MTEMP3(200),MTEMP4(200),MTEMP5(200),MTEMP6(200),
     &MTEMP7(200),MTEMP8(200),MSUM(200),MFUN(200),MQUAD(200)
      DIMENSION IX1(4),IX2(4),IX3(4),IX4(4),IW1(4),IW2(4),IW3(4),IW4(4)
C
      COMMON /CINIT/INIT
      COMMON /CWORK/MPI(200),MRTPI(200),MONE(200),MTWO(200),MTRTPI(200),
     *MZERO(200),MTOL(200),MDH(200),MNINE(200),NCDIG
      COMMON /GAUSS/ MX1(200),MX2(200),MX3(200),MX4(200),MW1(200),
```

```
     &MW2(200),MW3(200),MW4(200)
C
C      GAUSS ABSCISSAS AND WEIGHTS IN MP
      DATA (IX1(J),J=1,4)/0,18343464,24956498,05000000/
      DATA (IX2(J),J=1,4)/0,52553240,99163289,86000000/
      DATA (IX3(J),J=1,4)/0,79666647,74136267,40000000/
      DATA (IX4(J),J=1,4)/0,96028985,64975362,32000000/
      DATA (IW1(J),J=1,4)/0,36268378,33783619,83000000/
      DATA (IW2(J),J=1,4)/0,31370664,58778872,87000000/
      DATA (IW3(J),J=1,4)/0,22238103,44533744,71000000/
      DATA (IW4(J),J=1,4)/0,10122853,62903762,59000000/
C
      NDIG=NCDIG
      IF (INIT.EQ.0) THEN
        CALL FPEQU(MZERO,MX1,NDIG,NDIG)
        MX1(1)=IX1(1)
        MX1(2)=IX1(2)
        MX1(3)=IX1(3)
        MX1(4)=IX1(4)
        CALL FPEQU(MZERO,MX2,NDIG,NDIG)
        MX2(1)=IX2(1)
        MX2(2)=IX2(2)
        MX2(3)=IX2(3)
        MX2(4)=IX2(4)
        CALL FPEQU(MZERO,MX3,NDIG,NDIG)
        MX3(1)=IX3(1)
        MX3(2)=IX3(2)
        MX3(3)=IX3(3)
        MX3(4)=IX3(4)
        CALL FPEQU(MZERO,MX4,NDIG,NDIG)
        MX4(1)=IX4(1)
        MX4(2)=IX4(2)
        MX4(3)=IX4(3)
        MX4(4)=IX4(4)
        CALL FPEQU(MZERO,MW1,NDIG,NDIG)
        MW1(1)=IW1(1)
        MW1(2)=IW1(2)
        MW1(3)=IW1(3)
        MW1(4)=IW1(4)
        CALL FPEQU(MZERO,MW2,NDIG,NDIG)
        MW2(1)=IW2(1)
        MW2(2)=IW2(2)
        MW2(3)=IW2(3)
        MW2(4)=IW2(4)
        CALL FPEQU(MZERO,MW3,NDIG,NDIG)
        MW3(1)=IW3(1)
        MW3(2)=IW3(2)
        MW3(3)=IW3(3)
        MW3(4)=IW3(4)
        CALL FPEQU(MZERO,MW4,NDIG,NDIG)
        MW4(1)=IW4(1)
        MW4(2)=IW4(2)
        MW4(3)=IW4(3)
        MW4(4)=IW4(4)
        PRINT *,'GAUSSIAN ABSCISSAS AND WEIGHTS '
        CALL FPPRNT(MX1)
        CALL FPPRNT(MX2)
        CALL FPPRNT(MX3)
        CALL FPPRNT(MX4)
        CALL FPPRNT(MW1)
        CALL FPPRNT(MW2)
        CALL FPPRNT(MW3)
        CALL FPPRNT(MW4)
        PAUSE
        INIT=1
      ENDIF
      CALL FPADD(MA,MB,MAPB)
```

```
      CALL FPDIVI(MAPB,2,MAPB)
      CALL FPSUB(MB,MA,MBMA)
      CALL FPDIVI(MBMA,2,MBMA)
      CALL FPMPY(MX1,MBMA,MTEMP1)
      CALL FPMPY(MX2,MBMA,MTEMP2)
      CALL FPMPY(MX3,MBMA,MTEMP3)
      CALL FPMPY(MX4,MBMA,MTEMP4)
      I=-1
      CALL FPMPYI(MTEMP1,I,MTEMP5)
      CALL FPMPYI(MTEMP2,I,MTEMP6)
      CALL FPMPYI(MTEMP3,I,MTEMP7)
      CALL FPMPYI(MTEMP4,I,MTEMP8)
C
      CALL FPADD(MTEMP1,MAPB,MTEMP1)
      CALL FPADD(MTEMP2,MAPB,MTEMP2)
      CALL FPADD(MTEMP3,MAPB,MTEMP3)
      CALL FPADD(MTEMP4,MAPB,MTEMP4)
      CALL FPADD(MTEMP5,MAPB,MTEMP5)
      CALL FPADD(MTEMP6,MAPB,MTEMP6)
      CALL FPADD(MTEMP7,MAPB,MTEMP7)
      CALL FPADD(MTEMP8,MAPB,MTEMP8)
C
      CALL FPEQU(MZERO,MSUM,NDIG,NDIG)
      CALL MFOFX(MTEMP1,MFUN)
      CALL FPMPY(MW1,MFUN,MFUN)
      CALL FPADD(MSUM,MFUN,MSUM)
      CALL MFOFX(MTEMP2,MFUN)
      CALL FPMPY(MW2,MFUN,MFUN)
      CALL FPADD(MSUM,MFUN,MSUM)
      CALL MFOFX(MTEMP3,MFUN)
      CALL FPMPY(MW3,MFUN,MFUN)
      CALL FPADD(MSUM,MFUN,MSUM)
      CALL MFOFX(MTEMP4,MFUN)
      CALL FPMPY(MW4,MFUN,MFUN)
      CALL FPADD(MSUM,MFUN,MSUM)
      CALL MFOFX(MTEMP5,MFUN)
      CALL FPMPY(MW1,MFUN,MFUN)
      CALL FPADD(MSUM,MFUN,MSUM)
      CALL MFOFX(MTEMP6,MFUN)
      CALL FPMPY(MW2,MFUN,MFUN)
      CALL FPADD(MSUM,MFUN,MSUM)
      CALL MFOFX(MTEMP7,MFUN)
      CALL FPMPY(MW3,MFUN,MFUN)
      CALL FPADD(MSUM,MFUN,MSUM)
      CALL MFOFX(MTEMP8,MFUN)
      CALL FPMPY(MW4,MFUN,MFUN)
      CALL FPADD(MSUM,MFUN,MSUM)
      CALL FPMPY(MSUM,MBMA,MQUAD)
      RETURN
      END
      SUBROUTINE MFOFX(MT,MFUN)
C       DOUBLE PRECISION DSUM,DT
      LOGICAL FPCOMP
      DIMENSION MT(200),MT2(200),MTRM(200),MSUM(200),MCI(200),MCNT(200),
     &MTRM1(200),MTRMA(200),MTRMS(200),MTEMP(200),MSUMA(200),MAK(200),
     &MFUN(200),MSUMT(200)
      COMMON /CWORK/MPI(200),MRTPI(200),MONE(200),MTWO(200),MTRTPI(200),
     *MZERO(200),MTOL(200),MDH(200),MNINE(200),NCDIG
C
      NDIG=NCDIG
      CALL FPMPY(MT,MT,MT2)
      CALL FPEQU(MONE,MTRM,NDIG,NDIG)
      CALL FPEQU(MONE,MSUM,NDIG,NDIG)
      J=-1
      IF(FPCOMP(MT,"LT",MNINE)) THEN
        CALL FPEQU(MONE,MCI,NDIG,NDIG)
        CALL FPEQU(MTWO,MCNT,NDIG,NDIG)
```

```
        CALL FPADD(MONE,MCNT,MCNT)
C        CALL FPDIV(MONE,MT,MTRM1)
        DO 10 I=1,1000
          CALL FPMPY(MTRM,MT2,MTRM)
          CALL FPDIV(MTRM,MCI,MTRM)
          CALL FPMPYI(MTRM,J,MTRM)
          CALL FPDIV(MTRM,MCNT,MTRMS)
          CALL FPADD(MSUM,MTRMS,MSUM)
          IF(I.LT.5) GOTO 5
          CALL FPMPY(MSUM,MTRTPI,MTEMP)
          CALL FPMPY(MTEMP,MT,MTEMP)
C          CALL FPSUB(MTRM1,MTEMP,MSUMT)
          CALL FPSUB(MONE,MTEMP,MSUMT)
          CALL FPABS(MTRMS,MTRMA)
          CALL FPABS(MSUMT,MSUMA)
          CALL FPMPY(MTOL,MSUMA,MSUMA)
          IF (FPCOMP(MTRMA,'LT',MSUMA)) GOTO 15
  5       CONTINUE
          CALL FPADD(MCI,MONE,MCI)
          CALL FPADD(MCNT,MTWO,MCNT)
  10    CONTINUE
        PRINT*,"DROP THRU SERIES LOOP"
        PAUSE
  15    CONTINUE
      ELSE
        CALL FPEQU(MDH,MAK,NDIG,NDIG)
        DO 20 I=1,25
          CALL FPMPY(MTRM,MAK,MTRM)
          CALL FPDIV(MTRM,MT2,MTRM)
          CALL FPABS(MTRM,MTRMA)
          CALL FPABS(MSUM,MSUMA)
          CALL FPMPY(MTOL,MSUMA,MSUMA)
          IF (FPCOMP(MTRMA,'LE',MSUMA)) GOTO 25
          CALL FPMPYI(MTRM,J,MTRM)
          CALL FPADD(MSUM,MTRM,MSUM)
          CALL FPADD(MAK,MONE,MAK)
  20    CONTINUE
        PRINT*,"DROP THRU ASYMPTOTOIC SERIES LOOP"
        PAUSE
  25    CONTINUE
        CALL FPMPYI(MT2,J,MAK)
        CALL FPEXP(MAK,MTRM)
        CALL FPMPY(MSUM,MTRM,MSUM)
C        CALL FPMPY(MRTPI,MT2,MAK)
        CALL FPMPY(MRTPI,MT,MAK)
        CALL FPDIV(MSUM,MAK,MSUMT)
      ENDIF
      CALL FPEQU(MSUMT,MFUN,NDIG,NDIG)
C      CALL FPM2DP(MSUMT,DSUM)
C      CALL FPM2DP(MT,DT)
C      PRINT *,I,DT,DSUM
C      PAUSE
      RETURN
      END
      SUBROUTINE DATAS(N,A,DNAME,FNAME,FWORK,IERR)
C
C     WRITTEN BY D.E.AMOS, MAY, 1990.
C
C     REFERENCE
C
C     ABSTRACT
C        DATAS FORMATS A DOUBLE PRECISION SINGLY DIMENSIONED ARRAY
C        A(I), I=1,N INTO A DATA STATEMENT IN FILE FNAME. THE A
C        VECTOR IS FORMATTED BY 3E21.14 ACROSS A LINE AND GIVEN THE
C        NAME SPECIFIED BY THE PARAMETER DNAME. DATAS ALSO CREATES
C        A TEMPORARY FILE TEMPX.DAT.
C
```

```
C     DESCRIPTION OF ARGUMENTS
C
C        INPUT
C           N      -  NUMBER OF ELEMENTS OF A, 1.LE.N.LE.99
C           A      -  DOUBLE PRECISION ARRAY OF DIMENSION AT LEAST N
C           DNAME  -  CHARACTER*K VARIABLE FOR THE NAME OF THE DATA
C                     IN THE FILE DESIGNATED BY FNAME, 1.LE.K.LE.10
C           FNAME  -  CHARACTER*K VARIABLE FOR THE FILE NAME WHERE THE
C                     DATA STATEMENT IS TO BE WRITTEN, 1.LE.K.LE.12
C           FWORK  -  CHARACTER*K VARIABLE FOR A SCRATCH FILE NAME,
C                     1.LE.K.LE.12
C
C
C        OUTPUT
C           ****   -  FILE WITH ALIAS FNAME WHICH CONTAINS THE DATA
C                     STATEMENT
C           ****      FILE WITH ALIAS FWORK USED AS A SCRATCH FILE
C           IERR   -  ERROR INDICATOR
C                     IERR=0  NORMAL RETURN, NO ERRORS
C                     IERR=1  N IS LESS THAN 1
C                     IERR=2  N IS LARGER THAN 99
C                     IERR=3  DNAME IS BLANK
C                     IERR=4  FNAME IS BLANK
C                     IERR=5  FWORK IS BLANK
C
C     ERROR CONDITIONS
C        N LESS THAN 1 OR GREATER THAN 99 IS AN ERROR
C        EITHER DNAME OR FNAME BLANK IS AN ERROR
C-----------------------------------------------------------------------
      CHARACTER*(*) DNAME,FNAME,FWORK
      CHARACTER*72 LINE
      CHARACTER*2 NC
      CHARACTER*21 B(99)
      DOUBLE PRECISION A(N)
      IERR=0
      IF (N.GT.99) THEN
        IERR=2
        RETURN
      ENDIF
      IF (N.LT.1) THEN
        IERR=1
        RETURN
      ENDIF
      DO 5 I=1,12
        IF(FNAME(I:I).NE.' ') GO TO 6
    5 CONTINUE
      IERR=4
      RETURN
    6 CONTINUE
      DO 7 I=1,12
        IF(FWORK(I:I).NE.' ') GO TO 8
    7 CONTINUE
      IERR=5
      RETURN
    8 CONTINUE
      DO 15 I=1,10
        IF(DNAME(I:I).NE.' ') GO TO 20
   15 CONTINUE
      IERR=3
      RETURN
   20 CONTINUE
      K1=I
      DO 25 I=K1,10
        IF(DNAME(I:I).EQ.' ') GO TO 30
   25 CONTINUE
      I=11
   30 CONTINUE
```

```
      K2=I-1
      OPEN(10,FILE=FNAME,STATUS='UNKNOWN')
      REWIND(10)
C     POSITION FILE AT END OF LAST LINE
      DO 31 LOOP=1,400
        READ(10,'( )',END=32)
 31   CONTINUE
 32   CONTINUE
      ENDFILE(10)
      BACKSPACE(10)
C
      OPEN(20,FILE=FWORK,STATUS='UNKNOWN')
      WRITE(20,800) N
      REWIND(20)
      READ(20,801) NC
      REWIND(20)
      LINE='      DATA ('//DNAME(K1:K2)//'(I), I=1,'//NC//')/'
      WRITE(10,900)LINE
      DO 35 I=1,N
        WRITE(20,901) A(I)
   35 CONTINUE
      REWIND(20)
      DO 40 I=1,N
        READ(20,902) B(I)
   40 CONTINUE
      K=0
      DO 45 I=1,N
        LINE='     &'
        M=6
        DO 50 J=1,3
          K=K+1
          LINE=LINE(1:M)//B(K)(1:21)
          M=M+21
          IF(K.EQ.N) THEN
            LINE=LINE(1:M)//'/'
            GO TO 60
          ELSE
            LINE=LINE(1:M)//','
          ENDIF
          M=M+1
   50   CONTINUE
   60   CONTINUE
        WRITE(10,900)LINE
        IF (K.EQ.N) GO TO 70
   45 CONTINUE
   70 CONTINUE
      CLOSE(10)
C-----------------------------------------------------------------------
C     CLEAN UP THE TEMPORARY FILE
C-----------------------------------------------------------------------
      LINE='  '
      REWIND(20)
      WRITE(20,900) LINE
      CLOSE(20)
  800 FORMAT(I2)
  801 FORMAT(A2)
  900 FORMAT(A72)
  901 FORMAT(D21.14)
  902 FORMAT(A21)
      RETURN
      END
************************************************************************
      PROGRAM MCHEBYEV
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      CHARACTER*72 LINE
      COMMON /CINIT/INIT
      DIMENSION COEF(100)
```

```
      OPEN(7,FILE='MCHEBYEV.TXT',STATUS='UNKNOWN')
C-----------------------------------------------------------------------
C     READ A, B AND 18 DIGIT COEFFICIENTS FROM FILE CHEBY.DAT
C-----------------------------------------------------------------------
      OPEN(10,FILE='MPCHEBY.DAT',STATUS='UNKNOWN')
      REWIND(10)
      READ(10,196)NODES
 196  FORMAT(7X,I3)
      READ(10,199) A
 199  FORMAT(3X,D30.18)
      READ(10,199) B
      DO 4 J=1,3
C       SKIP OVER THE FIRST DATA STATEMENT
        READ(10,200) LINE
 4    CONTINUE
      DO 5 J=1,400
        READ(10,200) LINE
C        PRINT *,LINE
 200    FORMAT(A72)
        IF(LINE(7:10).EQ.'DATA') GOTO 201
 5    CONTINUE
      STOP
 201  CONTINUE
      M=1
      DO 6 J=1,400
        READ(10,202,IOSTAT=K) COEF(M),COEF(M+1)
C        PRINT *,J,K,COEF(M),COEF(M+1)
        IF(K.NE.0) GOTO 7
        M=M+2
 202    FORMAT(8X,D26.20,4X,D26.20)
 6    CONTINUE
 7    CONTINUE
C     K=-1 ONLY IF THE READ PRODUCES TWO ZERO ENTRIES
      CLOSE(10)
C
      IF(COEF(M).EQ.0.0D0.AND.COEF(M+1).EQ.0.0D0) THEN
        M=M-1
        IF(COEF(M).EQ.0.0D0) THEN
          M=M-1
        ENDIF
      ELSE
        IF(COEF(M).EQ.0.0D0) THEN
          M=M-1
        ENDIF
      ENDIF
C
      INIT=0
 400  CONTINUE
C      PRINT 195, NODES
      PRINT 198, NODES,A,B
      PRINT 194
      K=1
      MF=INT(M/2)
      DO 18 MM=1,MF
        PRINT 203, MM,COEF(K),COEF(K+1)
        K=K+2
 18   CONTINUE
      IF(MOD(M,2).NE.0) THEN
        PRINT 203, MM,COEF(M)
      ENDIF
 9    CONTINUE
C      PRINT *, ' '
      PRINT *, M,' COEFFICIENTS READ FROM FILE MPCHEBY.DAT'
C      PRINT *, ' '
      PRINT *,'INPUT THE NUMBER OF COEFFICIENTS TO BE USED IN THE SUM (S
     &TOP=0):  '
      READ *,N
```

```
      IF (N.GT.M) GOTO 9
      IF (N.EQ.0) STOP
      REWIND(7)
C       WRITE(7,195) NODES
C 195   FORMAT(' NODES= ',I3)
      WRITE(7,198) NODES,A,B
 198  FORMAT('  NODES= ',I3,'  A=',D25.18,' B=',D25.18)
      WRITE(7,194)
 194  FORMAT(7X,'CHEBYSHEV COEFFICIENTS')
      K=1
      MF=INT(M/2)
      DO 8 MM=1,MF
        WRITE(7,203) MM,COEF(K),COEF(K+1)
        K=K+2
 8    CONTINUE
      IF(MOD(M,2).NE.0) THEN
        WRITE(7,203) MM,COEF(M)
      ENDIF
 203  FORMAT(2X,I5,2D30.18)
      WRITE(7,193)N
 193  FORMAT(7X,'NUMBER OF COEFFICIENTS USED IN THE SUM: ',I3)
      WRITE(7,197)
 197  FORMAT(/'                X             Y         REL ERR')
      PRINT 197
      APB=A+B
      BMA=B-A
      AJ=1.0D0
      JL=51
      AJL=DBLE(FLOAT(JL))
      DO 10 J=1,JL
        X=((AJ-1.0D0)*B+(AJL-AJ)*A)/(AJL-1.0D0)
C         X2=X*X
        CALL DCHBYS(APB,BMA,X,N,COEF,ANS)
C         X3=X2*X
C         ANS=ANS*DEXP(-X2)/X3
C          Y=MPGOFX(X)
        Y=DPGOFX(X)
C         Y=1.0D0/X
        ERR=DABS(ANS-Y)/DABS(Y)
        WRITE(7,300) J,X,Y,ERR
 300    FORMAT(4X,I5,3D13.5)
        PRINT 300, J,X,Y,ERR
        IF(MOD(J,20).EQ.0) PAUSE
        AJ=AJ+1.0D0
 10   CONTINUE
      PAUSE
      DO 401 J=1,25
        PRINT *
 401  CONTINUE
      GOTO 400
      END
      SUBROUTINE DCHBYS(APB,BMA,X,N,COEF,SUM)
C-----------------------------------------------------------------------
C     CHBYS SUMS A CHEBYSHEV SERIES WITH COEFFICIENTS COEF(I), I=1,N
C     ON [A,B] AT X. APB=A+B , BMA=B-A
C-----------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION COEF(N)
      T=(X+X-APB)/BMA
      TT=T+T
      F1=0.0D0
      F2=0.0D0
      NM=N-1
      DO 160 I=1,NM
        J=N-I+1
        TEMP=F1
        F1=TT*F1-F2+COEF(J)
```

```
        F2=TEMP
  160 CONTINUE
      SUM=T*F1-F2+COEF(1)
      RETURN
      END
      DOUBLE PRECISION FUNCTION GOFX(X)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON /CFOFX/ CX
C      DATA RTPI /1.772453850905516D0/
      EXTERNAL FOFX
      CX=X
      INIT=0
      X1=X
      SIG=2.0D0
      REL=0.50D-15
C      PRINT *,"X=",X
C      PAUSE
      CALL DQUAD8(FOFX,INIT,X1,SIG,REL,X2,QANS,IERR)
      SIG=6.0D0
      CALL DQUAD8(FOFX,INIT,X1,SIG,REL,X2,QANS,IERR)
      XSQ=X*X
      GOFX=QANS*XSQ*X
      GOFX=GOFX*DEXP(XSQ)
C      GOFX=QANS
C     PRINT *,"GOFX=",X,GOFX
C      PAUSE
      RETURN
      END
      DOUBLE PRECISION FUNCTION FOFX(X)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      KODE=2
      FOFX=DRERF(X,KODE,NZ)/X
      RETURN
      END
      DOUBLE PRECISION FUNCTION DPGOFX(X)
      DOUBLE PRECISION X,PI,DH,DTOL
      DOUBLE PRECISION DPMAX
C     SINGLE PRECISION SPMAX

      PARAMETER ( MXNDIG=256 , NBITS=32 ,
     *          LPACK  = (MXNDIG+1)/2 + 1 , LUNPCK = (6*MXNDIG)/5 + 20,
     *          LMWA   = 2*LUNPCK         , LJSUMS = 8*LUNPCK,
     *          LMBUFF = ((LUNPCK+3)*(NBITS-1)*301)/2000 + 6  )
C
C
      DIMENSION MX(200),MGOFX(200)

C
      COMMON /FMUSER/ NDIG,JBASE,JFORM1,JFORM2,KRAD,
     *                KW,NTRACE,LVLTRC,KFLAG,KWARN,KROUND
C
      COMMON /FM/ MWA(LMWA),NCALL,MXEXP,MXEXP2,KARGSW,KEXPUN,KEXPOV,
     *            KUNKNO,IUNKNO,RUNKNO,MXBASE,MXNDG2,KSTACK(19),
     *            MAXINT,SPMAX,DPMAX
C
      COMMON /FMSAVE/ NDIGPI,NJBPI,NDIGE,NJBE,NDIGLB,NJBLB,NDIGLI,NJBLI,
     *                MPISAV(LUNPCK),MESAV(LUNPCK),MLBSAV(LUNPCK),
     *                MLN1(LUNPCK),MLN2(LUNPCK),MLN3(LUNPCK),
     *                MLN4(LUNPCK)
C
      COMMON /CINIT/INIT
      COMMON /CWORK/MPI(200),MRTPI(200),MONE(200),MTWO(200),MTRTPI(200),
     *MZERO(200),MTOL(200),MDH(200),MNINE(200),NCDIG
      COMMON /GAUSS/ MX1(200),MX2(200),MX3(200),MX4(200),MW1(200),
     &MW2(200),MW3(200),MW4(200)
C
C-----------------------------------------------------------------------
```

```
      IF (INIT.EQ.0) THEN
        NPREC=60
        CALL FPSET(NPREC)
C       FOR JBASE=10000, NDIG=NPREC/4+2
        NDIG=INT(NPREC/4)+5
        NCDIG=NDIG
        K=0
        CALL FPI2M(K,MZERO)
        K=1
        CALL FPI2M(K,MONE)
        K=2
        CALL FPI2M(K,MTWO)
        K=9
        CALL FPI2M(K,MNINE)
        CALL FPPI(MPI)
        CALL FPM2DP(MPI,PI)
        CALL FPSQRT(MPI,MRTPI)
        CALL FPDIV(MONE,MRTPI,MTEMP)
        CALL FPADD(MTEMP,MTEMP,MTRTPI)
C       MTRTPI=2/RTPI
        DTOL=0.50D-16
        CALL FPDP2M(DTOL,MTOL)
        DH=0.5D0
        CALL FPDP2M(DH,MDH)
C       INIT=1
C       INIT IS SET TO 1 AFTER THE INITIALIZATION OF MQUAD8
      ENDIF
      CALL FPDP2M(X,MX)
      CALL MPGOFX(MX,MGOFX)
C      CALL MFOFXA(MX,MGOFX)
      CALL FPM2DP(MGOFX,DPGOFX)
      END
      SUBROUTINE MPGOFX(MX,MGOFX)
C
C     COMPUTES INT ON (X,INF) OF ERFC(X)/X OR ERFC(X) IN MP ARITHMETIC
C     USING GAUS8 FORMULA ON REPEATED SUBDIVIDED INTERVALS
C
      DOUBLE PRECISION DSUM
      LOGICAL FPCOMP
C
      DIMENSION MX(200),MGOFX(200)
C
      DIMENSION MSUM(200),MSUMA(200),MANS(200),MANSA(200),
     &MTEMP(200),MSUM1(200),MDX(200),MTEMP1(200),MA(200),MB(200)
C
      COMMON /CWORK/MPI(200),MRTPI(200),MONE(200),MTWO(200),MTRTPI(200),
     *MZERO(200),MTOL(200),MDH(200),MNINE(200),NCDIG
C
      NDIG=NCDIG
      CALL FPEQU(MDH,MDX,NDIG,NDIG)
      CALL FPEQU(MZERO,MSUM1,NDIG,NDIG)
      KL=100
      DO 10 J=1,100
        CALL FPDIVI(MDX,2,MDX)
        KL=KL+KL
        CALL FPEQU(MZERO,MSUM,NDIG,NDIG)
        CALL FPEQU(MX,MB,NDIG,NDIG)
        DO 20 K=1,KL
          CALL FPEQU(MB,MA,NDIG,NDIG)
          CALL FPADD(MA,MDX,MB)
          CALL MQUAD8(MA,MB,MANS)
          CALL FPADD(MSUM,MANS,MSUM)
          CALL FPM2DP(MSUM,DSUM)
          IF(J.EQ.1) THEN
            CALL FPABS(MANS,MANSA)
            CALL FPABS(MSUM,MSUMA)
            CALL FPMPY(MSUMA,MTOL,MSUMA)
```

```
            IF (FPCOMP(MANSA,'LT',MSUMA)) GOTO 25
          ENDIF
 20     CONTINUE
 25     CONTINUE
        IF(J.EQ.1) THEN
          KL=K
          CALL FPEQU(MSUM,MSUM1,NDIG,NDIG)
          GOTO 10
        ENDIF
        CALL FPSUB(MSUM,MSUM1,MTEMP)
        CALL FPABS(MTEMP,MTEMP)
        CALL FPABS(MSUM,MSUMA)
        CALL FPMPY(MSUMA,MTOL,MSUMA)
        IF (FPCOMP(MTEMP,'LT',MSUMA)) GOTO 30
        CALL FPEQU(MSUM,MSUM1,NDIG,NDIG)
 10   CONTINUE
 30   CONTINUE
      CALL FPMPY(MX,MX,MTEMP)
      CALL FPEXP(MTEMP,MTEMP1)
C      CALL FPMPY(MTEMP,MX,MTEMP)
      CALL FPMPY(MTEMP1,MTEMP,MTEMP)
      CALL FPMPY(MTEMP,MSUM,MGOFX)
      RETURN
      END
      SUBROUTINE MQUAD8(MA,MB,MQUAD)
      DOUBLE PRECISION DQUAD
      DIMENSION MA(200),MB(200),MAPB(200),MBMA(200),MTEMP1(200),
     &MTEMP2(200),MTEMP3(200),MTEMP4(200),MTEMP5(200),MTEMP6(200),
     &MTEMP7(200),MTEMP8(200),MSUM(200),MFUN(200),MQUAD(200)
      DIMENSION IX1(4),IX2(4),IX3(4),IX4(4),IW1(4),IW2(4),IW3(4),IW4(4)
C
      COMMON /CINIT/INIT
      COMMON /CWORK/MPI(200),MRTPI(200),MONE(200),MTWO(200),MTRTPI(200),
     *MZERO(200),MTOL(200),MDH(200),MNINE(200),NCDIG
      COMMON /GAUSS/ MX1(200),MX2(200),MX3(200),MX4(200),MW1(200),
     &MW2(200),MW3(200),MW4(200)

C
C     GAUSS ABSCISSAS AND WEIGHTS IN MP
      DATA (IX1(J),J=1,4)/0,18343464,24956498,05000000/
      DATA (IX2(J),J=1,4)/0,52553240,99163289,86000000/
      DATA (IX3(J),J=1,4)/0,79666647,74136267,40000000/
      DATA (IX4(J),J=1,4)/0,96028985,64975362,32000000/
      DATA (IW1(J),J=1,4)/0,36268378,33783619,83000000/
      DATA (IW2(J),J=1,4)/0,31370664,58778872,87000000/
      DATA (IW3(J),J=1,4)/0,22238103,44533744,71000000/
      DATA (IW4(J),J=1,4)/0,10122853,62903762,59000000/
C
      NDIG=NCDIG
      IF (INIT.EQ.0) THEN
        CALL FPEQU(MZERO,MX1,NDIG,NDIG)
        MX1(1)=IX1(1)
        MX1(2)=IX1(2)
        MX1(3)=IX1(3)
        MX1(4)=IX1(4)
        CALL FPEQU(MZERO,MX2,NDIG,NDIG)
        MX2(1)=IX2(1)
        MX2(2)=IX2(2)
        MX2(3)=IX2(3)
        MX2(4)=IX2(4)
        CALL FPEQU(MZERO,MX3,NDIG,NDIG)
        MX3(1)=IX3(1)
        MX3(2)=IX3(2)
        MX3(3)=IX3(3)
        MX3(4)=IX3(4)
        CALL FPEQU(MZERO,MX4,NDIG,NDIG)
        MX4(1)=IX4(1)
```

```
         MX4(2)=IX4(2)
         MX4(3)=IX4(3)
         MX4(4)=IX4(4)
         CALL FPEQU(MZERO,MW1,NDIG,NDIG)
         MW1(1)=IW1(1)
         MW1(2)=IW1(2)
         MW1(3)=IW1(3)
         MW1(4)=IW1(4)
         CALL FPEQU(MZERO,MW2,NDIG,NDIG)
         MW2(1)=IW2(1)
         MW2(2)=IW2(2)
         MW2(3)=IW2(3)
         MW2(4)=IW2(4)
         CALL FPEQU(MZERO,MW3,NDIG,NDIG)
         MW3(1)=IW3(1)
         MW3(2)=IW3(2)
         MW3(3)=IW3(3)
         MW3(4)=IW3(4)
         CALL FPEQU(MZERO,MW4,NDIG,NDIG)
         MW4(1)=IW4(1)
         MW4(2)=IW4(2)
         MW4(3)=IW4(3)
         MW4(4)=IW4(4)
         PRINT *,'GAUSSIAN ABSCISSAS AND WEIGHTS '
         CALL FPPRNT(MX1)
         CALL FPPRNT(MX2)
         CALL FPPRNT(MX3)
         CALL FPPRNT(MX4)
         CALL FPPRNT(MW1)
         CALL FPPRNT(MW2)
         CALL FPPRNT(MW3)
         CALL FPPRNT(MW4)
         PAUSE
         INIT=1
       ENDIF
       CALL FPADD(MA,MB,MAPB)
       CALL FPDIVI(MAPB,2,MAPB)
       CALL FPSUB(MB,MA,MBMA)
       CALL FPDIVI(MBMA,2,MBMA)
       CALL FPMPY(MX1,MBMA,MTEMP1)
       CALL FPMPY(MX2,MBMA,MTEMP2)
       CALL FPMPY(MX3,MBMA,MTEMP3)
       CALL FPMPY(MX4,MBMA,MTEMP4)
       I=-1
       CALL FPMPYI(MTEMP1,I,MTEMP5)
       CALL FPMPYI(MTEMP2,I,MTEMP6)
       CALL FPMPYI(MTEMP3,I,MTEMP7)
       CALL FPMPYI(MTEMP4,I,MTEMP8)
C
       CALL FPADD(MTEMP1,MAPB,MTEMP1)
       CALL FPADD(MTEMP2,MAPB,MTEMP2)
       CALL FPADD(MTEMP3,MAPB,MTEMP3)
       CALL FPADD(MTEMP4,MAPB,MTEMP4)
       CALL FPADD(MTEMP5,MAPB,MTEMP5)
       CALL FPADD(MTEMP6,MAPB,MTEMP6)
       CALL FPADD(MTEMP7,MAPB,MTEMP7)
       CALL FPADD(MTEMP8,MAPB,MTEMP8)
C
       CALL FPEQU(MZERO,MSUM,NDIG,NDIG)
       CALL MFOFX(MTEMP1,MFUN)
       CALL FPMPY(MW1,MFUN,MFUN)
       CALL FPADD(MSUM,MFUN,MSUM)
       CALL MFOFX(MTEMP2,MFUN)
       CALL FPMPY(MW2,MFUN,MFUN)
       CALL FPADD(MSUM,MFUN,MSUM)
       CALL MFOFX(MTEMP3,MFUN)
       CALL FPMPY(MW3,MFUN,MFUN)
```

```
      CALL FPADD(MSUM,MFUN,MSUM)
      CALL MFOFX(MTEMP4,MFUN)
      CALL FPMPY(MW4,MFUN,MFUN)
      CALL FPADD(MSUM,MFUN,MSUM)
      CALL MFOFX(MTEMP5,MFUN)
      CALL FPMPY(MW1,MFUN,MFUN)
      CALL FPADD(MSUM,MFUN,MSUM)
      CALL MFOFX(MTEMP6,MFUN)
      CALL FPMPY(MW2,MFUN,MFUN)
      CALL FPADD(MSUM,MFUN,MSUM)
      CALL MFOFX(MTEMP7,MFUN)
      CALL FPMPY(MW3,MFUN,MFUN)
      CALL FPADD(MSUM,MFUN,MSUM)
      CALL MFOFX(MTEMP8,MFUN)
      CALL FPMPY(MW4,MFUN,MFUN)
      CALL FPADD(MSUM,MFUN,MSUM)
      CALL FPM2DP(MFUN,DQUAD)
C      PRINT*,'MFUN:',DQUAD
      CALL FPMPY(MSUM,MBMA,MQUAD)
C      CALL FPM2DP(MQUAD,DQUAD)
C      PRINT *,'IN MQUAD, ANS=',DQUAD
C      PAUSE
      RETURN
      END
      SUBROUTINE MFOFXA(MT,MFUN)
      DOUBLE PRECISION DSUM,DMT
      LOGICAL FPCOMP
      DIMENSION MT(200),MT2(200),MTRM(200),MSUM(200),MCI(200),MCNT(200),
     &MTRM1(200),MTRMA(200),MTRMS(200),MTEMP(200),MSUMA(200),MAK(200),
     &MFUN(200),MSUMT(200)
      COMMON /CWORK/MPI(200),MRTPI(200),MONE(200),MTWO(200),MTRTPI(200),
     *MZERO(200),MTOL(200),MDH(200),MNINE(200),NCDIG
C
      NDIG=NCDIG
      CALL FPM2DP(MT,DMT)
      CALL FPMPY(MT,MT,MT2)
      CALL FPEQU(MONE,MTRM,NDIG,NDIG)
      CALL FPEQU(MONE,MSUM,NDIG,NDIG)
      J=-1
      IF(FPCOMP(MT,"LT",MNINE)) THEN
        CALL FPEQU(MONE,MCI,NDIG,NDIG)
        CALL FPEQU(MTWO,MCNT,NDIG,NDIG)
        CALL FPADD(MONE,MCNT,MCNT)
C        CALL FPDIV(MONE,MT,MTRM1)
        DO 10 I=1,1000
          CALL FPMPY(MTRM,MT2,MTRM)
          CALL FPDIV(MTRM,MCI,MTRM)
          CALL FPMPYI(MTRM,J,MTRM)
          CALL FPDIV(MTRM,MCNT,MTRMS)
          CALL FPADD(MSUM,MTRMS,MSUM)
          CALL FPM2DP(MSUM,DSUM)
C          PRINT*,'IN MFOFX, X,SUM ',DMT,DSUM
          IF(I.LT.5) GOTO 5
          CALL FPMPY(MSUM,MTRTPI,MSUMT)
C          CALL FPSUB(MTRM1,MTEMP,MSUMT)
          CALL FPABS(MTRMS,MTRMA)
          CALL FPABS(MSUMT,MSUMA)
          CALL FPMPY(MTOL,MSUMA,MSUMA)
          IF (FPCOMP(MTRMA,'LT',MSUMA)) GOTO 15
  5       CONTINUE
          CALL FPADD(MCI,MONE,MCI)
          CALL FPADD(MCNT,MTWO,MCNT)
 10     CONTINUE
        PRINT*,"DROP THRU SERIES LOOP"
        PAUSE
 15     CONTINUE
      ELSE
```

```
        CALL FPEQU(MDH,MAK,NDIG,NDIG)
        DO 20 I=1,25
          CALL FPMPY(MTRM,MAK,MTRM)
          CALL FPDIV(MTRM,MT2,MTRM)
          CALL FPABS(MTRM,MTRMA)
          IF (FPCOMP(MTRMA,'LE',MTOL)) GOTO 25
          CALL FPMPYI(MTRM,J,MTRM)
          CALL FPADD(MSUM,MTRM,MSUM)
          CALL FPADD(MAK,MONE,MAK)
  20    CONTINUE
        PRINT*,"DROP THRU ASYMPTOTOIC SERIES LOOP"
        PAUSE
  25    CONTINUE
        CALL FPMPYI(MT2,J,MAK)
        CALL FPEXP(MAK,MTRM)
        CALL FPMPY(MSUM,MTRM,MSUM)
        CALL FPMPY(MRTPI,MT2,MAK)
        CALL FPDIV(MSUM,MAK,MSUMT)
      ENDIF
      CALL FPEQU(MSUMT,MFUN,NDIG,NDIG)
      RETURN
      END
      SUBROUTINE MFOFX(MT,MFUN)
      DOUBLE PRECISION DSUM,DMT
      LOGICAL FPCOMP
      DIMENSION MT(200),MT2(200),MTRM(200),MSUM(200),MCI(200),MCNT(200),
     &MTRM1(200),MTRMA(200),MTRMS(200),MTEMP(200),MSUMA(200),MAK(200),
     &MFUN(200),MSUMT(200)
      COMMON /CWORK/MPI(200),MRTPI(200),MONE(200),MTWO(200),MTRTPI(200),
     *MZERO(200),MTOL(200),MDH(200),MNINE(200),NCDIG
C
      NDIG=NCDIG
      CALL FPM2DP(MT,DMT)
      CALL FPMPY(MT,MT,MT2)
      CALL FPEQU(MONE,MTRM,NDIG,NDIG)
      CALL FPEQU(MONE,MSUM,NDIG,NDIG)
      J=-1
      IF(FPCOMP(MT,"LT",MNINE)) THEN
        CALL FPEQU(MONE,MCI,NDIG,NDIG)
        CALL FPEQU(MTWO,MCNT,NDIG,NDIG)
        CALL FPADD(MONE,MCNT,MCNT)
        CALL FPDIV(MONE,MT,MTRM1)
        DO 10 I=1,1000
          CALL FPMPY(MTRM,MT2,MTRM)
          CALL FPDIV(MTRM,MCI,MTRM)
          CALL FPMPYI(MTRM,J,MTRM)
          CALL FPDIV(MTRM,MCNT,MTRMS)
          CALL FPADD(MSUM,MTRMS,MSUM)
          CALL FPM2DP(MSUM,DSUM)
C          PRINT*,'IN MFOFX, X,SUM ',DMT,DSUM
          IF(I.LT.5) GOTO 5
          CALL FPMPY(MSUM,MTRTPI,MTEMP)
C          CALL FPSUB(MTRM1,MTEMP,MSUMT)
          CALL FPMPY(MTEMP,MT,MTEMP)
          CALL FPSUB(MONE,MTEMP,MSUMT)
          CALL FPABS(MTRMS,MTRMA)
          CALL FPABS(MSUMT,MSUMA)
          CALL FPMPY(MTOL,MSUMA,MSUMA)
          IF (FPCOMP(MTRMA,'LT',MSUMA)) GOTO 15
  5       CONTINUE
          CALL FPADD(MCI,MONE,MCI)
          CALL FPADD(MCNT,MTWO,MCNT)
  10    CONTINUE
        PRINT*,"DROP THRU SERIES LOOP"
        PAUSE
  15    CONTINUE
      ELSE
```

```
        CALL FPEQU(MDH,MAK,NDIG,NDIG)
        DO 20 I=1,25
          CALL FPMPY(MTRM,MAK,MTRM)
          CALL FPDIV(MTRM,MT2,MTRM)
          CALL FPABS(MTRM,MTRMA)
          IF (FPCOMP(MTRMA,'LE',MTOL)) GOTO 25
          CALL FPMPYI(MTRM,J,MTRM)
          CALL FPADD(MSUM,MTRM,MSUM)
          CALL FPADD(MAK,MONE,MAK)
   20   CONTINUE
        PRINT*,"DROP THRU ASYMPTOTOIC SERIES LOOP"
        PAUSE
   25   CONTINUE
        CALL FPMPYI(MT2,J,MAK)
        CALL FPEXP(MAK,MTRM)
        CALL FPMPY(MSUM,MTRM,MSUM)
C         CALL FPMPY(MRTPI,MT2,MAK)
        CALL FPMPY(MRTPI,MT,MAK)
        CALL FPDIV(MSUM,MAK,MSUMT)
      ENDIF
      CALL FPEQU(MSUMT,MFUN,NDIG,NDIG)
      RETURN
      END
************************************************************************
      PROGRAM DPCHEBY
C
C     WRITTEN BY D.E.AMOS, DECEMBER,1972; MODIFIED MAY, 1990.
C
C     REFERENCE SANDIA NATIONAL LABORATORIES REPORT SC-DR-72 0917
C
C     ABSTRACT
C
C     PROGRAM DPCHEBY CALCULATES THE CHEBYSHEV COEFFICIENTS FOR A
C     DP FUNCTION GOFX(X) ON AN INTERVAL [A,B].
C
C     FINAL EXPANSION IS  S = 0.5*A(1)+SUM( A(J)*CT(J,XX), J=2,N) WHERE
C
C          XX=(2*X-(A+B))/(B-A)   FOR    A.LE.X.LE.B
C                                 OR
C          XX=2*A/X-1.0           FOR    A.LE.X.LT.INFINITY,
C
C     T(J,XX) ARE CHEBYSHEV POLYNOMIALS AND A(J) ARE THE COEFFICIENTS
C     GENERATED BY THIS PROGRAM AND STORED IN A DATA STATEMENT IN A
C     FILE CALLED DPCHEBY.DAT.
C
C     DESCRIPTION OF ARGUMENTS
C
C     INPUT
C       A, B    = INTERVAL [A,B] END POINTS. IF A=B, THE EXPANSION
C                 IS TAKEN ON A TO INFINITY USING T(J,2*A/X-1)
C       N       = THE NUMBER OF COEFFICIENTS DESIRED (USUALLY N < 26)
C       NCHEBY  = NUMBER OF CHEBYSHEV NODES FOR THE APPROXIMATION
C                 NCHEBY IS MUCH LARGER THAN N  (USUALLY NCHEBY > 40)
C       IERR    = 1 GIVES A RELATIVE ERROR PRINT
C               = 0 GIVES AN ABSOLUTE ERROR PRINT
C       DNAME   = NAME OF THE COEFFICIENT ARRAY IN THE DATA STATEMENT
C       GOFX(X) = DOUBLE PRECISION FUNCTION BEING APPROXIMATED. X MUST
C                 BE DECLARED DOUBLE AND GOFX PROGRAMMED BY THE USER
C                 AS DOUBLE PRECISION FUNCTION GOFX(X)
C     OUTPUT
C       FILE DPCHEBY.FIT SHOWING THE COEFFICIENTS AND ERRORS
C       FILE DPCHEBY.DAT CONTAINING THE DATA STATEMENT FOR THE
C                        COEFFICIENTS 0.5*A(1) THRU A(N)
C       FILE CHEBZ.DAT IS CREATED AND USED AS A SCRATCH FILE
C
C     PROGRAM DPCHEBY USES GOFX AND DATAS PLUS ANY OTHERS RELATED TO THE
C     COMPUTATION OF GOFX
```

```
C-----------------------------------------------------------------------
      CHARACTER*12 FNAME,DNAME,FWORK
      CHARACTER*10 ILAB(2)
C      CHARACTER*1 CA,CB
      CHARACTER*72 LINE
      CHARACTER*2 CAB
      DOUBLE PRECISION T,PI,DN,THET,VK,VONE,TK,TONE,APB,BMA,X
      DOUBLE PRECISION Z,TZ,B1,B2,F,TEMP,ANS,AA,BB,AR
      DOUBLE PRECISION GOFX
      DATA (ILAB(I),I=1,2)/' RELATIVE ',' ABSOLUTE '/
      DIMENSION T(301),F(301),X(301),AR(301)
C
      COMMON /CINIT/INIT
C-----------------------------------------------------------------------
      PRINT 950
  950 FORMAT(' APPROXIMATION INTERVAL A, B= ')
      READ *,AA,BB
      PRINT 951
  951 FORMAT(' NUMBER OF CHEBYSHEV COEFFICIENTS DESIRED (N~25), N= ')
      READ *,NAR
      PRINT 952
  952 FORMAT(' NUMBER OF CHEBYSHEV NODES (NCHEBY~40), NCHEBY= ')
      READ *,NCHEBY
      PRINT 953
  953 FORMAT(' NAME OF THE COEFFICIENTS IN THE DATA STATEMENT, DNAME= ')
      READ 900,DNAME
  900 FORMAT(A8)
      PRINT 954
  954 FORMAT(' 1 FOR RELATIVE ERROR; 2 FOR ABSOLUTE ERROR, IERR= ')
      READ *,IERR
      IF (IERR.NE.1 .AND. IERR.NE.2) THEN
        IERR=2
      ENDIF
C-----------------------------------------------------------------------
C     END INPUT
C-----------------------------------------------------------------------
      INIT=0
      N1 = NCHEBY-1
      N2 = NCHEBY-1
      PI=3.14159265358979323846264 33D0
      APB=AA+BB
      BMA=BB-AA
      C1= SNGL(APB)
      C2= SNGL(BMA)
      IF(BB.EQ.AA) C1=SNGL(AA+AA)
      OPEN(10,FILE='DPCHEBY.FIT',STATUS='UNKNOWN')
      DO 500 N=N1,N2
        IF (NAR.GT.N) THEN
          NAR=N+1
        ENDIF
        NP1=N+1
        DN=DBLE(FLOAT(N))
        THET=PI/DN
        T(1)=1.0D0
        T(2)=DCOS(THET)
        TONE =T(2)
        VONE=DSIN(THET)
        VK = VONE
C-----------------------------------------------------------------------
C     GET CHEBYSHEV NODES ON [A,B] IN X ARRAY
C     TRIG IDENTITIES SIN((K+1)*X) = SIN(K*X)*COS(X)+COS(K*X)*SIN(X)
C                     COS((K+1)*X) = COS(K*X)*COS(X)-SIN(K*X)*SIN(X)
C     FOR CHEBYSHEV NODES T(K)=COS((K-1)*PI/N), K=1,N+1 BY RECURRENCE
C-----------------------------------------------------------------------
        DO 5 K=2,N
          TK=T(K)
          T(K+1)=TONE*TK-VK*VONE
```

```
          VK=VK*TONE+TK*VONE
          X(K)=(TK*BMA+APB)/2.0D0
          IF(AA.NE.BB) GO TO 5
          IF(TK.NE.-1.0D0) GO TO 10
          X(K)=1.0D+16
          GO TO 5
   10     CONTINUE
          X(K)=(AA+AA)/(1.0D0+TK)
    5   CONTINUE
        X(1)=(T(1)*BMA+APB)/2.0D0
        X(NP1)=(T(NP1)*BMA+APB)/2.0D0
        IF(AA.NE.BB) GO TO 20
        X(1)=(AA+AA)/(1.0D0+T(1))
        X(NP1)=1.0D+16
   20   CONTINUE
C-----------------------------------------------------------------------
        WRITE(10, 52)NP1,AA,BB
   52 FORMAT(' NODES='I5/' A    ='D23.14/' B    ='D23.14/)
        WRITE(10, 51)(X(I),I=1,NP1)
   51 FORMAT(' CHEBYSHEV NODES  X(J)'/(1X,3D23.14))
C-----------------------------------------------------------------------
C     FUNCTION EVALUATION AT CHEBYSHEV NODES X(I), I=1,NP1
C-----------------------------------------------------------------------
        DO 7 I=1,NP1
          F(I) = GOFX(X(I))
    7   CONTINUE
        F(NP1)=F(NP1)/2.0D0
C-----------------------------------------------------------------------
C     END OF FUNCTION EVALUATION
C-----------------------------------------------------------------------
        DO 100 IR=1,NP1
          Z=T(IR)
          TZ=Z+Z
          B2=0.0D0
          B1= 0.0D0
          DO 15 J=1,N
            K=NP1-J+1
            TEMP=B1
            B1=TZ*B1-B2+F(K)
            B2=TEMP
   15     CONTINUE
          ANS=Z*B1-B2+F(1)/2.0D0
          AR(IR)=(ANS+ANS)/DN
  100   CONTINUE
C-----------------------------------------------------------------------
        WRITE(10,153) (AR(K),K=1,NP1)
  153 FORMAT(' CHEBYSHEV COEFFICIENTS A(J)'/(1X,3D23.14))
        F(NP1)=F(NP1)+F(NP1)
        WRITE(10, 54)(F(I),I=1,NP1)
   54 FORMAT(' FUNCTION VALUES AT CHEBYSHEV NODES F(X(J))'/(1X,3D23.14))
C-----------------------------------------------------------------------
        NARM=NAR-1
        DO 30 I=1,NP1
          XX=SNGL(X(I))
          IF(AA.NE.BB) GO TO 26
          SZ=C1/XX-1.0E0
          GO TO 27
   26     CONTINUE
          SZ=(XX+XX-C1)/C2
   27     CONTINUE
          STZ=SZ+SZ
          S2=0.0E0
          S1=0.0E0
          DO 25 J=1,NARM
            K=NAR-J+1
            ST=S1
            SA=AR(K)
```

```
            S1=STZ*S1-S2+SA
            S2=ST
   25     CONTINUE
          SA=AR(1)
          FX=SZ*S1-S2+SA/2.0
          T(I)=DBLE(FX)-F(I)
          IF(IERR.EQ.1) THEN
            IF(F(I).NE.0.0D0) THEN
              T(I)=T(I)/F(I)
            ENDIF
          ENDIF
   30   CONTINUE
C------------------------------------------------------------------------
        WRITE(10, 53)ILAB(IERR),NAR,AA,BB,(T(I),I=1,NP1)
   53   FORMAT(/' SINGLE PRECISION',A10,'ERRORS AT CHEBYSHEV NODES USING
     &',I5,' TERMS'/' OF THE CHEBYSHEV SUM ON',2D23.14//(1X,3D23.14))
C------------------------------------------------------------------------
        DO 35 I=1,NP1
          IF(AA.NE.BB) GO TO 36
          Z=(AA+AA)/X(I)-1.0D0
          GO TO 37
   36     CONTINUE
          Z=(X(I)+X(I)-APB)/BMA
   37     CONTINUE
          TZ=Z+Z
          B2=0.0D0
          B1=0.0D0
          DO 40 J=1,NARM
            K=NAR-J+1
            TEMP=B1
            B1=TZ*B1-B2+AR(K)
            B2=TEMP
   40     CONTINUE
          TEMP=Z*B1-B2+AR(1)/2.0D0
          T(I)=TEMP-F(I)
          IF(IERR.EQ.1) THEN
            IF(F(I).NE.0.0D0) THEN
              T(I)=T(I)/F(I)
            ENDIF
          ENDIF
   35   CONTINUE
C------------------------------------------------------------------------
        WRITE(10, 55)ILAB(IERR),NAR,AA,BB,(T(I),I=1,NP1)
   55   FORMAT(/' DOUBLE PRECISION',A10,'ERRORS AT CHEBYSHEV NODES USING
     &',I5,' TERMS'/' OF THE CHEBYSHEV SUM ON',2D23.14//(1X,3D23.14))
C------------------------------------------------------------------------
  500 CONTINUE
      CLOSE(10)
      FNAME='DPCHEBY.DAT'
      FWORK='DPCHEBZ.DAT'
      AR(1)=AR(1)*0.5D0
      OPEN(10,FILE='DPCHEBY.DAT',STATUS='UNKNOWN')
      REWIND(10)
      WRITE(10,531)NP1
 531  FORMAT('NODES= ',I3)
      WRITE(10,530) AA,BB
 530  FORMAT('A= 'D30.18/'B= ',D30.18/)
      CLOSE(10)
C------------------------------------------------------------------------
C     WRITE NORMAL DATA STATEMENT WITH 14 DIGITS, 3 ON A LINE
C------------------------------------------------------------------------
      CALL DATAS(NAR,AR,DNAME,FNAME,FWORK,KERR)
C------------------------------------------------------------------------
C     WRITE DATA STATEMENT IN 2 COLUMNS OF 18 DIGITS
C------------------------------------------------------------------------
      DO 515 I=1,10
        IF(DNAME(I:I).NE.' ') GO TO 520
```

```
  515 CONTINUE
      PRINT *, 'THE DATA NAME IS BLANK'
      STOP
  520 CONTINUE
      K1=I
      DO 501 I=K1,10
        IF(DNAME(I:I).EQ.' ') GO TO 502
 501  CONTINUE
      I=11
 502  CONTINUE
      K2=I-1
C     READ TO THE BOTTOM OF THE FILE (TO SKIP WHAT IS ALREADY THERE)
      OPEN(15,FILE=FNAME,STATUS='UNKNOWN')
      REWIND(15)
      DO 511 LOOP=1,400
        READ(15,'( )',END=512)
 511  CONTINUE
 512  CONTINUE
      ENDFILE(15)
      BACKSPACE(15)
      OPEN(20,FILE=FWORK,STATUS='UNKNOWN')
      WRITE(20,509) NAR
 509  FORMAT(I2)
      REWIND(20)
      READ(20,510) CAB
 510  FORMAT(A2)
      CLOSE(20)
      LINE='      DATA ('//DNAME(K1:K2)//'(I), I=1,'//CAB//')/'
      WRITE(15,506)LINE
 506  FORMAT(A72)
      WRITE(15,507) (AR(J),J=1,NAR)
 507  FORMAT(5X,'&',D30.18,',',D30.18,',')
      CLOSE(15)
C-----------------------------------------------------------------------
      IF (KERR.NE.0) THEN
        PRINT *,' ERROR IN DATAS, IERR = ',KERR
      ELSE
        PRINT *,'  '
        PRINT *,' APPROXIMATION USING ',NP1,' CHEBYSHEV NODES'
        PRINT *,'  '
        PRINT *,' FILES WRITTEN:'
        PRINT *,'  '
        PRINT *,' DPCHEBY.FIT:    THE COEFFIENTS, NODES, AND ERRORS,'
        PRINT *,' DPCHEBY.DAT:    THE DATA STATEMENT,'
        PRINT *,' DPCHEBZ.DAT:    SCRATCH FILE.'
      ENDIF
      PAUSE
      STOP
      END
      SUBROUTINE DATAS(N,A,DNAME,FNAME,FWORK,IERR)
C
C     WRITTEN BY D.E.AMOS, MAY, 1990.
C
C     REFERENCE
C
C     ABSTRACT
C        DATAS FORMATS A DOUBLE PRECISION SINGLY DIMENSIONED ARRAY
C        A(I), I=1,N INTO A DATA STATEMENT IN FILE FNAME. THE A
C        VECTOR IS FORMATTED BY 3E21.14 ACROSS A LINE AND GIVEN THE
C        NAME SPECIFIED BY THE PARAMETER DNAME. DATAS ALSO CREATES
C        A TEMPORARY FILE TEMPX.DAT.
C
C     DESCRIPTION OF ARGUMENTS
C
C        INPUT
C           N      -  NUMBER OF ELEMENTS OF A, 1.LE.N.LE.99
C           A      -  DOUBLE PRECISION ARRAY OF DIMENSION AT LEAST N
```

```
C           DNAME  -  CHARACTER*K VARIABLE FOR THE NAME OF THE DATA
C                     IN THE FILE DESIGNATED BY FNAME, 1.LE.K.LE.10
C           FNAME  -  CHARACTER*K VARIABLE FOR THE FILE NAME WHERE THE
C                     DATA STATEMENT IS TO BE WRITTEN, 1.LE.K.LE.12
C           FWORK  -  CHARACTER*K VARIABLE FOR A SCRATCH FILE NAME,
C                     1.LE.K.LE.12
C
C
C        OUTPUT
C           ****   -  FILE WITH ALIAS FNAME WHICH CONTAINS THE DATA
C                     STATEMENT
C           ****      FILE WITH ALIAS FWORK USED AS A SCRATCH FILE
C           IERR   -  ERROR INDICATOR
C                     IERR=0  NORMAL RETURN, NO ERRORS
C                     IERR=1  N IS LESS THAN 1
C                     IERR=2  N IS LARGER THAN 99
C                     IERR=3  DNAME IS BLANK
C                     IERR=4  FNAME IS BLANK
C                     IERR=5  FWORK IS BLANK
C
C     ERROR CONDITIONS
C        N LESS THAN 1 OR GREATER THAN 99 IS AN ERROR
C        EITHER DNAME OR FNAME BLANK IS AN ERROR
C-----------------------------------------------------------------------
      CHARACTER*(*) DNAME,FNAME,FWORK
      CHARACTER*72 LINE
      CHARACTER*2 NC
      CHARACTER*21 B(99)
      DOUBLE PRECISION A(N)
      IERR=0
      IF (N.GT.99) THEN
        IERR=2
        RETURN
      ENDIF
      IF (N.LT.1) THEN
        IERR=1
        RETURN
      ENDIF
      DO 5 I=1,12
        IF(FNAME(I:I).NE.' ') GO TO 6
    5 CONTINUE
      IERR=4
      RETURN
    6 CONTINUE
      DO 7 I=1,12
        IF(FWORK(I:I).NE.' ') GO TO 8
    7 CONTINUE
      IERR=5
      RETURN
    8 CONTINUE
      DO 15 I=1,10
        IF(DNAME(I:I).NE.' ') GO TO 20
   15 CONTINUE
      IERR=3
      RETURN
   20 CONTINUE
      K1=I
      DO 25 I=K1,10
        IF(DNAME(I:I).EQ.' ') GO TO 30
   25 CONTINUE
      I=11
   30 CONTINUE
      K2=I-1
      OPEN(10,FILE=FNAME,STATUS='UNKNOWN')
      REWIND(10)
C     POSITION FILE AT END OF LAST LINE
      DO 31 LOOP=1,400
```

```
        READ(10,'( )',END=32)
 31   CONTINUE
 32   CONTINUE
      ENDFILE(10)
      BACKSPACE(10)
C
      OPEN(20,FILE=FWORK,STATUS='UNKNOWN')
      WRITE(20,800) N
      REWIND(20)
      READ(20,801) NC
      REWIND(20)
      LINE='      DATA ('//DNAME(K1:K2)//'(I), I=1,'//NC//')/'
      WRITE(10,900)LINE
      DO 35 I=1,N
        WRITE(20,901) A(I)
   35 CONTINUE
      REWIND(20)
      DO 40 I=1,N
        READ(20,902) B(I)
   40 CONTINUE
      K=0
      DO 45 I=1,N
        LINE='     &'
        M=6
        DO 50 J=1,3
          K=K+1
          LINE=LINE(1:M)//B(K)(1:21)
          M=M+21
          IF(K.EQ.N) THEN
            LINE=LINE(1:M)//'/'
            GO TO 60
          ELSE
            LINE=LINE(1:M)//','
          ENDIF
          M=M+1
   50   CONTINUE
   60   CONTINUE
        WRITE(10,900)LINE
        IF (K.EQ.N) GO TO 70
   45 CONTINUE
   70 CONTINUE
      CLOSE(10)
C-----------------------------------------------------------------------
C     CLEAN UP THE TEMPORARY FILE
C-----------------------------------------------------------------------
      LINE='  '
      REWIND(20)
      WRITE(20,900) LINE
      CLOSE(20)
  800 FORMAT(I2)
  801 FORMAT(A2)
  900 FORMAT(A72)
  901 FORMAT(D21.14)
  902 FORMAT(A21)
      RETURN
      END
      DOUBLE PRECISION FUNCTION GOFX(X)
      DOUBLE PRECISION X
      GOFX=1.0D0/X
      END
************************************************************************
      PROGRAM DCHEBYEV
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      CHARACTER*72 LINE
      DIMENSION COEF(100)
      OPEN(7,FILE='DCHEBYEV.TXT',STATUS='UNKNOWN')
C-----------------------------------------------------------------------
```

```
C      READ A, B AND 18 DIGIT COEFFICIENTS FROM FILE CHEBY.DAT
C----------------------------------------------------------------------
      OPEN(10,FILE='DPCHEBY.DAT',STATUS='UNKNOWN')
      REWIND(10)
      READ(10,196)NODES
 196  FORMAT(7X,I3)
      READ(10,199) A
 199  FORMAT(3X,D30.18)
      READ(10,199) B
      DO 4 J=1,3
C       SKIP OVER THE FIRST DATA STATEMENT
        READ(10,200) LINE
 4    CONTINUE
      DO 5 J=1,400
        READ(10,200) LINE
        PRINT *,LINE
 200    FORMAT(A72)
        IF(LINE(7:10).EQ.'DATA') GOTO 201
 5    CONTINUE
      STOP
 201  CONTINUE
      M=1
      DO 6 J=1,400
        READ(10,202,IOSTAT=K) COEF(M),COEF(M+1)
C        PRINT *,J,K,COEF(M),COEF(M+1)
        IF(K.NE.0) GOTO 7
        M=M+2
 202    FORMAT(6X,D30.18,1X,D30.18)
 6    CONTINUE
 7    CONTINUE
C      K=-1 ONLY IF THE READ PRODUCES TWO ZERO ENTRIES
      CLOSE(10)
C
      IF(COEF(M).EQ.0.0D0.AND.COEF(M+1).EQ.0.0D0) THEN
        M=M-1
        IF(COEF(M).EQ.0.0D0) THEN
          M=M-1
        ENDIF
      ELSE
        IF(COEF(M).EQ.0.0D0) THEN
          M=M-1
        ENDIF
      ENDIF
C
 400  CONTINUE
C       PRINT 195, NODES
      PRINT 198, NODES,A,B
      PRINT 194
      K=1
      MF=INT(M/2)
      DO 18 MM=1,MF
        PRINT 203, MM,COEF(K),COEF(K+1)
        K=K+2
 18   CONTINUE
      IF(MOD(M,2).NE.0) THEN
        PRINT 203, MM,COEF(M)
      ENDIF
 9    CONTINUE
C       PRINT *, ' '
      PRINT *, M,' COEFFICIENTS READ FROM FILE DPCHEBY.DAT'
C       PRINT *, ' '
      PRINT *,'INPUT THE NUMBER OF COEFFICIENTS TO BE USED IN THE SUM (S
     &TOP=0):  '
      READ *,N
      IF (N.GT.M) GOTO 9
      IF (N.EQ.0) STOP
      REWIND(7)
```

```
C      WRITE(7,195) NODES
C 195  FORMAT(' NODES= ',I3)
      WRITE(7,198) NODES,A,B
 198  FORMAT('  NODES= ',I3,'  A=',D25.18,' B=',D25.18)
      WRITE(7,194)
 194  FORMAT(7X,'CHEBYSHEV COEFFICIENTS')
      K=1
      MF=INT(M/2)
      DO 8 MM=1,MF
        WRITE(7,203) MM,COEF(K),COEF(K+1)
        K=K+2
 8    CONTINUE
      IF(MOD(M,2).NE.0) THEN
        WRITE(7,203) MM,COEF(M)
      ENDIF
 203  FORMAT(2X,I5,2D30.18)
      WRITE(7,193)N
 193  FORMAT(7X,'NUMBER OF COEFFICIENTS USED IN THE SUM: ',I3)
      WRITE(7,197)
 197  FORMAT(/'               X            Y         REL ERR')
      PRINT 197
      APB=A+B
      BMA=B-A
      AJ=1.0D0
      JL=51
      AJL=DBLE(FLOAT(JL))
      DO 10 J=1,JL
        X=((AJ-1.0D0)*B+(AJL-AJ)*A)/(AJL-1.0D0)
        X2=X*X
        CALL DCHBYS(APB,BMA,X,N,COEF,ANS)
C        X3=X2*X
C        ANS=ANS*DEXP(-X2)/X3
C        Y=GOFX(X)
        Y=1.0D0/X
        ERR=DABS(ANS-Y)/DABS(Y)
        WRITE(7,300) J,X,Y,ERR
 300    FORMAT(4X,I5,3D13.5)
        PRINT 300, J,X,Y,ERR
        IF(MOD(J,20).EQ.0) PAUSE
        AJ=AJ+1.0D0
 10   CONTINUE
      PAUSE
      DO 401 J=1,25
        PRINT *
 401  CONTINUE
      GOTO 400
      END
      SUBROUTINE DCHBYS(APB,BMA,X,N,COEF,SUM)
C-----------------------------------------------------------------------
C     CHBYS SUMS A CHEBYSHEV SERIES WITH COEFFICIENTS COEF(I), I=1,N
C     ON [A,B] AT X. APB=A+B , BMA=B-A
C-----------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION COEF(N)
      T=(X+X-APB)/BMA
      TT=T+T
      F1=0.0D0
      F2=0.0D0
      NM=N-1
      DO 160 I=1,NM
        J=N-I+1
        TEMP=F1
        F1=TT*F1-F2+COEF(J)
        F2=TEMP
  160 CONTINUE
      SUM=T*F1-F2+COEF(1)
      RETURN
```

```
      END
************************************************************************
      PROGRAM MPCHEBY
C
C     WRITTEN BY D.E.AMOS, DECEMBER,1972; MODIFIED MAY, 1990.
C
C     REFERENCE SANDIA NATIONAL LABORATORIES REPORT SC-DR-72 0917
C
C     ABSTRACT
C
C     PROGRAM MPCHEBY CALCULATES THE CHEBYSHEV COEFFICIENTS FOR A
C     MULTIPLE PRECISION FUNCTION MPGOFX(X) ON AN INTERVAL [A,B].
C
C     FINAL EXPANSION IS  S = 0.5*A(1)+SUM( A(J)*T(J,XX), J=2,N) WHERE
C
C          XX=(2*X-(A+B))/(B-A)   FOR    A.LE.X.LE.B
C                                 OR
C          XX=2*A/X-1.0           FOR    A.LE.X.LT.INFINITY,
C
C     T(J,XX) ARE CHEBYSHEV POLYNOMIALS AND A(J) ARE THE COEFFICIENTS
C     GENERATED BY THIS PROGRAM AND STORED IN A DATA STATEMENT IN A
C     FILE CALLED MPCHEBY.DAT.
C
C     DESCRIPTION OF ARGUMENTS
C
C     INPUT
C       A, B    = INTERVAL [A,B] END POINTS. IF A=B, THE EXPANSION
C                 IS TAKEN ON A TO INFINITY USING T(J,2*A/X-1)
C       N       = THE NUMBER OF COEFFICIENTS DESIRED (USUALLY N < 26)
C       NCHEBY  = NUMBER OF CHEBYSHEV NODES FOR THE APPROXIMATION
C                 NCHEBY IS MUCH LARGER THAN N  (USUALLY NCHEBY > 40)
C       IERR    = 1 GIVES A RELATIVE ERROR PRINT
C               = 0 GIVES AN ABSOLUTE ERROR PRINT
C       DNAME   = NAME OF THE COEFFICIENT ARRAY IN THE DATA STATEMENT
C       MPGOFX  = MULTIPLE PRECISION SUBROUTINE FOR THE FUNCTION BEING
C                 APPROXIMATED. SUBROUTINE MPGOFX(MX,MGOFX)
C     OUTPUT
C       FILE MPCHEBY.FIT SHOWING THE COEFFICIENTS AND ERRORS
C       FILE MPCHEBY.DAT CONTAINING THE DATA STATEMENT FOR THE
C                      COEFFICIENTS 0.5*A(1) THRU A(N)
C       FILE MPCHEBY.DAT IS CREATED AND USED AS A SCRATCH FILE
C
C     PROGRAM MPCHEBY USES MPGOFX AND DATAS PLUS ANY OTHERS RELATED TO
C     THE COMPUTATION OF GOFX
C-----------------------------------------------------------------------
      CHARACTER*2 NK,NAB
      CHARACTER*10 ILAB(2)
      CHARACTER*12 FNAME,DNAME,FWORK
C      CHARACTER*25 LINEA
      CHARACTER*80 LINE,LINEA
      CHARACTER*160 LTEMP,LBLANK
      DOUBLE PRECISION T,PI,DN,THETA,VK,VONE,TK,TONE,APB,BMA,X
      DOUBLE PRECISION Z,TZ,B1,B2,F,AA,BB,AR,TEMP,TEMP1
C
C-----------------------------------------------------------------------
      DOUBLE PRECISION DTOL,DH
      DOUBLE PRECISION DPMAX
C     SINGLE PRECISION SPMAX

      PARAMETER ( MXNDIG=256 , NBITS=32 ,
     *          LPACK  = (MXNDIG+1)/2 + 1 , LUNPCK = (6*MXNDIG)/5 + 20,
     *          LMWA   = 2*LUNPCK         , LJSUMS = 8*LUNPCK,
     *          LMBUFF = ((LUNPCK+3)*(NBITS-1)*301)/2000 + 6  )
C
C
      DIMENSION MTEMP1(200),MTEMP2(200),MAPB(200),MBMA(200),MAA(200),
     &MBB(200),MVK(200),MVONE(200),MTK(200),MTHETA(200),MZ(200),
```

```
     &MTZ(200),MB1(200),MB2(200)
      DIMENSION T(201),F(201),X(201),AR(201)
      DIMENSION MT(200,100),MF(200,100),MX(200,100),MAR(200,100),
     &MTEMP(200)

C
      COMMON /FMUSER/ NDIG,JBASE,JFORM1,JFORM2,KRAD,
     *                KW,NTRACE,LVLTRC,KFLAG,KWARN,KROUND
C
      COMMON /FM/ MWA(LMWA),NCALL,MXEXP,MXEXP2,KARGSW,KEXPUN,KEXPOV,
     *            KUNKNO,IUNKNO,RUNKNO,MXBASE,MXNDG2,KSTACK(19),
     *            MAXINT,SPMAX,DPMAX
C
      COMMON /FMSAVE/ NDIGPI,NJBPI,NDIGE,NJBE,NDIGLB,NJBLB,NDIGLI,NJBLI,
     *                MPISAV(LUNPCK),MESAV(LUNPCK),MLBSAV(LUNPCK),
     *                MLN1(LUNPCK),MLN2(LUNPCK),MLN3(LUNPCK),
     *                MLN4(LUNPCK)
C
      COMMON /CINIT/INIT
      COMMON /CWORK/MPI(200),MRTPI(200),MONE(200),MTWO(200),MTRTPI(200),
     *MZERO(200),MTOL(200),MDH(200),MNINE(200),NCDIG
      COMMON /GAUSS/ MX1(200),MX2(200),MX3(200),MX4(200),MW1(200),
     &MW2(200),MW3(200),MW4(200)
C
      DATA (ILAB(I),I=1,2)/' RELATIVE ',' ABSOLUTE '/

C------------------------------------------------------------------------
      PRINT 950
  950 FORMAT(' APPROXIMATION INTERVAL A, B= ')
      READ *,AA,BB
      PRINT 951
  951 FORMAT(' NUMBER OF CHEBYSHEV COEFFICIENTS DESIRED (N~25), N= ')
      READ *,NAR
      PRINT 952
  952 FORMAT(' NUMBER OF CHEBYSHEV NODES (NCHEBY~40), NCHEBY= ')
      READ *,NCHEBY
      PRINT 953
  953 FORMAT(' NAME OF THE COEFFICIENTS IN THE DATA STATEMENT, DNAME= ')
      READ 900,DNAME
  900 FORMAT(A8)
      PRINT 954
  954 FORMAT(' 1 FOR RELATIVE ERROR; 2 FOR ABSOLUTE ERROR, IERR= ')
      READ *,IERR
      IF (IERR.NE.1 .AND. IERR.NE.2) THEN
        IERR=2
      ENDIF
C------------------------------------------------------------------------
C     END INPUT
C------------------------------------------------------------------------
      INIT=0
      IF (INIT.EQ.0) THEN
        NPREC=65
        CALL FPSET(NPREC)
C       FOR JBASE=10000, NDIG=NPREC/4+2
        NDIG=INT(NPREC/4)+5
        NCDIG=NDIG
        K=0
        CALL FPI2M(K,MZERO)
        K=1
        CALL FPI2M(K,MONE)
        K=2
        CALL FPI2M(K,MTWO)
        K=9
        CALL FPI2M(K,MNINE)
        CALL FPPI(MPI)
        CALL FPM2DP(MPI,PI)
        CALL FPSQRT(MPI,MRTPI)
```

```
C       MRTPI=SQRT(PI)
        CALL FPDIV(MONE,MRTPI,MTEMP)
        CALL FPADD(MTEMP,MTEMP,MTRTPI)
C       MTRTPI=2/RTPI
        DTOL=0.50D-19
        CALL FPDP2M(DTOL,MTOL)
        I=2
        CALL FPDIVI(MONE,I,MDH)
C       INIT=1
C       INIT IS SET TO 1 AFTER THE INITIALIZATION OF MQUAD8
      ENDIF
      N1 = NCHEBY-1
      N2 = NCHEBY-1
      APB=AA+BB
      BMA=BB-AA
      CALL FPDP2M(AA,MAA)
      CALL FPDP2M(BB,MBB)
      CALL FPADD(MAA,MBB,MAPB)
      CALL FPSUB(MBB,MAA,MBMA)
      C1= SNGL(APB)
      C2= SNGL(BMA)
      IF(BB.EQ.AA) C1=SNGL(AA+AA)
      OPEN(10,FILE='MPCHEBY.FIT',STATUS='UNKNOWN')
      DO 500 N=N1,N2
        IF (NAR.GT.N) THEN
          NAR=N+1
        ENDIF
        NP1=N+1
        DN=DBLE(FLOAT(N))
        THETA=PI/DN
        CALL FPDIVI(MPI,N,MTHETA)
        T(1)=1.0D0
        CALL COPYTO(MONE,1,MT)
        T(2)=DCOS(THETA)
        CALL FPCOS(MTHETA,MTEMP)
        CALL COPYTO(MTEMP,2,MT)
        CALL FPEQU(MTEMP,MTONE,NDIG,NDIG)
        TONE=T(2)
        VONE=DSIN(THETA)
        CALL FPSIN(MTHETA,MVONE)
        VK = VONE
        CALL FPEQU(MVONE,MVK,NDIG,NDIG)
C-----------------------------------------------------------------------
C     GET CHEBYSHEV NODES ON [A,B] IN X ARRAY
C     TRIG IDENTITIES SIN((K+1)*X) = SIN(K*X)*COS(X)+COS(K*X)*SIN(X)
C                     COS((K+1)*X) = COS(K*X)*COS(X)-SIN(K*X)*SIN(X)
C     FOR CHEBYSHEV NODES T(K)=COS((K-1)*PI/N), K=1,N+1 BY RECURRENCE
C-----------------------------------------------------------------------
        DO 5 K=2,N
          TK=T(K)
          CALL COPYFR(MT,K,MTK)
          T(K+1)=TONE*TK-VK*VONE
          CALL FPMPY(MTONE,MTK,MTEMP)
          CALL FPMPY(MVK,MVONE,MTEMP1)
          CALL FPSUB(MTEMP,MTEMP1,MTEMP)
          CALL COPYTO(MTEMP,K+1,MT)
          VK=VK*TONE+TK*VONE
          CALL FPMPY(MVK,MTONE,MTEMP)
          CALL FPMPY(MTK,MVONE,MTEMP1)
          CALL FPADD(MTEMP,MTEMP1,MVK)
          X(K)=(TK*BMA+APB)/2.0D0
          CALL FPMPY(MTK,MBMA,MTEMP)
          CALL FPADD(MTEMP,MAPB,MTEMP)
          CALL FPDIVI(MTEMP,2,MTEMP)
          CALL COPYTO(MTEMP,K,MX)
          CALL FPM2DP(MTEMP,TEMP)
          PRINT *,'K, X(K) ',K,TEMP
```

```
C          IF(AA.NE.BB) GO TO 5
C          IF(TK.NE.-1.0D0) GO TO 10
C          X(K)=1.0D+16
C          GO TO 5
C   10     CONTINUE
C          X(K)=(AA+AA)/(1.0D0+TK)
    5   CONTINUE
        X(1)=(T(1)*BMA+APB)/2.0D0
        CALL COPYFR(MT,1,MTEMP)
        CALL FPMPY(MTEMP,MBMA,MTEMP)
        CALL FPADD(MTEMP,MAPB,MTEMP)
        CALL FPDIVI(MTEMP,2,MTEMP)
        CALL COPYTO(MTEMP,1,MX)
        X(NP1)=(T(NP1)*BMA+APB)/2.0D0
        CALL COPYFR(MT,NP1,MTEMP)
        CALL FPMPY(MTEMP,MBMA,MTEMP)
        CALL FPADD(MTEMP,MAPB,MTEMP)
        CALL FPDIVI(MTEMP,2,MTEMP)
        CALL COPYTO(MTEMP,NP1,MX)
C        IF(AA.NE.BB) GO TO 20
C        X(1)=(AA+AA)/(1.0D0+T(1))
C        X(NP1)=1.0D+16
C   20   CONTINUE
C--------------------------------------------------------------------
        WRITE(10, 52)NP1,AA,BB
   52 FORMAT(' NODES='I5/' A    ='D23.14/' B    ='D23.14/)
        WRITE(10, 51)(X(I),I=1,NP1)
   51 FORMAT(' CHEBYSHEV NODES  X(J)'/(1X,3D23.14))
C--------------------------------------------------------------------
C     FUNCTION EVALUATION AT CHEBYSHEV NODES X(I), I=1,NP1
C--------------------------------------------------------------------
        DO 7 I=1,NP1
          CALL COPYFR(MX,I,MTEMP)
          CALL FPM2DP(MTEMP,TEMP1)
          CALL MPGOFX(MTEMP,MTEMP1)
          CALL COPYTO(MTEMP1,I,MF)
          CALL FPM2DP(MTEMP1,TEMP)
          F(I)=TEMP
          PRINT *,'I,X(I),F(I)',I,TEMP1,TEMP
    7   CONTINUE
        F(NP1)=TEMP/2.0D0
        CALL FPDIVI(MTEMP1,2,MTEMP)
        CALL COPYTO(MTEMP,NP1,MF)
C--------------------------------------------------------------------
C     END OF FUNCTION EVALUATION
C--------------------------------------------------------------------
        DO 100 IR=1,NP1
C          Z=T(IR)
          CALL COPYFR(MT,IR,MZ)
C          TZ=Z+Z
          CALL FPMPYI(MZ,2,MTZ)
C          B2=0.0D0
          CALL FPEQU(MZERO,MB2,NDIG,NDIG)
C          B1= 0.0D0
          CALL FPEQU(MZERO,MB1,NDIG,NDIG)
          DO 15 J=1,N
            K=NP1-J+1
            TEMP=B1
            CALL FPEQU(MB1,MTEMP,NDIG,NDIG)
C            B1=TZ*B1-B2+F(K)
            CALL COPYFR(MF,K,MTEMP1)
            CALL FPMPY(MTZ,MB1,MTEMP2)
            CALL FPSUB(MTEMP2,MB2,MTEMP2)
            CALL FPADD(MTEMP2,MTEMP1,MB1)
C            B2=TEMP
            CALL FPEQU(MTEMP,MB2,NDIG,NDIG)
   15     CONTINUE
```

```
C          ANS=Z*B1-B2+F(1)/2.0D0
          CALL COPYFR(MF,1,MTEMP)
          CALL FPDIVI(MTEMP,2,MTEMP)
          CALL FPMPY(MZ,MB1,MTEMP1)
          CALL FPSUB(MTEMP1,MB2,MTEMP1)
          CALL FPADD(MTEMP1,MTEMP,MTEMP)
C          AR(IR)=(ANS+ANS)/DN
          CALL FPMPYI(MTEMP,2,MTEMP)
          CALL FPDIVI(MTEMP,N,MTEMP)
          CALL COPYTO(MTEMP,IR,MAR)
          CALL FPM2DP(MTEMP,AR(IR))
  100   CONTINUE
C-----------------------------------------------------------------------
        WRITE(10,153) (AR(K),K=1,NP1)
  153 FORMAT(' CHEBYSHEV COEFFICIENTS A(J)'/(1X,3D23.14))
C-----------------------------------------------------------------------
        F(NP1)=F(NP1)+F(NP1)
        CALL COPYFR(MF,NP1,MTEMP)
        CALL FPMPYI(MTEMP,2,MTEMP)
        CALL COPYTO(MTEMP,NP1,MF)
C-----------------------------------------------------------------------
        WRITE(10, 54)(F(I),I=1,NP1)
   54 FORMAT(' FUNCTION VALUES AT CHEBYSHEV NODES F(X(J))'/(1X,3D23.14))
C-----------------------------------------------------------------------
        NARM=NAR-1
        DO 30 I=1,NP1
          XX=SNGL(X(I))
          IF(AA.NE.BB) GO TO 26
          SZ=C1/XX-1.0E0
          GO TO 27
   26     CONTINUE
          SZ=(XX+XX-C1)/C2
   27     CONTINUE
          STZ=SZ+SZ
          S2=0.0E0
          S1=0.0E0
          DO 25 J=1,NARM
            K=NAR-J+1
            ST=S1
            SA=AR(K)
            S1=STZ*S1-S2+SA
            S2=ST
   25     CONTINUE
          SA=AR(1)
          FX=SZ*S1-S2+SA/2.0
          T(I)=DBLE(FX)-F(I)
          IF(IERR.EQ.1) THEN
            IF(F(I).NE.0.0D0) THEN
              T(I)=T(I)/F(I)
            ENDIF
          ENDIF
   30   CONTINUE
C-----------------------------------------------------------------------
        WRITE(10, 53)ILAB(IERR),NAR,AA,BB,(T(I),I=1,NP1)
   53   FORMAT(/' SINGLE PRECISION',A10,'ERRORS AT CHEBYSHEV NODES USING
     &',I5,' TERMS'/' OF THE CHEBYSHEV SUM ON',2D23.14//(1X,3D23.14))
C-----------------------------------------------------------------------
        DO 35 I=1,NP1
          IF(AA.NE.BB) GO TO 36
          Z=(AA+AA)/X(I)-1.0D0
          GO TO 37
   36     CONTINUE
          Z=(X(I)+X(I)-APB)/BMA
   37     CONTINUE
          TZ=Z+Z
          B2=0.0D0
          B1=0.0D0
```

```
          DO 40 J=1,NARM
            K=NAR-J+1
            TEMP=B1
            CALL COPYFR(MAR,K,MTEMP)
            CALL FPM2DP(MTEMP,TEMP1)
C            B1=TZ*B1-B2+AR(K)
            B1=TZ*B1-B2+TEMP1
            B2=TEMP
   40     CONTINUE
          CALL COPYFR(MAR,1,MTEMP)
          CALL FPM2DP(MTEMP,TEMP)
C          TEMP=Z*B1-B2+AR(1)/2.0D0
          TEMP=Z*B1-B2+TEMP/2.0D0
          CALL COPYFR(MF,I,MTEMP)
          CALL FPM2DP(MTEMP,TEMP1)
C          T(I)=TEMP-F(I)
          T(I)=TEMP-TEMP1
          IF(IERR.EQ.1) THEN
            IF(F(I).NE.0.0D0) THEN
C              T(I)=T(I)/F(I)
               T(I)=T(I)/TEMP1
            ENDIF
          ENDIF
   35   CONTINUE
C-----------------------------------------------------------------------
        WRITE(10, 55)ILAB(IERR),NAR,AA,BB,(T(I),I=1,NP1)
   55   FORMAT(/' DOUBLE PRECISION',A10,'ERRORS AT CHEBYSHEV NODES USING
     &',I5,' TERMS'/' OF THE CHEBYSHEV SUM ON',2D23.14//(1X,3D23.14))
C-----------------------------------------------------------------------
  500 CONTINUE
      CLOSE(10)
      FNAME='MPCHEBY.DAT'
      FWORK='MPCHEBZ.DAT'
      AR(1)=AR(1)*0.5D0
      CALL COPYFR(MAR,1,MTEMP)
      CALL FPDIVI(MTEMP,2,MTEMP)
      CALL COPYTO(MTEMP,1,MAR)
      OPEN(10,FILE='MPCHEBY.DAT',STATUS='UNKNOWN')
      REWIND(10)
      WRITE(10,531)NP1
 531  FORMAT('NODES= ',I3)
      WRITE(10,530) AA,BB
 530  FORMAT('A= 'D30.18/'B= ',D30.18/)
      CLOSE(10)
C-----------------------------------------------------------------------
C     WRITE NORMAL DATA STATEMENT WITH 14 DIGITS, 3 ON A LINE
C-----------------------------------------------------------------------
      CALL DATAS(NAR,AR,DNAME,FNAME,FWORK,KERR)
C-----------------------------------------------------------------------
C     WRITE DATA STATEMENT IN 2 COLUMNS OF 18 DIGITS
C-----------------------------------------------------------------------
      DO 515 I=1,10
        IF(DNAME(I:I).NE.' ') GO TO 520
  515 CONTINUE
      PRINT *, 'THE DATA NAME IS BLANK'
      STOP
  520 CONTINUE
      K1=I
      DO 501 I=K1,10
        IF(DNAME(I:I).EQ.' ') GO TO 502
 501  CONTINUE
      I=11
 502  CONTINUE
      K2=I-1
C     READ TO THE BOTTOM OF THE FILE (TO SKIP WHAT IS ALREADY THERE)
      OPEN(15,FILE=FNAME,STATUS='UNKNOWN')
      REWIND(15)
```

```
      DO 511 LOOP=1,400
        READ(15,'( )',END=512)
 511  CONTINUE
 512  CONTINUE
      ENDFILE(15)
      BACKSPACE(15)
      OPEN(20,FILE=FWORK,STATUS='UNKNOWN')
      WRITE(20,509) NAR
 509  FORMAT(I2)
      REWIND(20)
      READ(20,510) NAB
 510  FORMAT(A2)
      REWIND (20)
      LINE='      DATA ('//DNAME(K1:K2)//'(I), I=1,'//NAB//')/'
      WRITE(15,506)LINE
 506  FORMAT(A72)
      LBLANK='                                                                        '
      LL=72
      JFORM1=0
      K=1
      NMOD=MOD(NAR,2)
      IF(NMOD.EQ.0) THEN
        NJ=NAR-1
      ELSE
        NJ=NAR-2
      ENDIF
      DO 300 J=1,NJ,2
        WRITE(20,509) K
        REWIND(20)
        READ(20,510) NK
        REWIND(20)
        CALL COPYFR(MAR,J,MTEMP)
        LTEMP='    '//NK//'    '
        CALL FPOUT(MTEMP,LINEA,LL)
        I1=NPREC+4
        I2=I1+3
        LINEA=LINEA(1:22)//'D'//LINEA(I1:I2)//',  '
        LTEMP=LTEMP(1:8)//LINEA(1:30)
        CALL COPYFR(MAR,J+1,MTEMP)
        CALL FPOUT(MTEMP,LINEA,LL)
        IF (J.EQ.NJ) THEN
          IF(NMOD.EQ.0) THEN
            LINEA=LINEA(1:22)//'D'//LINEA(I1:I2)//'/'
          ELSE
            LINEA=LINEA(1:22)//'D'//LINEA(I1:I2)//','
          ENDIF
        ELSE
          LINEA=LINEA(1:22)//'D'//LINEA(I1:I2)//','
        ENDIF
        LTEMP=LTEMP(1:38)//LINEA(1:28)//LBLANK
        LINE=LTEMP(1:72)
        WRITE(15,301)LINE
 301    FORMAT(A72)
        K=K+1
 300  CONTINUE
      IF(NMOD.NE.0) THEN
        WRITE(20,509) K
        REWIND(20)
        READ(20,510) NK
        REWIND(20)
        CALL COPYFR(MAR,NAR,MTEMP)
        LTEMP='    '//NK//'    '
        CALL FPOUT(MTEMP,LINEA,LL)
        LINEA=LINEA(1:22)//'D'//LINEA(64:67)//'/  '
        LTEMP=LTEMP(1:8)//LINEA(1:28)//LBLANK
        LINE=LTEMP(1:72)
        WRITE(15,301)LINE
```

```
      ENDIF
      CLOSE(15)
C-----------------------------------------------------------------------
      IF (KERR.NE.0) THEN
        PRINT *,' ERROR IN DATAS, IERR = ',KERR
      ELSE
        PRINT *,'  '
        PRINT *,' APPROXIMATION USING ',NP1,' CHEBYSHEV NODES'
        PRINT *,'  '
        PRINT *,' FILES WRITTEN:'
        PRINT *,'  '
        PRINT *,' MPCHEBY.FIT:    THE COEFFIENTS, NODES, AND ERRORS,'
        PRINT *,' MPCHEBY.DAT:    THE DATA STATEMENT,'
        PRINT *,' MPCHEBZ.DAT:    SCRATCH FILE.'
      ENDIF
      PAUSE
      STOP
      END
      SUBROUTINE COPYTO(MA,K,MB)
C     COPY VECTOR MA to KTH COLUMN OF MATRIX MB
      DIMENSION MB(200,100),MA(200)
      DO 10 J=1,200
        MB(J,K)=MA(J)
 10   CONTINUE
      RETURN
      END
      SUBROUTINE COPYFR(MB,K,MA)
C     COPY from KTH COLUMN OF MATRIX MB TO VECTOR MA
      DIMENSION MB(200,100),MA(200)
      DO 10 J=1,200
        MA(J)=MB(J,K)
 10   CONTINUE
      RETURN
      END
      SUBROUTINE MPGOFXX(MX,MGOFX)
      DIMENSION MX(200),MGOFX(200)
      COMMON /CWORK/MPI(200),MRTPI(200),MONE(200),MTWO(200),MTRTPI(200),
     *MZERO(200),MTOL(200),MDH(200),MNINE(200),NCDIG
      CALL FPDIV(MONE,MX,MGOFX)
      RETURN
      END
      SUBROUTINE MPGOFXY(MT,MFUN)
C     COMPUTES FOR ERF(X)
      LOGICAL FPCOMP
      DIMENSION MT(200),MT2(200),MTRM(200),MSUM(200),MCI(200),MCNT(200),
     &MTRM1(200),MTRMA(200),MTRMS(200),MTEMP(200),MSUMA(200),MAK(200),
     &MFUN(200),MSUMT(200)
      COMMON /CWORK/MPI(200),MRTPI(200),MONE(200),MTWO(200),MTRTPI(200),
     *MZERO(200),MTOL(200),MDH(200),MNINE(200),NCDIG
C
C     CALCULATE ERF ON [0,2]
C
      NDIG=NCDIG
      CALL FPMPY(MT,MT,MT2)
      CALL FPEQU(MONE,MTRM,NDIG,NDIG)
      CALL FPEQU(MONE,MSUM,NDIG,NDIG)
      J=-1
      IF(FPCOMP(MT,"LT",MNINE)) THEN
        CALL FPEQU(MONE,MCI,NDIG,NDIG)
        CALL FPEQU(MTWO,MCNT,NDIG,NDIG)
        CALL FPADD(MONE,MCNT,MCNT)
C         CALL FPDIV(MONE,MT,MTRM1)
        DO 10 I=1,1000
          CALL FPMPY(MTRM,MT2,MTRM)
          CALL FPDIV(MTRM,MCI,MTRM)
          CALL FPMPYI(MTRM,J,MTRM)
          CALL FPDIV(MTRM,MCNT,MTRMS)
```

```
          CALL FPADD(MSUM,MTRMS,MSUM)
          IF(I.LT.5) GOTO 5
          CALL FPMPY(MSUM,MTRTPI,MSUMT)
          CALL FPABS(MTRMS,MTRMA)
          CALL FPABS(MSUMT,MSUMA)
          CALL FPMPY(MTOL,MSUMA,MSUMA)
          IF (FPCOMP(MTRMA,'LT',MSUMA)) GOTO 15
  5       CONTINUE
          CALL FPADD(MCI,MONE,MCI)
          CALL FPADD(MCNT,MTWO,MCNT)
  10    CONTINUE
        PRINT*,"DROP THRU SERIES LOOP"
        PAUSE
  15    CONTINUE
      ELSE
        CALL FPEQU(MDH,MAK,NDIG,NDIG)
        DO 20 I=1,25
          CALL FPMPY(MTRM,MAK,MTRM)
          CALL FPDIV(MTRM,MT2,MTRM)
          CALL FPABS(MTRM,MTRMA)
          IF (FPCOMP(MTRMA,'LE',MTOL)) GOTO 25
          CALL FPMPYI(MTRM,J,MTRM)
          CALL FPADD(MSUM,MTRM,MSUM)
          CALL FPADD(MAK,MONE,MAK)
  20    CONTINUE
        PRINT*,"DROP THRU ASYMPTOTOIC SERIES LOOP"
        PAUSE
  25    CONTINUE
        CALL FPMPYI(MT2,J,MAK)
        CALL FPEXP(MAK,MTRM)
        CALL FPMPY(MSUM,MTRM,MSUM)
        CALL FPMPY(MRTPI,MT,MAK)
        CALL FPDIV(MSUM,MAK,MSUMT)
      ENDIF
      CALL FPEQU(MSUMT,MFUN,NDIG,NDIG)
      RETURN
      END
      SUBROUTINE MPGOFX(MX,MGOFX)
C
C     COMPUTES INT ON (X,INF) OF ERFC(X)/X IN MP ARITHMETIC USING
C     GAUS8 FORMULA ON REPEATED SUBDIVIDED INTERVALS
C
C      DOUBLE PRECISION DSUM
      LOGICAL FPCOMP
      DIMENSION MSUM(200),MSUMA(200),MANS(200),MANSA(200),MTOLA(200),
     &MTEMP(200),MA(200),MB(200),MX(200),MSUM1(200),MDX(200)
C
      DIMENSION MGOFX(200)
C
      COMMON /CWORK/MPI(200),MRTPI(200),MONE(200),MTWO(200),MTRTPI(200),
     *MZERO(200),MTOL(200),MDH(200),MNINE(200),NCDIG
C
      NDIG=NCDIG
      KL=10
      CALL FPMPYI(MTOL,KL,MTOLA)
      CALL FPEQU(MDH,MDX,NDIG,NDIG)
      CALL FPEQU(MZERO,MSUM1,NDIG,NDIG)
      KL=100
      DO 10 J=1,100
        CALL FPDIVI(MDX,2,MDX)
        KL=KL+KL
        CALL FPEQU(MZERO,MSUM,NDIG,NDIG)
        CALL FPEQU(MX,MB,NDIG,NDIG)
        DO 20 K=1,KL
          CALL FPEQU(MB,MA,NDIG,NDIG)
          CALL FPADD(MA,MDX,MB)
          CALL MQUAD8(MA,MB,MANS)
```

```
          CALL FPADD(MSUM,MANS,MSUM)
          IF(J.EQ.1) THEN
            CALL FPABS(MANS,MANSA)
            CALL FPABS(MSUM,MSUMA)
            CALL FPMPY(MSUMA,MTOL,MSUMA)
            IF (FPCOMP(MANSA,'LT',MSUMA)) GOTO 25
          ENDIF
 20     CONTINUE
 25     CONTINUE
        IF(J.EQ.1) THEN
          KL=K
          CALL FPEQU(MSUM,MSUM1,NDIG,NDIG)
          GOTO 10
        ENDIF
        CALL FPSUB(MSUM,MSUM1,MTEMP)
        CALL FPABS(MTEMP,MTEMP)
        CALL FPABS(MSUM,MSUMA)
        CALL FPMPY(MSUMA,MTOLA,MSUMA)
        IF (FPCOMP(MTEMP,'LE',MSUMA)) GOTO 30
        CALL FPEQU(MSUM,MSUM1,NDIG,NDIG)
C        CALL FPM2DP(MSUM,DSUM)
C        PRINT *,KL,DSUM
 10   CONTINUE
 30   CONTINUE
      CALL FPMPY(MX,MX,MTEMP)
      CALL FPEXP(MTEMP,MB)
C      CALL FPMPY(MTEMP,MX,MA)
C      CALL FPMPY(MA,MB,MTEMP)
      CALL FPMPY(MTEMP,MB,MTEMP)
      CALL FPMPY(MTEMP,MSUM,MGOFX)
      RETURN
      END
      SUBROUTINE MQUAD8(MA,MB,MQUAD)
      DIMENSION MA(200),MB(200),MAPB(200),MBMA(200),MTEMP1(200),
     &MTEMP2(200),MTEMP3(200),MTEMP4(200),MTEMP5(200),MTEMP6(200),
     &MTEMP7(200),MTEMP8(200),MSUM(200),MFUN(200),MQUAD(200)
      DIMENSION IX1(4),IX2(4),IX3(4),IX4(4),IW1(4),IW2(4),IW3(4),IW4(4)
C
      COMMON /CINIT/INIT
      COMMON /CWORK/MPI(200),MRTPI(200),MONE(200),MTWO(200),MTRTPI(200),
     *MZERO(200),MTOL(200),MDH(200),MNINE(200),NCDIG
      COMMON /GAUSS/ MX1(200),MX2(200),MX3(200),MX4(200),MW1(200),
     &MW2(200),MW3(200),MW4(200)
C
C     GAUSS ABSCISSAS AND WEIGHTS IN MP
      DATA (IX1(J),J=1,4)/0,18343464,24956498,05000000/
      DATA (IX2(J),J=1,4)/0,52553240,99163289,86000000/
      DATA (IX3(J),J=1,4)/0,79666647,74136267,40000000/
      DATA (IX4(J),J=1,4)/0,96028985,64975362,32000000/
      DATA (IW1(J),J=1,4)/0,36268378,33783619,83000000/
      DATA (IW2(J),J=1,4)/0,31370664,58778872,87000000/
      DATA (IW3(J),J=1,4)/0,22238103,44533744,71000000/
      DATA (IW4(J),J=1,4)/0,10122853,62903762,59000000/
C
      NDIG=NCDIG
      IF (INIT.EQ.0) THEN
        CALL FPEQU(MZERO,MX1,NDIG,NDIG)
        MX1(1)=IX1(1)
        MX1(2)=IX1(2)
        MX1(3)=IX1(3)
        MX1(4)=IX1(4)
        CALL FPEQU(MZERO,MX2,NDIG,NDIG)
        MX2(1)=IX2(1)
        MX2(2)=IX2(2)
        MX2(3)=IX2(3)
        MX2(4)=IX2(4)
        CALL FPEQU(MZERO,MX3,NDIG,NDIG)
```

```
          MX3(1)=IX3(1)
          MX3(2)=IX3(2)
          MX3(3)=IX3(3)
          MX3(4)=IX3(4)
          CALL FPEQU(MZERO,MX4,NDIG,NDIG)
          MX4(1)=IX4(1)
          MX4(2)=IX4(2)
          MX4(3)=IX4(3)
          MX4(4)=IX4(4)
          CALL FPEQU(MZERO,MW1,NDIG,NDIG)
          MW1(1)=IW1(1)
          MW1(2)=IW1(2)
          MW1(3)=IW1(3)
          MW1(4)=IW1(4)
          CALL FPEQU(MZERO,MW2,NDIG,NDIG)
          MW2(1)=IW2(1)
          MW2(2)=IW2(2)
          MW2(3)=IW2(3)
          MW2(4)=IW2(4)
          CALL FPEQU(MZERO,MW3,NDIG,NDIG)
          MW3(1)=IW3(1)
          MW3(2)=IW3(2)
          MW3(3)=IW3(3)
          MW3(4)=IW3(4)
          CALL FPEQU(MZERO,MW4,NDIG,NDIG)
          MW4(1)=IW4(1)
          MW4(2)=IW4(2)
          MW4(3)=IW4(3)
          MW4(4)=IW4(4)
          PRINT *,'GAUSSIAN ABSCISSAS AND WEIGHTS '
          CALL FPPRNT(MX1)
          CALL FPPRNT(MX2)
          CALL FPPRNT(MX3)
          CALL FPPRNT(MX4)
          CALL FPPRNT(MW1)
          CALL FPPRNT(MW2)
          CALL FPPRNT(MW3)
          CALL FPPRNT(MW4)
          PAUSE
          INIT=1
        ENDIF
        CALL FPADD(MA,MB,MAPB)
        CALL FPDIVI(MAPB,2,MAPB)
        CALL FPSUB(MB,MA,MBMA)
        CALL FPDIVI(MBMA,2,MBMA)
        CALL FPMPY(MX1,MBMA,MTEMP1)
        CALL FPMPY(MX2,MBMA,MTEMP2)
        CALL FPMPY(MX3,MBMA,MTEMP3)
        CALL FPMPY(MX4,MBMA,MTEMP4)
        I=-1
        CALL FPMPYI(MTEMP1,I,MTEMP5)
        CALL FPMPYI(MTEMP2,I,MTEMP6)
        CALL FPMPYI(MTEMP3,I,MTEMP7)
        CALL FPMPYI(MTEMP4,I,MTEMP8)
C
        CALL FPADD(MTEMP1,MAPB,MTEMP1)
        CALL FPADD(MTEMP2,MAPB,MTEMP2)
        CALL FPADD(MTEMP3,MAPB,MTEMP3)
        CALL FPADD(MTEMP4,MAPB,MTEMP4)
        CALL FPADD(MTEMP5,MAPB,MTEMP5)
        CALL FPADD(MTEMP6,MAPB,MTEMP6)
        CALL FPADD(MTEMP7,MAPB,MTEMP7)
        CALL FPADD(MTEMP8,MAPB,MTEMP8)
C
        CALL FPEQU(MZERO,MSUM,NDIG,NDIG)
        CALL MFOFX(MTEMP1,MFUN)
        CALL FPMPY(MW1,MFUN,MFUN)
```

```
      CALL FPADD(MSUM,MFUN,MSUM)
      CALL MFOFX(MTEMP2,MFUN)
      CALL FPMPY(MW2,MFUN,MFUN)
      CALL FPADD(MSUM,MFUN,MSUM)
      CALL MFOFX(MTEMP3,MFUN)
      CALL FPMPY(MW3,MFUN,MFUN)
      CALL FPADD(MSUM,MFUN,MSUM)
      CALL MFOFX(MTEMP4,MFUN)
      CALL FPMPY(MW4,MFUN,MFUN)
      CALL FPADD(MSUM,MFUN,MSUM)
      CALL MFOFX(MTEMP5,MFUN)
      CALL FPMPY(MW1,MFUN,MFUN)
      CALL FPADD(MSUM,MFUN,MSUM)
      CALL MFOFX(MTEMP6,MFUN)
      CALL FPMPY(MW2,MFUN,MFUN)
      CALL FPADD(MSUM,MFUN,MSUM)
      CALL MFOFX(MTEMP7,MFUN)
      CALL FPMPY(MW3,MFUN,MFUN)
      CALL FPADD(MSUM,MFUN,MSUM)
      CALL MFOFX(MTEMP8,MFUN)
      CALL FPMPY(MW4,MFUN,MFUN)
      CALL FPADD(MSUM,MFUN,MSUM)
      CALL FPMPY(MSUM,MBMA,MQUAD)
      RETURN
      END
      SUBROUTINE MFOFX(MT,MFUN)
C       DOUBLE PRECISION DSUM,DT
      LOGICAL FPCOMP
      DIMENSION MT(200),MT2(200),MTRM(200),MSUM(200),MCI(200),MCNT(200),
     &MTRM1(200),MTRMA(200),MTRMS(200),MTEMP(200),MSUMA(200),MAK(200),
     &MFUN(200),MSUMT(200)
      COMMON /CWORK/MPI(200),MRTPI(200),MONE(200),MTWO(200),MTRTPI(200),
     *MZERO(200),MTOL(200),MDH(200),MNINE(200),NCDIG
C
      NDIG=NCDIG
      CALL FPMPY(MT,MT,MT2)
      CALL FPEQU(MONE,MTRM,NDIG,NDIG)
      CALL FPEQU(MONE,MSUM,NDIG,NDIG)
      J=-1
      IF(FPCOMP(MT,"LT",MNINE)) THEN
        CALL FPEQU(MONE,MCI,NDIG,NDIG)
        CALL FPEQU(MTWO,MCNT,NDIG,NDIG)
        CALL FPADD(MONE,MCNT,MCNT)
C         CALL FPDIV(MONE,MT,MTRM1)
        DO 10 I=1,1000
          CALL FPMPY(MTRM,MT2,MTRM)
          CALL FPDIV(MTRM,MCI,MTRM)
          CALL FPMPYI(MTRM,J,MTRM)
          CALL FPDIV(MTRM,MCNT,MTRMS)
          CALL FPADD(MSUM,MTRMS,MSUM)
          IF(I.LT.5) GOTO 5
          CALL FPMPY(MSUM,MTRTPI,MTEMP)
          CALL FPMPY(MTEMP,MT,MTEMP)
C           CALL FPSUB(MTRM1,MTEMP,MSUMT)
          CALL FPSUB(MONE,MTEMP,MSUMT)
          CALL FPABS(MTRMS,MTRMA)
          CALL FPABS(MSUMT,MSUMA)
          CALL FPMPY(MTOL,MSUMA,MSUMA)
          IF (FPCOMP(MTRMA,'LT',MSUMA)) GOTO 15
  5       CONTINUE
          CALL FPADD(MCI,MONE,MCI)
          CALL FPADD(MCNT,MTWO,MCNT)
  10    CONTINUE
        PRINT*,"DROP THRU SERIES LOOP"
        PAUSE
  15    CONTINUE
      ELSE
```

```
        CALL FPEQU(MDH,MAK,NDIG,NDIG)
        DO 20 I=1,25
          CALL FPMPY(MTRM,MAK,MTRM)
          CALL FPDIV(MTRM,MT2,MTRM)
          CALL FPABS(MTRM,MTRMA)
          CALL FPABS(MSUM,MSUMA)
          CALL FPMPY(MTOL,MSUMA,MSUMA)
          IF (FPCOMP(MTRMA,'LE',MSUMA)) GOTO 25
          CALL FPMPYI(MTRM,J,MTRM)
          CALL FPADD(MSUM,MTRM,MSUM)
          CALL FPADD(MAK,MONE,MAK)
  20    CONTINUE
        PRINT*,"DROP THRU ASYMPTOTOIC SERIES LOOP"
        PAUSE
  25    CONTINUE
        CALL FPMPYI(MT2,J,MAK)
        CALL FPEXP(MAK,MTRM)
        CALL FPMPY(MSUM,MTRM,MSUM)
C        CALL FPMPY(MRTPI,MT2,MAK)
        CALL FPMPY(MRTPI,MT,MAK)
        CALL FPDIV(MSUM,MAK,MSUMT)
      ENDIF
      CALL FPEQU(MSUMT,MFUN,NDIG,NDIG)
C      CALL FPM2DP(MSUMT,DSUM)
C      CALL FPM2DP(MT,DT)
C      PRINT *,I,DT,DSUM
C      PAUSE
      RETURN
      END
      SUBROUTINE DATAS(N,A,DNAME,FNAME,FWORK,IERR)
C
C     WRITTEN BY D.E.AMOS, MAY, 1990.
C
C     REFERENCE
C
C     ABSTRACT
C        DATAS FORMATS A DOUBLE PRECISION SINGLY DIMENSIONED ARRAY
C        A(I), I=1,N INTO A DATA STATEMENT IN FILE FNAME. THE A
C        VECTOR IS FORMATTED BY 3E21.14 ACROSS A LINE AND GIVEN THE
C        NAME SPECIFIED BY THE PARAMETER DNAME. DATAS ALSO CREATES
C        A TEMPORARY FILE TEMPX.DAT.
C
C     DESCRIPTION OF ARGUMENTS
C
C        INPUT
C           N      -  NUMBER OF ELEMENTS OF A, 1.LE.N.LE.99
C           A      -  DOUBLE PRECISION ARRAY OF DIMENSION AT LEAST N
C           DNAME  -  CHARACTER*K VARIABLE FOR THE NAME OF THE DATA
C                     IN THE FILE DESIGNATED BY FNAME, 1.LE.K.LE.10
C           FNAME  -  CHARACTER*K VARIABLE FOR THE FILE NAME WHERE THE
C                     DATA STATEMENT IS TO BE WRITTEN, 1.LE.K.LE.12
C           FWORK  -  CHARACTER*K VARIABLE FOR A SCRATCH FILE NAME,
C                     1.LE.K.LE.12
C
C
C        OUTPUT
C           ****   -  FILE WITH ALIAS FNAME WHICH CONTAINS THE DATA
C                     STATEMENT
C           ****      FILE WITH ALIAS FWORK USED AS A SCRATCH FILE
C           IERR   -  ERROR INDICATOR
C                     IERR=0  NORMAL RETURN, NO ERRORS
C                     IERR=1  N IS LESS THAN 1
C                     IERR=2  N IS LARGER THAN 99
C                     IERR=3  DNAME IS BLANK
C                     IERR=4  FNAME IS BLANK
C                     IERR=5  FWORK IS BLANK
C
```

```
C     ERROR CONDITIONS
C        N LESS THAN 1 OR GREATER THAN 99 IS AN ERROR
C        EITHER DNAME OR FNAME BLANK IS AN ERROR
C-----------------------------------------------------------------------
      CHARACTER*(*) DNAME,FNAME,FWORK
      CHARACTER*72 LINE
      CHARACTER*2 NC
      CHARACTER*21 B(99)
      DOUBLE PRECISION A(N)
      IERR=0
      IF (N.GT.99) THEN
        IERR=2
        RETURN
      ENDIF
      IF (N.LT.1) THEN
        IERR=1
        RETURN
      ENDIF
      DO 5 I=1,12
        IF(FNAME(I:I).NE.' ') GO TO 6
    5 CONTINUE
      IERR=4
      RETURN
    6 CONTINUE
      DO 7 I=1,12
        IF(FWORK(I:I).NE.' ') GO TO 8
    7 CONTINUE
      IERR=5
      RETURN
    8 CONTINUE
      DO 15 I=1,10
        IF(DNAME(I:I).NE.' ') GO TO 20
   15 CONTINUE
      IERR=3
      RETURN
   20 CONTINUE
      K1=I
      DO 25 I=K1,10
        IF(DNAME(I:I).EQ.' ') GO TO 30
   25 CONTINUE
      I=11
   30 CONTINUE
      K2=I-1
      OPEN(10,FILE=FNAME,STATUS='UNKNOWN')
      REWIND(10)
C     POSITION FILE AT END OF LAST LINE
      DO 31 LOOP=1,400
        READ(10,'( )',END=32)
 31   CONTINUE
 32   CONTINUE
      ENDFILE(10)
      BACKSPACE(10)
C
      OPEN(20,FILE=FWORK,STATUS='UNKNOWN')
      WRITE(20,800) N
      REWIND(20)
      READ(20,801) NC
      REWIND(20)
      LINE='      DATA ('//DNAME(K1:K2)//'(I), I=1,'//NC//')/'
      WRITE(10,900)LINE
      DO 35 I=1,N
        WRITE(20,901) A(I)
   35 CONTINUE
      REWIND(20)
      DO 40 I=1,N
        READ(20,902) B(I)
   40 CONTINUE
```

```
      K=0
      DO 45 I=1,N
        LINE='     &'
        M=6
        DO 50 J=1,3
          K=K+1
          LINE=LINE(1:M)//B(K)(1:21)
          M=M+21
          IF(K.EQ.N) THEN
            LINE=LINE(1:M)//'/'
            GO TO 60
          ELSE
            LINE=LINE(1:M)//','
          ENDIF
          M=M+1
   50   CONTINUE
   60   CONTINUE
        WRITE(10,900)LINE
        IF (K.EQ.N) GO TO 70
   45 CONTINUE
   70 CONTINUE
      CLOSE(10)
C-----------------------------------------------------------------------
C     CLEAN UP THE TEMPORARY FILE
C-----------------------------------------------------------------------
      LINE='  '
      REWIND(20)
      WRITE(20,900) LINE
      CLOSE(20)
  800 FORMAT(I2)
  801 FORMAT(A2)
  900 FORMAT(A72)
  901 FORMAT(D21.14)
  902 FORMAT(A21)
      RETURN
      END
************************************************************************
      PROGRAM MCHEBYEV
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      CHARACTER*72 LINE
      COMMON /CINIT/INIT
      DIMENSION COEF(100)
      OPEN(7,FILE='MCHEBYEV.TXT',STATUS='UNKNOWN')
C-----------------------------------------------------------------------
C     READ A, B AND 18 DIGIT COEFFICIENTS FROM FILE CHEBY.DAT
C-----------------------------------------------------------------------
      OPEN(10,FILE='MPCHEBY.DAT',STATUS='UNKNOWN')
      REWIND(10)
      READ(10,196)NODES
 196  FORMAT(7X,I3)
      READ(10,199) A
 199  FORMAT(3X,D30.18)
      READ(10,199) B
      DO 4 J=1,3
C       SKIP OVER THE FIRST DATA STATEMENT
        READ(10,200) LINE
 4    CONTINUE
      DO 5 J=1,400
        READ(10,200) LINE
C        PRINT *,LINE
 200    FORMAT(A72)
        IF(LINE(7:10).EQ.'DATA') GOTO 201
 5    CONTINUE
      STOP
 201  CONTINUE
      M=1
      DO 6 J=1,400
```

```
        READ(10,202,IOSTAT=K) COEF(M),COEF(M+1)
C        PRINT *,J,K,COEF(M),COEF(M+1)
        IF(K.NE.0) GOTO 7
        M=M+2
 202    FORMAT(8X,D26.20,4X,D26.20)
 6    CONTINUE
 7    CONTINUE
C     K=-1 ONLY IF THE READ PRODUCES TWO ZERO ENTRIES
      CLOSE(10)
C
      IF(COEF(M).EQ.0.0D0.AND.COEF(M+1).EQ.0.0D0) THEN
        M=M-1
        IF(COEF(M).EQ.0.0D0) THEN
          M=M-1
        ENDIF
      ELSE
        IF(COEF(M).EQ.0.0D0) THEN
          M=M-1
        ENDIF
      ENDIF
C
      INIT=0
 400  CONTINUE
C      PRINT 195, NODES
      PRINT 198, NODES,A,B
      PRINT 194
      K=1
      MF=INT(M/2)
      DO 18 MM=1,MF
        PRINT 203, MM,COEF(K),COEF(K+1)
        K=K+2
 18   CONTINUE
      IF(MOD(M,2).NE.0) THEN
        PRINT 203, MM,COEF(M)
      ENDIF
 9    CONTINUE
C      PRINT *, ' '
      PRINT *, M,' COEFFICIENTS READ FROM FILE MPCHEBY.DAT'
C      PRINT *, ' '
      PRINT *,'INPUT THE NUMBER OF COEFFICIENTS TO BE USED IN THE SUM (S
     &TOP=0):  '
      READ *,N
      IF (N.GT.M) GOTO 9
      IF (N.EQ.0) STOP
      REWIND(7)
C      WRITE(7,195) NODES
C 195  FORMAT(' NODES= ',I3)
      WRITE(7,198) NODES,A,B
 198  FORMAT('  NODES= ',I3,'  A=',D25.18,' B=',D25.18)
      WRITE(7,194)
 194  FORMAT(7X,'CHEBYSHEV COEFFICIENTS')
      K=1
      MF=INT(M/2)
      DO 8 MM=1,MF
        WRITE(7,203) MM,COEF(K),COEF(K+1)
        K=K+2
 8    CONTINUE
      IF(MOD(M,2).NE.0) THEN
        WRITE(7,203) MM,COEF(M)
      ENDIF
 203  FORMAT(2X,I5,2D30.18)
      WRITE(7,193)N
 193  FORMAT(7X,'NUMBER OF COEFFICIENTS USED IN THE SUM: ',I3)
      WRITE(7,197)
 197  FORMAT(/'               X            Y         REL ERR')
      PRINT 197
      APB=A+B
```

```
      BMA=B-A
      AJ=1.0D0
      JL=51
      AJL=DBLE(FLOAT(JL))
      DO 10 J=1,JL
        X=((AJ-1.0D0)*B+(AJL-AJ)*A)/(AJL-1.0D0)
C        X2=X*X
        CALL DCHBYS(APB,BMA,X,N,COEF,ANS)
C        X3=X2*X
C        ANS=ANS*DEXP(-X2)/X3
C         Y=MPGOFX(X)
        Y=DPGOFX(X)
C        Y=1.0D0/X
        ERR=DABS(ANS-Y)/DABS(Y)
        WRITE(7,300) J,X,Y,ERR
 300    FORMAT(4X,I5,3D13.5)
        PRINT 300, J,X,Y,ERR
        IF(MOD(J,20).EQ.0) PAUSE
        AJ=AJ+1.0D0
 10   CONTINUE
      PAUSE
      DO 401 J=1,25
        PRINT *
 401  CONTINUE
      GOTO 400
      END
      SUBROUTINE DCHBYS(APB,BMA,X,N,COEF,SUM)
C-----------------------------------------------------------------------
C     CHBYS SUMS A CHEBYSHEV SERIES WITH COEFFICIENTS COEF(I), I=1,N
C     ON [A,B] AT X. APB=A+B , BMA=B-A
C-----------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION COEF(N)
      T=(X+X-APB)/BMA
      TT=T+T
      F1=0.0D0
      F2=0.0D0
      NM=N-1
      DO 160 I=1,NM
        J=N-I+1
        TEMP=F1
        F1=TT*F1-F2+COEF(J)
        F2=TEMP
  160 CONTINUE
      SUM=T*F1-F2+COEF(1)
      RETURN
      END
      DOUBLE PRECISION FUNCTION GOFX(X)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON /CFOFX/ CX
C      DATA RTPI /1.772453850905516D0/
      EXTERNAL FOFX
      CX=X
      INIT=0
      X1=X
      SIG=2.0D0
      REL=0.50D-15
C      PRINT *,"X=",X
C      PAUSE
      CALL DQUAD8(FOFX,INIT,X1,SIG,REL,X2,QANS,IERR)
      SIG=6.0D0
      CALL DQUAD8(FOFX,INIT,X1,SIG,REL,X2,QANS,IERR)
      XSQ=X*X
      GOFX=QANS*XSQ*X
      GOFX=GOFX*DEXP(XSQ)
C      GOFX=QANS
C     PRINT *,"GOFX=",X,GOFX
```

```
C      PAUSE
      RETURN
      END
      DOUBLE PRECISION FUNCTION FOFX(X)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      KODE=2
      FOFX=DRERF(X,KODE,NZ)/X
      RETURN
      END
      DOUBLE PRECISION FUNCTION DPGOFX(X)
      DOUBLE PRECISION X,PI,DH,DTOL
      DOUBLE PRECISION DPMAX
C     SINGLE PRECISION SPMAX

      PARAMETER ( MXNDIG=256 , NBITS=32 ,
     *          LPACK  = (MXNDIG+1)/2 + 1 , LUNPCK = (6*MXNDIG)/5 + 20,
     *          LMWA   = 2*LUNPCK         , LJSUMS = 8*LUNPCK,
     *          LMBUFF = ((LUNPCK+3)*(NBITS-1)*301)/2000 + 6  )
C
C
      DIMENSION MX(200),MGOFX(200)

C
      COMMON /FMUSER/ NDIG,JBASE,JFORM1,JFORM2,KRAD,
     *                KW,NTRACE,LVLTRC,KFLAG,KWARN,KROUND
C
      COMMON /FM/ MWA(LMWA),NCALL,MXEXP,MXEXP2,KARGSW,KEXPUN,KEXPOV,
     *            KUNKNO,IUNKNO,RUNKNO,MXBASE,MXNDG2,KSTACK(19),
     *            MAXINT,SPMAX,DPMAX
C
      COMMON /FMSAVE/ NDIGPI,NJBPI,NDIGE,NJBE,NDIGLB,NJBLB,NDIGLI,NJBLI,
     *                MPISAV(LUNPCK),MESAV(LUNPCK),MLBSAV(LUNPCK),
     *                MLN1(LUNPCK),MLN2(LUNPCK),MLN3(LUNPCK),
     *                MLN4(LUNPCK)
C
      COMMON /CINIT/INIT
      COMMON /CWORK/MPI(200),MRTPI(200),MONE(200),MTWO(200),MTRTPI(200),
     *MZERO(200),MTOL(200),MDH(200),MNINE(200),NCDIG
      COMMON /GAUSS/ MX1(200),MX2(200),MX3(200),MX4(200),MW1(200),
     &MW2(200),MW3(200),MW4(200)
C
C-----------------------------------------------------------------------
      IF (INIT.EQ.0) THEN
        NPREC=60
        CALL FPSET(NPREC)
C       FOR JBASE=10000, NDIG=NPREC/4+2
        NDIG=INT(NPREC/4)+5
        NCDIG=NDIG
        K=0
        CALL FPI2M(K,MZERO)
        K=1
        CALL FPI2M(K,MONE)
        K=2
        CALL FPI2M(K,MTWO)
        K=9
        CALL FPI2M(K,MNINE)
        CALL FPPI(MPI)
        CALL FPM2DP(MPI,PI)
        CALL FPSQRT(MPI,MRTPI)
        CALL FPDIV(MONE,MRTPI,MTEMP)
        CALL FPADD(MTEMP,MTEMP,MTRTPI)
C       MTRTPI=2/RTPI
        DTOL=0.50D-16
        CALL FPDP2M(DTOL,MTOL)
        DH=0.5D0
        CALL FPDP2M(DH,MDH)
C       INIT=1
```

```
C         INIT IS SET TO 1 AFTER THE INITIALIZATION OF MQUAD8
      ENDIF
      CALL FPDP2M(X,MX)
      CALL MPGOFX(MX,MGOFX)
C      CALL MFOFXA(MX,MGOFX)
      CALL FPM2DP(MGOFX,DPGOFX)
      END
      SUBROUTINE MPGOFX(MX,MGOFX)
C
C     COMPUTES INT ON (X,INF) OF ERFC(X)/X OR ERFC(X) IN MP ARITHMETIC
C     USING GAUS8 FORMULA ON REPEATED SUBDIVIDED INTERVALS
C
      DOUBLE PRECISION DSUM
      LOGICAL FPCOMP
C
      DIMENSION MX(200),MGOFX(200)
C
      DIMENSION MSUM(200),MSUMA(200),MANS(200),MANSA(200),
     &MTEMP(200),MSUM1(200),MDX(200),MTEMP1(200),MA(200),MB(200)
C
      COMMON /CWORK/MPI(200),MRTPI(200),MONE(200),MTWO(200),MTRTPI(200),
     *MZERO(200),MTOL(200),MDH(200),MNINE(200),NCDIG
C
      NDIG=NCDIG
      CALL FPEQU(MDH,MDX,NDIG,NDIG)
      CALL FPEQU(MZERO,MSUM1,NDIG,NDIG)
      KL=100
      DO 10 J=1,100
        CALL FPDIVI(MDX,2,MDX)
        KL=KL+KL
        CALL FPEQU(MZERO,MSUM,NDIG,NDIG)
        CALL FPEQU(MX,MB,NDIG,NDIG)
        DO 20 K=1,KL
          CALL FPEQU(MB,MA,NDIG,NDIG)
          CALL FPADD(MA,MDX,MB)
          CALL MQUAD8(MA,MB,MANS)
          CALL FPADD(MSUM,MANS,MSUM)
          CALL FPM2DP(MSUM,DSUM)
          IF(J.EQ.1) THEN
            CALL FPABS(MANS,MANSA)
            CALL FPABS(MSUM,MSUMA)
            CALL FPMPY(MSUMA,MTOL,MSUMA)
            IF (FPCOMP(MANSA,'LT',MSUMA)) GOTO 25
          ENDIF
 20     CONTINUE
 25     CONTINUE
        IF(J.EQ.1) THEN
          KL=K
          CALL FPEQU(MSUM,MSUM1,NDIG,NDIG)
          GOTO 10
        ENDIF
        CALL FPSUB(MSUM,MSUM1,MTEMP)
        CALL FPABS(MTEMP,MTEMP)
        CALL FPABS(MSUM,MSUMA)
        CALL FPMPY(MSUMA,MTOL,MSUMA)
        IF (FPCOMP(MTEMP,'LT',MSUMA)) GOTO 30
        CALL FPEQU(MSUM,MSUM1,NDIG,NDIG)
 10   CONTINUE
 30   CONTINUE
      CALL FPMPY(MX,MX,MTEMP)
      CALL FPEXP(MTEMP,MTEMP1)
C      CALL FPMPY(MTEMP,MX,MTEMP)
      CALL FPMPY(MTEMP1,MTEMP,MTEMP)
      CALL FPMPY(MTEMP,MSUM,MGOFX)
      RETURN
      END
      SUBROUTINE MQUAD8(MA,MB,MQUAD)
```

```
      DOUBLE PRECISION DQUAD
      DIMENSION MA(200),MB(200),MAPB(200),MBMA(200),MTEMP1(200),
     &MTEMP2(200),MTEMP3(200),MTEMP4(200),MTEMP5(200),MTEMP6(200),
     &MTEMP7(200),MTEMP8(200),MSUM(200),MFUN(200),MQUAD(200)
      DIMENSION IX1(4),IX2(4),IX3(4),IX4(4),IW1(4),IW2(4),IW3(4),IW4(4)
C
      COMMON /CINIT/INIT
      COMMON /CWORK/MPI(200),MRTPI(200),MONE(200),MTWO(200),MTRTPI(200),
     *MZERO(200),MTOL(200),MDH(200),MNINE(200),NCDIG
      COMMON /GAUSS/ MX1(200),MX2(200),MX3(200),MX4(200),MW1(200),
     &MW2(200),MW3(200),MW4(200)

C
C     GAUSS ABSCISSAS AND WEIGHTS IN MP
      DATA (IX1(J),J=1,4)/0,18343464,24956498,05000000/
      DATA (IX2(J),J=1,4)/0,52553240,99163289,86000000/
      DATA (IX3(J),J=1,4)/0,79666647,74136267,40000000/
      DATA (IX4(J),J=1,4)/0,96028985,64975362,32000000/
      DATA (IW1(J),J=1,4)/0,36268378,33783619,83000000/
      DATA (IW2(J),J=1,4)/0,31370664,58778872,87000000/
      DATA (IW3(J),J=1,4)/0,22238103,44533744,71000000/
      DATA (IW4(J),J=1,4)/0,10122853,62903762,59000000/
C
      NDIG=NCDIG
      IF (INIT.EQ.0) THEN
        CALL FPEQU(MZERO,MX1,NDIG,NDIG)
        MX1(1)=IX1(1)
        MX1(2)=IX1(2)
        MX1(3)=IX1(3)
        MX1(4)=IX1(4)
        CALL FPEQU(MZERO,MX2,NDIG,NDIG)
        MX2(1)=IX2(1)
        MX2(2)=IX2(2)
        MX2(3)=IX2(3)
        MX2(4)=IX2(4)
        CALL FPEQU(MZERO,MX3,NDIG,NDIG)
        MX3(1)=IX3(1)
        MX3(2)=IX3(2)
        MX3(3)=IX3(3)
        MX3(4)=IX3(4)
        CALL FPEQU(MZERO,MX4,NDIG,NDIG)
        MX4(1)=IX4(1)
        MX4(2)=IX4(2)
        MX4(3)=IX4(3)
        MX4(4)=IX4(4)
        CALL FPEQU(MZERO,MW1,NDIG,NDIG)
        MW1(1)=IW1(1)
        MW1(2)=IW1(2)
        MW1(3)=IW1(3)
        MW1(4)=IW1(4)
        CALL FPEQU(MZERO,MW2,NDIG,NDIG)
        MW2(1)=IW2(1)
        MW2(2)=IW2(2)
        MW2(3)=IW2(3)
        MW2(4)=IW2(4)
        CALL FPEQU(MZERO,MW3,NDIG,NDIG)
        MW3(1)=IW3(1)
        MW3(2)=IW3(2)
        MW3(3)=IW3(3)
        MW3(4)=IW3(4)
        CALL FPEQU(MZERO,MW4,NDIG,NDIG)
        MW4(1)=IW4(1)
        MW4(2)=IW4(2)
        MW4(3)=IW4(3)
        MW4(4)=IW4(4)
        PRINT *,'GAUSSIAN ABSCISSAS AND WEIGHTS '
        CALL FPPRNT(MX1)
```

```
          CALL FPPRNT(MX2)
          CALL FPPRNT(MX3)
          CALL FPPRNT(MX4)
          CALL FPPRNT(MW1)
          CALL FPPRNT(MW2)
          CALL FPPRNT(MW3)
          CALL FPPRNT(MW4)
          PAUSE
          INIT=1
        ENDIF
        CALL FPADD(MA,MB,MAPB)
        CALL FPDIVI(MAPB,2,MAPB)
        CALL FPSUB(MB,MA,MBMA)
        CALL FPDIVI(MBMA,2,MBMA)
        CALL FPMPY(MX1,MBMA,MTEMP1)
        CALL FPMPY(MX2,MBMA,MTEMP2)
        CALL FPMPY(MX3,MBMA,MTEMP3)
        CALL FPMPY(MX4,MBMA,MTEMP4)
        I=-1
        CALL FPMPYI(MTEMP1,I,MTEMP5)
        CALL FPMPYI(MTEMP2,I,MTEMP6)
        CALL FPMPYI(MTEMP3,I,MTEMP7)
        CALL FPMPYI(MTEMP4,I,MTEMP8)
C
        CALL FPADD(MTEMP1,MAPB,MTEMP1)
        CALL FPADD(MTEMP2,MAPB,MTEMP2)
        CALL FPADD(MTEMP3,MAPB,MTEMP3)
        CALL FPADD(MTEMP4,MAPB,MTEMP4)
        CALL FPADD(MTEMP5,MAPB,MTEMP5)
        CALL FPADD(MTEMP6,MAPB,MTEMP6)
        CALL FPADD(MTEMP7,MAPB,MTEMP7)
        CALL FPADD(MTEMP8,MAPB,MTEMP8)
C
        CALL FPEQU(MZERO,MSUM,NDIG,NDIG)
        CALL MFOFX(MTEMP1,MFUN)
        CALL FPMPY(MW1,MFUN,MFUN)
        CALL FPADD(MSUM,MFUN,MSUM)
        CALL MFOFX(MTEMP2,MFUN)
        CALL FPMPY(MW2,MFUN,MFUN)
        CALL FPADD(MSUM,MFUN,MSUM)
        CALL MFOFX(MTEMP3,MFUN)
        CALL FPMPY(MW3,MFUN,MFUN)
        CALL FPADD(MSUM,MFUN,MSUM)
        CALL MFOFX(MTEMP4,MFUN)
        CALL FPMPY(MW4,MFUN,MFUN)
        CALL FPADD(MSUM,MFUN,MSUM)
        CALL MFOFX(MTEMP5,MFUN)
        CALL FPMPY(MW1,MFUN,MFUN)
        CALL FPADD(MSUM,MFUN,MSUM)
        CALL MFOFX(MTEMP6,MFUN)
        CALL FPMPY(MW2,MFUN,MFUN)
        CALL FPADD(MSUM,MFUN,MSUM)
        CALL MFOFX(MTEMP7,MFUN)
        CALL FPMPY(MW3,MFUN,MFUN)
        CALL FPADD(MSUM,MFUN,MSUM)
        CALL MFOFX(MTEMP8,MFUN)
        CALL FPMPY(MW4,MFUN,MFUN)
        CALL FPADD(MSUM,MFUN,MSUM)
        CALL FPM2DP(MFUN,DQUAD)
C        PRINT*,'MFUN:',DQUAD
        CALL FPMPY(MSUM,MBMA,MQUAD)
C        CALL FPM2DP(MQUAD,DQUAD)
C        PRINT *,'IN MQUAD, ANS=',DQUAD
C        PAUSE
        RETURN
        END
        SUBROUTINE MFOFXA(MT,MFUN)
```

```
      DOUBLE PRECISION DSUM,DMT
      LOGICAL FPCOMP
      DIMENSION MT(200),MT2(200),MTRM(200),MSUM(200),MCI(200),MCNT(200),
     &MTRM1(200),MTRMA(200),MTRMS(200),MTEMP(200),MSUMA(200),MAK(200),
     &MFUN(200),MSUMT(200)
      COMMON /CWORK/MPI(200),MRTPI(200),MONE(200),MTWO(200),MTRTPI(200),
     *MZERO(200),MTOL(200),MDH(200),MNINE(200),NCDIG
C
      NDIG=NCDIG
      CALL FPM2DP(MT,DMT)
      CALL FPMPY(MT,MT,MT2)
      CALL FPEQU(MONE,MTRM,NDIG,NDIG)
      CALL FPEQU(MONE,MSUM,NDIG,NDIG)
      J=-1
      IF(FPCOMP(MT,"LT",MNINE)) THEN
        CALL FPEQU(MONE,MCI,NDIG,NDIG)
        CALL FPEQU(MTWO,MCNT,NDIG,NDIG)
        CALL FPADD(MONE,MCNT,MCNT)
C         CALL FPDIV(MONE,MT,MTRM1)
        DO 10 I=1,1000
          CALL FPMPY(MTRM,MT2,MTRM)
          CALL FPDIV(MTRM,MCI,MTRM)
          CALL FPMPYI(MTRM,J,MTRM)
          CALL FPDIV(MTRM,MCNT,MTRMS)
          CALL FPADD(MSUM,MTRMS,MSUM)
          CALL FPM2DP(MSUM,DSUM)
C           PRINT*,'IN MFOFX, X,SUM ',DMT,DSUM
          IF(I.LT.5) GOTO 5
          CALL FPMPY(MSUM,MTRTPI,MSUMT)
C           CALL FPSUB(MTRM1,MTEMP,MSUMT)
          CALL FPABS(MTRMS,MTRMA)
          CALL FPABS(MSUMT,MSUMA)
          CALL FPMPY(MTOL,MSUMA,MSUMA)
          IF (FPCOMP(MTRMA,'LT',MSUMA)) GOTO 15
  5       CONTINUE
          CALL FPADD(MCI,MONE,MCI)
          CALL FPADD(MCNT,MTWO,MCNT)
  10    CONTINUE
        PRINT*,"DROP THRU SERIES LOOP"
        PAUSE
  15    CONTINUE
      ELSE
        CALL FPEQU(MDH,MAK,NDIG,NDIG)
        DO 20 I=1,25
          CALL FPMPY(MTRM,MAK,MTRM)
          CALL FPDIV(MTRM,MT2,MTRM)
          CALL FPABS(MTRM,MTRMA)
          IF (FPCOMP(MTRMA,'LE',MTOL)) GOTO 25
          CALL FPMPYI(MTRM,J,MTRM)
          CALL FPADD(MSUM,MTRM,MSUM)
          CALL FPADD(MAK,MONE,MAK)
  20    CONTINUE
        PRINT*,"DROP THRU ASYMPTOTOIC SERIES LOOP"
        PAUSE
  25    CONTINUE
        CALL FPMPYI(MT2,J,MAK)
        CALL FPEXP(MAK,MTRM)
        CALL FPMPY(MSUM,MTRM,MSUM)
        CALL FPMPY(MRTPI,MT2,MAK)
        CALL FPDIV(MSUM,MAK,MSUMT)
      ENDIF
      CALL FPEQU(MSUMT,MFUN,NDIG,NDIG)
      RETURN
      END
      SUBROUTINE MFOFX(MT,MFUN)
      DOUBLE PRECISION DSUM,DMT
      LOGICAL FPCOMP
```

```
      DIMENSION MT(200),MT2(200),MTRM(200),MSUM(200),MCI(200),MCNT(200),
     &MTRM1(200),MTRMA(200),MTRMS(200),MTEMP(200),MSUMA(200),MAK(200),
     &MFUN(200),MSUMT(200)
      COMMON /CWORK/MPI(200),MRTPI(200),MONE(200),MTWO(200),MTRTPI(200),
     *MZERO(200),MTOL(200),MDH(200),MNINE(200),NCDIG
C
      NDIG=NCDIG
      CALL FPM2DP(MT,DMT)
      CALL FPMPY(MT,MT,MT2)
      CALL FPEQU(MONE,MTRM,NDIG,NDIG)
      CALL FPEQU(MONE,MSUM,NDIG,NDIG)
      J=-1
      IF(FPCOMP(MT,"LT",MNINE)) THEN
        CALL FPEQU(MONE,MCI,NDIG,NDIG)
        CALL FPEQU(MTWO,MCNT,NDIG,NDIG)
        CALL FPADD(MONE,MCNT,MCNT)
        CALL FPDIV(MONE,MT,MTRM1)
        DO 10 I=1,1000
          CALL FPMPY(MTRM,MT2,MTRM)
          CALL FPDIV(MTRM,MCI,MTRM)
          CALL FPMPYI(MTRM,J,MTRM)
          CALL FPDIV(MTRM,MCNT,MTRMS)
          CALL FPADD(MSUM,MTRMS,MSUM)
          CALL FPM2DP(MSUM,DSUM)
C           PRINT*,'IN MFOFX, X,SUM ',DMT,DSUM
          IF(I.LT.5) GOTO 5
          CALL FPMPY(MSUM,MTRTPI,MTEMP)
C           CALL FPSUB(MTRM1,MTEMP,MSUMT)
          CALL FPMPY(MTEMP,MT,MTEMP)
          CALL FPSUB(MONE,MTEMP,MSUMT)
          CALL FPABS(MTRMS,MTRMA)
          CALL FPABS(MSUMT,MSUMA)
          CALL FPMPY(MTOL,MSUMA,MSUMA)
          IF (FPCOMP(MTRMA,'LT',MSUMA)) GOTO 15
  5       CONTINUE
          CALL FPADD(MCI,MONE,MCI)
          CALL FPADD(MCNT,MTWO,MCNT)
  10    CONTINUE
        PRINT*,"DROP THRU SERIES LOOP"
        PAUSE
  15    CONTINUE
      ELSE
        CALL FPEQU(MDH,MAK,NDIG,NDIG)
        DO 20 I=1,25
          CALL FPMPY(MTRM,MAK,MTRM)
          CALL FPDIV(MTRM,MT2,MTRM)
          CALL FPABS(MTRM,MTRMA)
          IF (FPCOMP(MTRMA,'LE',MTOL)) GOTO 25
          CALL FPMPYI(MTRM,J,MTRM)
          CALL FPADD(MSUM,MTRM,MSUM)
          CALL FPADD(MAK,MONE,MAK)
  20    CONTINUE
        PRINT*,"DROP THRU ASYMPTOTOIC SERIES LOOP"
        PAUSE
  25    CONTINUE
        CALL FPMPYI(MT2,J,MAK)
        CALL FPEXP(MAK,MTRM)
        CALL FPMPY(MSUM,MTRM,MSUM)
C         CALL FPMPY(MRTPI,MT2,MAK)
        CALL FPMPY(MRTPI,MT,MAK)
        CALL FPDIV(MSUM,MAK,MSUMT)
      ENDIF
      CALL FPEQU(MSUMT,MFUN,NDIG,NDIG)
      RETURN
      END
```

```
      SUBROUTINE ZERF(ZR,ZI,KODE,ZANSR,ZANSI,IERR)
C
C     WRITTEN BY D.E. AMOS, FEBRUARY, 2002
C
C     REFERENCE
C         NBS HANDBOOK OF MATHEMATICAL FUNCTIONS, AMS 55, BY
C         M. ABRAMOWITZ AND I.A. STEGUN, DECEMBER,1955, CHAPTER 7.
C
C     ABSTRACT
C                      ***A DOUBLE PRECISION ***
C         A Z PREFIX BEFORE A FUNCTION OR SUBROUTINE INDICATES A
C         DOUBLE PRECISON COMPLEX ROUTINE. DOUBLE PRECISION COMPLEX
C         NUMBERS ARE CARRIED AS ORDERED PAIRS, I.E. Z=(ZR,ZI).
C
C         ZERF COMPUTES THE ERROR FUNCTION, ERF(Z), OR THE SCALED
C         COERROR FUNCTION, EXP(Z**2)*ERFC(Z), FOR Z=CMPLX(X,Y).
C         IF X.LT.0.0, Z IS REPLACED BY -Z AND ALL COMPUTATION IS
C         DONE IN THE RIGHT HALF PLANE, EXCEPT FOR Z INSIDE THE CIRCLE
C         ZABS(Z).LT.RLB. ERF(-Z)=-ERF(Z) AND ERFC(-Z)=2.0-ERFC(Z).
C
C         THE REGIONS FOR COMPUTATION ARE DIVIDED AS FOLLOWS:
C
C           (1) ZABS(Z).LE.RLB. - POWER SERIES, NBS HANDBOOK, P. 297
C           (2) ZABS(Z).GT.RLB AND 0.LE.X.LE.RLB AND CABS(Z).LT.6 -
C               SERIES, NBS HANDBOOK, P. 299
C           (3) X.GT.RLB AND ZABS(Z).LT.6.0 - CONTINUED FRACTION, NBS
C               HANDBOOK, P. 298
C           (4) ZABS(Z).GE.6.- ASYMPTOTIC EXPANSION, NBS HANDBOOK, P.298
C
C               RLB IS SET IN A DATA STATEMENT SO THAT THE ERRORS
C               ON EACH SIDE OF RLB ARE ABOUT THE SAME (RLB=1.50D0).
C
C         THE SERIES IN (1) IS A CONFLUENT HYPERGEOMETRIC POWER SERIES
C         WHILE THE SERIES IN (3) IS TAKEN FROM MATH COMP,5, 67-70(1951)
C
C         THE WORKING TOLERANCE IS MAX(UNIT ROUNDOFF, 1.0E-14)
C
C         SEVERAL COMMON FUNCTIONS ARE SPECIAL CASES --
C
C         DAWSON-S INTEGRAL
C
C               F(Y)=-I*SQRT(PI)*EXP(-Y*Y)*ERF(I*Y)/2.
C
C         FRESNEL INTEGRALS
C
C               C(Z)+I*S(Z)=(0.5+0.5*I)*ERF(SQRT(PI)*(1.-I)*Z/2.)
C
C         PLASMA DISPERSION FUNCTION
C
C               W(Z)= EXP((-I*Z)**2)*ERFC(-I*Z)
C                   = EXP( - Z*Z   )*ERFC(-I*Z)
C
C     DESCRIPTION OF ARGUMENTS
C         INPUT
C           ZR     - REAL PART OF COMPLEX ARGUMENT Z
C           ZI     - IMAGINARY PART OF COMPLEX ARGUMENT Z
C           KODE   - A SELECTION PARAMETER
C                    KODE=1 RETURNS VALUES FOR ERF(Z)
C                    KODE=2 RETURNS VALUES FOR EXP(Z**2)*ERFC(Z) WHERE
C                           ERFC(Z)=1.0-ERF(Z)
C
C         OUTPUT
C           ZANSR  - REAL PART OF A COMPLEX NUMBER FOR ERF(Z) OR
C                    EXP(Z**2)*ERFC(Z) DEPENDING ON KODE
C           ZANSI  - IMAGINARY PART OF A COMPLEX NUMBER FOR ERF(Z) OR
C                    EXP(Z**2)*ERFC(Z) DEPENDING ON KODE
C
```

```
C     ERROR CONDITIONS
C         IERR=0 NORMAL RETURN, COMPUTATION COMPLETED.
C         IERR=1 KODE NOT 1 OR 2, NO RESULT,
C                ZANS=(0.0D0,0.0D0) RETURNED.
C         IERR=2 OVERFLOW, NO RESULT, ZANS=(0.0D0,0.0D0) RETURNED.
C
C     USES SUBROUTINES I1MACH,R1MACH,ZABS,ZMLT,ZDIV
C-----------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C     FNORM=2/SQRT(PI), GNORM=1/SQRT(PI)
      DATA FNORM,GNORM /1.12837916709551D0,  0.564189583547756D0/
C     EH=DEXP(-0.50D0), EF=DEXP(-0.25D0)
      DATA EH,EF       /0.606530659712633D0, 0.778800783071405D0/
      DATA PI          /3.14159265358979D0/
      DATA RLB         /1.50D0/
      IERR=0
      IF(KODE.LT.1.OR.KODE.GT.2) THEN
        IERR=1
        ZANSR=0.0D0
        ZANSI=0.0D0
        RETURN
      ENDIF
      TOL=MAX(D1MACH(4),1.0D-15)
      KK=MIN(IABS(I1MACH(15)),IABS(I1MACH(16)))
      ELIM=2.303D0*DBLE((FLOAT(KK))*D1MACH(5)-3.0D0)
C-----------------------------------------------------------------------
C     SERIES CEXP(Z*Z)*ERF(Z) FOR ZABS(Z).LT.RLB
C-----------------------------------------------------------------------
      AZ=ZABS(ZR,ZI)
      IF(AZ.LT.DSQRT(10.0D0*D1MACH(1))) THEN
        CZR=0.0D0
        CZI=0.0D0
      ELSE
        CALL ZMLT(ZR,ZI,ZR,ZI,CZR,CZI)
      ENDIF
      IF(AZ.GE.RLB) GO TO 20
      SUMR=1.0D0
      SUMI=0.0D0
      TERMR=SUMR
      TERMI=SUMI
      AK=1.5D0
      RTOL=TOL/10.0D0
      DO 5 I=1,100
        CALL ZMLT(TERMR,TERMI,CZR,CZI,TEMPR,TEMPI)
        TERMR=TEMPR/AK
        TERMI=TEMPI/AK
        SUMR=SUMR+TERMR
        SUMI=SUMI+TERMI
        AT=ZABS(TERMR,TERMI)
        ASUM=ZABS(SUMR,SUMI)
        IF(AT.LE.RTOL*ASUM) GOTO 10
        AK=AK+1.0D0
    5 CONTINUE
   10 CONTINUE
      CZR=-CZR
      CZI=-CZI
      CALL ZMLT(SUMR,SUMI,ZR,ZI,ANSR,ANSI)
      SUMR=FNORM*ANSR
      SUMI=FNORM*ANSI
      TEMPR=DEXP(CZR)
      CZR=TEMPR*DCOS(CZI)
      CZI=TEMPR*DSIN(CZI)
      IF(KODE.EQ.1) THEN
        CALL ZMLT(SUMR,SUMI,CZR,CZI,ZANSR,ZANSI)
      ELSE
        TEMPR=1.0D0
        TEMPI=0.0D0
```

```
        CALL ZDIV(TEMPR,TEMPI,CZR,CZI,CZR,CZI)
        ZANSR=CZR-SUMR
        ZANSI=CZI-SUMI
      ENDIF
      RETURN
   20 CONTINUE
C-----------------------------------------------------------------------
C     COMPUTE IN THE RIGHT HALF PLANE FOR (2),(3) & (4) IN THE PROLOGUE
C-----------------------------------------------------------------------
      ZPR=ZR
      ZPI=ZI
      X=ZR
      Y=ZI
      IF(X.LT.0.0D0) THEN
        ZPR=-ZR
        ZPI=-ZI
      ENDIF
      CALL ZMLT(ZPR,ZPI,ZPR,ZPI,CZR,CZI)
      IF(AZ.GT.6.0D0) GOTO 200
      XP=ZPR
      YP=ZPI
      IF(XP.LE.RLB) GOTO 150
C-----------------------------------------------------------------------
C     CONTINUED FRACTION FOR ERFC(Z), X.GT.RLB AND CABS(Z).LE.6.0.
C     N=INT(75.0/AZ+5.0) FOR TOL=MAX(UNIT ROUND OFF,1.E-15)
C-----------------------------------------------------------------------
      N=INT(SNGL((75.0D0/AZ+5.0D0)))
      FN=DBLE(FLOAT(N))
      TERMR=CZR
      TERMI=CZI
      DO 30 I=1,N
        FNH=FN-0.5D0
        TEMPR=FN+TERMR
        TEMPI=TERMI
        ANSR=FNH*TERMR
        ANSI=FNH*TERMI
        CALL ZDIV(ANSR,ANSI,TEMPR,TEMPI,TEMPR,TEMPI)
        TERMR=CZR+TEMPR
        TERMI=CZI+TEMPI
        FN=FN-1.0D0
   30 CONTINUE
      IF(KODE.EQ.2) GO TO 35
      CZR=-CZR
      CZI=-CZI
      IF(DABS(CZR).GT.ELIM) THEN
        IF(CZR.GT.0.0D0) THEN
          IERR=2
          ZANSR=0.0D0
          ZANSI=0.0D0
          RETURN
        ELSE
          RER=0.0D0
        ENDIF
      ELSE
        RER=DEXP(CZR)
      ENDIF
      CZR=RER*DCOS(CZI)
      CZI=RER*DSIN(CZI)
      TEMPR=GNORM*ZPR
      TEMPI=GNORM*ZPI
      CALL ZDIV(TEMPR,TEMPI,TERMR,TERMI,SUMR,SUMI)
      GOTO 203
   35 CONTINUE
      TEMPR=GNORM*ZPR
      TEMPI=GNORM*ZPI
      CALL ZDIV(TEMPR,TEMPI,TERMR,TERMI,ZANSR,ZANSI)
      IF(X.GE.0.0D0) RETURN
```

```
      IF(DABS(CZR).GT.ELIM) THEN
        IF(CZR.GT.0.0D0) THEN
          IERR=2
          ZANSR=0.0D0
          ZANSI=0.0D0
          RETURN
        ELSE
          RER=0.0D0
        ENDIF
      ELSE
        RER=DEXP(CZR)
      ENDIF
      CEIR=DCOS(CZI)
      CEII=DSIN(CZI)
      CZR=RER*CEIR
      CZI=RER*CEII
      ZANSR=CZR+CZR-ZANSR
      ZANSI=CZI+CZI-ZANSI
      RETURN
  150 CONTINUE
C-----------------------------------------------------------------------
C    SERIES (2) FOR ZABS(Z).GE.RLB AND 0.0.LE.XP.LE.RLB AND ZABS(Z).LT.6
C-----------------------------------------------------------------------
      S1=0.0D0
      S2=0.0D0
      IF(DABS(XP).LT.SQRT(D1MACH(1))) THEN
        X2=0.0D0
      ELSE
        X2=XP*XP
      ENDIF
      AY=DABS(YP)
      IF(AY.NE.0.0D0) THEN
C       TOL IS DECREASED BY 100 TO BE SLIGHTLY CONSERVATIVE IN DECIDING
C       WHEN EXP(-2*N*Y) IS SMALL COMPARED TO 1.0D0
        ETOL=DLOG(TOL/100.0D0)
        FKK=MIN(100.0D0,-0.5D0*ETOL/AY+2.0D0)
        KK=INT(FKK)
        FKK=MIN(100.0D0,ELIM/AY+2.0D0)
        KKP=INT(FKK)
      ELSE
        SXY=0.0D0
        CXY=1.0D0
        EX=DEXP(-X2)
        GOTO 96
      ENDIF
      FX2=4.0D0*X2
      TX=XP+XP
      XY=XP*AY
      TXY=XY+XY
      SXYH=DSIN(XY)
      SXY=DSIN(TXY)
      CXY=DCOS(TXY)
      FN=1.0D0
      FNH=0.5D0
      EY=DEXP(AY)
      REY=1.0D0/EY
      REY2=REY*REY
      REY2N=REY2
      TXM=TX*REY
      EHN=EH
      UN=EF*EY
      VN=1.0D0
      RTOL=TOL/10.0D0
      DO 90 I=1,100
        CSH=1.0D0+REY2N
        TM=XP*CSH
        SSH=1.0D0-REY2N
```

```
        TMP=FNH*SSH
        RN=TXM-TM*CXY+TMP*SXY
        AIN=TM*SXY+TMP*CXY
        CF=UN/(VN+FX2)
        RN=CF*RN
        AIN=CF*AIN
        S1=S1+RN
        S2=S2+AIN
        AT=DABS(RN)+DABS(AIN)
        AS=DABS(S1)+DABS(S2)
        IF(AT.LE.RTOL*AS) GO TO 95
        UN=UN*EHN*EF*EY
        EHN=EHN*EH
        IF (I.GE.KK) THEN
          REY2N=0.0D0
        ELSE
          REY2N=REY2N*REY2
        ENDIF
        IF(I.GE.KKP) THEN
          TXM=0.0D0
        ELSE
          TXM=TXM*REY
        ENDIF
        VN=VN+FN+FN+1.0D0
        FNH=FNH+0.5D0
        FN=FN+1.0D0
   90 CONTINUE
   95 S1=S1+S1
      S2=S2+S2
      IF(X.NE.0.0D0) THEN
        S1=S1+SXYH*SXYH/XP
        S2=S2+SXY/TX
      ELSE
        S2=S2+AY
      ENDIF
C-----------------------------------------------------------------------
C     POWER SERIES FOR DEXP(XP*XP)*DERF(XP), 0.LE.XP.LE.RLB
C-----------------------------------------------------------------------
   96 CONTINUE
      W=1.0D0
      AK=1.5D0
      TM=1.0D0
      RTOL=TOL/10.0D0
      DO 98 I=1,100
        TM=TM*X2/AK
        W=W+TM
        IF(DABS(TM).LE.DABS(W)*RTOL) GO TO 99
        AK=AK+1.0D0
   98 CONTINUE
   99 CONTINUE
      EX=DEXP(-X2)
      W=W*XP*FNORM
      IF(KODE.EQ.2) GO TO 100
      CF=EX/PI
      S1=S1*CF+W*EX
      S2=S2*CF
      IF(YP.LT.0.0D0) S2=-S2
      ZANSR=S1
      ZANSI=S2
      IF(X.LT.0.0D0) THEN
        ZANSR=-ZANSR
        ZANSI=-ZANSI
      ENDIF
      RETURN
  100 CONTINUE
      IF((X.NE.0.0D0).AND.(Y.NE.0.0D0)) THEN
        SXY=SXY*DSIGN(1.0D0,Y)*DSIGN(1.0D0,X)
```

```
      ENDIF
      RCZR=CXY
      RCZI=SXY
      Y2=YP*YP
      EY=DEXP(-Y2)
      RER=EY/EX
      CZR=RER*RCZR
      CZI=RER*RCZI
      S1=EY*(W+S1/PI)
      S2=EY*S2/PI
      IF(YP.LT.0.0D0) S2=-S2
      IF(X.LT.0.0D0) THEN
        S1=-S1
        S2=-S2
      ENDIF
      CALL ZMLT(RCZR,RCZI,S1,S2,TEMPR,TEMPI)
      ZANSR=CZR-TEMPR
      ZANSI=CZI-TEMPI
      RETURN
  200 CONTINUE
C-----------------------------------------------------------------------
C     ASYMPTOTIC EXPANSION FOR CABS(Z).GT.6
C-----------------------------------------------------------------------
      TEMPR=0.5D0
      TEMPI=0.0D0
      CALL ZDIV(TEMPR,TEMPI,CZR,CZI,RCZR,RCZI)
      SUMR=1.0D0
      SUMI=0.0D0
      TERMR=SUMR
      TERMI=SUMI
      AK=1.0D0
      DO 201 I=1,35
        CALL ZMLT(TERMR,TERMI,RCZR,RCZI,TEMPR,TEMPI)
        TERMR=-AK*TEMPR
        TERMI=-AK*TEMPI
        SUMR=SUMR+TERMR
        SUMI=SUMI+TERMI
        AT=ZABS(TERMR,TERMI)
        AS=ZABS(SUMR,SUMI)
        IF(AT.LE.TOL*AS) GO TO 202
        AK=AK+2.0D0
  201 CONTINUE
      PRINT *,"DROP THRU ASYMPTOTIC EXPANSION LOOP-- AN ERROR FOR"
      PRINT *,"Z = ",ZR,ZI, " AND KODE = ",KODE
  202 CONTINUE
      ANSR=SUMR*GNORM
      ANSI=SUMI*GNORM
      CALL ZDIV(ANSR,ANSI,ZPR,ZPI,SUMR,SUMI)
      IF(KODE.EQ.2) GO TO 206
      CZR=-CZR
      CZI=-CZI
      IF(DABS(CZR).GT.ELIM) THEN
        IF(CZR.GT.0.0D0) THEN
          IERR=2
          ZANSR=0.0D0
          ZANSI=0.0D0
          RETURN
        ELSE
          RER=0.0D0
        ENDIF
      ELSE
        RER=DEXP(CZR)
      ENDIF
      CZR=RER*DCOS(CZI)
      CZI=RER*DSIN(CZI)
C-----------------------------------------------------------------------
C     ERF(Z)=1.0-ERFC(Z),
```

```
C     FOR ABS(Z) NEAR 6 ERFC(Z)=O(EXP(-Z*Z)=EXP(Y*Y-X*X)) AND ERFC IS
C     SUBTRACTED FROM 1.0 ONLY IF ERFC AND 1.0-ERFC HAVE DIFFERENT
C     MAGNITUDES. THIS PRESERVES THE ODDNESS OF ERF WHEN Z IS CLOSE TO
C     I*Y WHERE ERF(Z) IS PURELY IMAGINARY. (Y*Y-X*X IS RELATIVELY LARGE
C     FOR Y>X, I.E. EXP(36) IS O(10**(16)) AND 1.0-EXP(36)=EXP(36) TO
C     MACHINE PRECISION.) ABS((1-U)*(1-U)+V*V)=U*U+V*V+1-2*U AND 1-2U IS
C     THE CHANGE IN MAGNITUDE SQUARED. THEREFORE ABS(1-2*U)/(U*U+V*V)
C     MUST BE LESS THAN UNIT ROUNDOFF TO SEE NO CHANGE IN MAGNITUDE
C     WHERE U=REAL(ERFC(Z)) AND V=AIMAG(ERFC(Z)).
C-----------------------------------------------------------------------
  203 CONTINUE
      CALL ZMLT(-CZR,-CZI,SUMR,SUMI,TERMR,TERMI)
      ATERM=ZABS(TERMR,TERMI)
      DEL=DABS(1.0D0-TERMR-TERMR)
      RTOL=DSQRT(TOL)
      IF(SQRT(DEL).LT.ATERM*RTOL) THEN
        ZANSR=TERMR
        ZANSI=TERMI
      ELSE
        ZANSR=1.0D0+TERMR
        ZANSI=TERMI
      ENDIF
      IF(X.LT.0.0D0) THEN
        ZANSR=-ZANSR
        ZANSI=-ZANSI
      ENDIF
      RETURN
  206 CONTINUE
C-----------------------------------------------------------------------
C     CEXP((-Z)**2)*ERFC(-Z)=2.0*CEXP(Z*Z)-CEXP(Z*Z)*ERFC(Z)
C-----------------------------------------------------------------------
      IF(X.LT.0.0D0)THEN
        IF(DABS(CZR).GT.ELIM) THEN
          IF(CZR.GT.0.0D0) THEN
            IERR=2
            ZANSR=0.0D0
            ZANSI=0.0D0
            RETURN
          ELSE
            RER=0.0D0
          ENDIF
        ELSE
          RER=DEXP(CZR)
        ENDIF
        CEIR=DCOS(CZI)
        CEII=DSIN(CZI)
        CZR=RER*CEIR
        CZI=RER*CEII
        SUMR=CZR+CZR-SUMR
        SUMI=CZI+CZI-SUMI
      ENDIF
      ZANSR=SUMR
      ZANSI=SUMI
      RETURN
      END
      SUBROUTINE ZMLT(AR, AI, BR, BI, CR, CI)
C***BEGIN PROLOGUE  ZMLT
C
C     DOUBLE PRECISION COMPLEX MULTIPLY, C=A*B.
C
C***ROUTINES CALLED  (NONE)
C***END PROLOGUE  ZMLT
      DOUBLE PRECISION AR, AI, BR, BI, CR, CI, CA, CB
      CA = AR*BR - AI*BI
      CB = AR*BI + AI*BR
      CR = CA
      CI = CB
```

```
      RETURN
      END
      SUBROUTINE ZDIV(AR, AI, BR, BI, CR, CI)
C***BEGIN PROLOGUE  ZDIV
C
C     DOUBLE PRECISION COMPLEX DIVIDE C=A/B.
C
C***ROUTINES CALLED  ZABS
C***END PROLOGUE  ZDIV
      DOUBLE PRECISION AR, AI, BR, BI, CR, CI, BM, CA, CB, CC, CD
      DOUBLE PRECISION ZABS
      BM = 1.0D0/ZABS(BR,BI)
      CC = BR*BM
      CD = BI*BM
      CA = (AR*CC+AI*CD)*BM
      CB = (AI*CC-AR*CD)*BM
      CR = CA
      CI = CB
      RETURN
      END
      SUBROUTINE ZSQRT(AR, AI, BR, BI)
C***BEGIN PROLOGUE  ZSQRT
C
C     DOUBLE PRECISION COMPLEX SQUARE ROOT, B=CSQRT(A)
C
C***ROUTINES CALLED  ZABS
C***END PROLOGUE  ZSQRT
      DOUBLE PRECISION AR, AI, BR, BI, ZM, DTHETA, DPI, DRT
      DOUBLE PRECISION ZABS
      DATA DRT , DPI / 7.071067811865475244008443621D-1,
     1                 3.141592653589793238462643383D+0/
      ZM = ZABS(AR,AI)
      ZM = DSQRT(ZM)
      IF (AR.EQ.0.0D+0) GO TO 10
      IF (AI.EQ.0.0D+0) GO TO 20
      DTHETA = DATAN(AI/AR)
      IF (DTHETA.LE.0.0D+0) GO TO 40
      IF (AR.LT.0.0D+0) DTHETA = DTHETA - DPI
      GO TO 50
   10 IF (AI.GT.0.0D+0) GO TO 60
      IF (AI.LT.0.0D+0) GO TO 70
      BR = 0.0D+0
      BI = 0.0D+0
      RETURN
   20 IF (AR.GT.0.0D+0) GO TO 30
      BR = 0.0D+0
      BI = DSQRT(DABS(AR))
      RETURN
   30 BR = DSQRT(AR)
      BI = 0.0D+0
      RETURN
   40 IF (AR.LT.0.0D+0) DTHETA = DTHETA + DPI
   50 DTHETA = DTHETA*0.5D+0
      BR = ZM*DCOS(DTHETA)
      BI = ZM*DSIN(DTHETA)
      RETURN
   60 BR = ZM*DRT
      BI = ZM*DRT
      RETURN
   70 BR = ZM*DRT
      BI = -ZM*DRT
      RETURN
      END
      SUBROUTINE ZEXP(AR, AI, BR, BI)
C***BEGIN PROLOGUE  ZEXP
C
C     DOUBLE PRECISION COMPLEX EXPONENTIAL FUNCTION B=EXP(A)
```

```
C
C***ROUTINES CALLED  (NONE)
C***END PROLOGUE  ZEXP
      DOUBLE PRECISION AR, AI, BR, BI, ZM, CA, CB
      ZM = DEXP(AR)
      CA = ZM*DCOS(AI)
      CB = ZM*DSIN(AI)
      BR = CA
      BI = CB
      RETURN
      END
      SUBROUTINE ZLOG(AR, AI, BR, BI, IERR)
C***BEGIN PROLOGUE  ZLOG
C
C     DOUBLE PRECISION COMPLEX LOGARITHM B=CLOG(A)
C     IERR=0,NORMAL RETURN      IERR=1, Z=CMPLX(0.0,0.0)
C
C***ROUTINES CALLED  ZABS
C***END PROLOGUE  ZLOG
      DOUBLE PRECISION AR, AI, BR, BI, ZM, DTHETA, DPI, DHPI
      DOUBLE PRECISION ZABS
      DATA DPI , DHPI  / 3.141592653589793238462643383D+0,
     1                   1.570796326794896619231321696D+0/
C
      IERR=0
      IF (AR.EQ.0.0D+0) GO TO 10
      IF (AI.EQ.0.0D+0) GO TO 20
      DTHETA = DATAN(AI/AR)
      IF (DTHETA.LE.0.0D+0) GO TO 40
      IF (AR.LT.0.0D+0) DTHETA = DTHETA - DPI
      GO TO 50
   10 IF (AI.EQ.0.0D+0) GO TO 60
      BI = DHPI
      BR = DLOG(DABS(AI))
      IF (AI.LT.0.0D+0) BI = -BI
      RETURN
   20 IF (AR.GT.0.0D+0) GO TO 30
      BR = DLOG(DABS(AR))
      BI = DPI
      RETURN
   30 BR = DLOG(AR)
      BI = 0.0D+0
      RETURN
   40 IF (AR.LT.0.0D+0) DTHETA = DTHETA + DPI
   50 ZM = ZABS(AR,AI)
      BR = DLOG(ZM)
      BI = DTHETA
      RETURN
   60 CONTINUE
      IERR=1
      RETURN
      END
      DOUBLE PRECISION FUNCTION ZABS(ZR, ZI)
C***BEGIN PROLOGUE  ZABS
C
C     ZABS COMPUTES THE ABSOLUTE VALUE OR MAGNITUDE OF A DOUBLE
C     PRECISION COMPLEX VARIABLE CMPLX(ZR,ZI)
C
C***ROUTINES CALLED  (NONE)
C***END PROLOGUE  ZABS
      DOUBLE PRECISION ZR, ZI, U, V, Q, S
      U = DABS(ZR)
      V = DABS(ZI)
      S = U + V
C-----------------------------------------------------------------------
C     S*1.0D0 MAKES AN UNNORMALIZED UNDERFLOW ON CDC MACHINES INTO A
C     TRUE FLOATING ZERO
```

```
C-----------------------------------------------------------------------
      S = S*1.0D+0
      IF (S.EQ.0.0D+0) GO TO 20
      IF (U.GT.V) GO TO 10
      Q = U/V
      ZABS = V*DSQRT(1.D+0+Q*Q)
      RETURN
   10 Q = V/U
      ZABS = U*DSQRT(1.D+0+Q*Q)
      RETURN
   20 ZABS = 0.0D+0
      RETURN
      END
```

```
      PROGRAM QCZERF
C-----------------------------------------------------------------------
C     PROGRAM TO TEST SUBROUTINE CERF FOR THE COMPLEX ERROR FUNCTION
C     AGAINST THE NBS HANDBOOK (A&S) VALUES (CHAPTER 7)
C-----------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PI=4.0D0*DATAN(1.0D0)
      HRPI=0.5D0*DSQRT(PI)
    5 CONTINUE
      PRINT *,' '
      PRINT *,'CASES 3 THRU 6 REPRODUCE TABLES IN A&S CHAPTER 7'
      PRINT *,' '
      PRINT *,'ICASE=1: CHECKS ERF(X)+ERFC(X)-1=0 FOR X.GE.0'
      PRINT *,'ICASE=2: CHECKS ODDNESS OF ERF(Z): ERF(Z)+ERF(-Z)=0'
      PRINT *,'ICASE=3: DAWSON INTEGRAL=ERF(I*Y)/I TABLE 7.5, Y.LE.2'
      PRINT *,'ICASE=4: DAWSON INTEGRAL=ERF(I*Y)/I TABLE 7.5, Y.GE.2'
      PRINT *,'ICASE=5: FRESNEL INTEGRALS C(X), S(X) TABLE 7.7'
      PRINT *,'ICASE=6: W(Z)=CEXP(-Z*Z)*ERFC(-I*Z), TABLE 7.9'
      PRINT *,'ICASE=7: CHECKS W(Z) AGAINST A DP QUADRATURE'
      PRINT *,'ICASE=8: CHECKS CONJG(W(Z))-W(CONJG(Z))=0 FOR X.LT.0'
      PRINT *,'ICASE=9: EXIT PROGRAM'
      PRINT *,' '
      PRINT *,'INPUT THE CASE (1 TO 9): '
      READ *,ICASE
      IF (ICASE.LE.0) THEN
        PRINT *,'ICASE MUST BE BETWEEN 1 AND 9 INCLUSIVELY'
        GO TO 5
      ENDIF
      IF(ICASE.GT.8) GOTO 6
      KOUNT=0
      IF(ICASE.EQ.1) THEN
C---------------------------------------------------
C     ICASE=1     ERF(X)+ERFC(X)=1, X.GT.0.0 ,Y=0.0
C---------------------------------------------------
        KODE=1
        KKODE=2
        JL=1
        IL=88
        PRINT *, "            CHECK IDENTITY   ERF(X)+ERFC(X)=1"
        PRINT *, "              Z                         RESULT"
        DO 20 J=1,JL
          Y=DBLE(FLOAT(J-1))/10.0D0
        DO 10 I=1,IL
          X=DBLE(FLOAT(I-1))/10.0D0
          ZR=X
          ZI=Y
          ZTR=X
          ZTI=0.0D0
          CALL ZERF(ZTR,ZTI,KKODE,CY5R,CY5I,IERR)
          EX=DEXP(-X*X)
          CY5R=CY5R*EX
          CY5I=CY5I*EX
          CALL ZERF(ZTR,ZTI,KODE,CYR,CYI,IERR)
          CYR=CYR+CY5R-1.0D0
          CYI=CYI+CY5I
          KOUNT=KOUNT+1
          IF(MOD(KOUNT,20).EQ.0) THEN
            PAUSE
            PRINT *, "            CHECK IDENTITY   ERF(X)+ERFC(X)=1"
            PRINT *,"              Z                          RESULT"
          ENDIF
          PRINT 900,ZR,ZI,CYR,CYI
  900     FORMAT(4D14.6)
   10   CONTINUE
   20   CONTINUE
      ENDIF
      IF (ICASE.EQ.2) THEN
```

```
C--------------------------------------------
C     ICASE=2     CHECK FOR ODDNESS OF ERF(Z)
C--------------------------------------------
        KODE=1
        JL=15
        IL=15
        PRINT *,"     ODDNESS CHECK FOR COMPLEX ERROR FUNCTION"
        PRINT *,"              Z                        RESULT"
        DO 130 J=1,JL
          Y=DBLE(FLOAT(J-1))/2.0D0
        DO 140 I=1,IL
          X=DBLE(FLOAT(I-1))/2.0D0
          ZR=X
          ZI=Y
          CALL ZERF(ZR,ZI,KODE,CY7R,CY7I,IERR)
          ZTR=-ZR
          ZTI=-ZI
          CALL ZERF(ZTR,ZTI,KODE,CYR,CYI,IERR)
          CYR=CYR+CY7R
          CYI=CYI+CY7I
          XX=ZABS(CY7R,CY7I)
          IF(XX.NE.0.0D0) THEN
            CYR=CYR/XX
            CYI=CYI/XX
          ENDIF
          KOUNT=KOUNT+1
          IF(MOD(KOUNT,20).EQ.0) THEN
            PAUSE
            PRINT *,"     ODDNESS CHECK FOR COMPLEX ERROR FUNCTION"
            PRINT *,"              Z                        RESULT"
          ENDIF
          PRINT 900,ZR,ZI,CYR,CYI
  140   CONTINUE
  130   CONTINUE
      ENDIF
      IF(ICASE.EQ.3) THEN
C--------------------------------------------------
C     ICASE=3     DAWSON'S INTEGRAL=F(Y)=ERF(I*Y)/I
C--------------------------------------------------
        KODE=1
        JL=36
        IL=1
        PRINT *,"       DAWSON'S INTEGRAL, Y.LE.2.0, TABLE 7.5"
        PRINT *,"              Z                        RESULT"
        DO 30 J=1,JL
          Y=DBLE(FLOAT(J-1))/5.0D0
        DO 40 I=1,IL
          X=DBLE(FLOAT(I-1))/10.0D0
          ZR=X
          ZI=Y
          COEF=HRPI*DEXP(-Y*Y)
          ZTR=0.0D0
          ZTI=Y
          CALL ZERF(ZTR,ZTI,KODE,CYR,CYI,IERR)
          CYR=COEF*CYR
          CYI=COEF*CYI
          KOUNT=KOUNT+1
          IF(MOD(KOUNT,20).EQ.0) THEN
            PAUSE
            PRINT *,"       DAWSON'S INTEGRAL, Y.LE.2.0, TABLE 7.5"
            PRINT *,"              Z                        RESULT"
          ENDIF
          PRINT 900,ZR,ZI,CYR,CYI
  40   CONTINUE
  30   CONTINUE
      ENDIF
      IF(ICASE.EQ.4) THEN
```

```
C-----------------------------------------------------------
C     ICASE=4     DAWSON'S INTEGRAL=F(Y)=ERF(I*Y)/I,Y.GT.2
C-----------------------------------------------------------
        KODE=1
        JL=48
        IL=1
        PRINT *,"      DAWSON'S INTEGRAL, Y.GE.2.0, TABLE 7.5"
        PRINT *,"                 1/(Y*Y)              RESULT"
        DO 50 J=1,JL
          Y=DBLE(FLOAT(J-1))/5.0D0
        DO 60 I=1,IL
          X=DBLE(FLOAT(I-1))/10.0D0
          R=0.250D0-DBLE(FLOAT(J-1))*0.005D0
          Y=1.0D0/DSQRT(R)
          COEF=HRPI*Y*DEXP(-Y*Y)
          ZTR=0.0D0
          ZTI=Y
          CALL ZERF(ZTR,ZTI,KODE,CYR,CYI,IERR)
          CYR=COEF*CYR
          CYI=COEF*CYI
          ZR=0.0D0
          ZI=R
          KOUNT=KOUNT+1
          IF(MOD(KOUNT,20).EQ.0) THEN
            PAUSE
            PRINT *,"      DAWSON'S INTEGRAL, Y.GE.2.0, TABLE 7.5"
            PRINT *,"                 1/(Y*Y)              RESULT"
          ENDIF
          PRINT 900,ZR,ZI,CYR,CYI
   60   CONTINUE
   50   CONTINUE
      ENDIF
      IF(ICASE.EQ.5) THEN
C----------------------------------------------
C     ICASE=5     FRESNEL INTEGRALS C(X), S(X)
C----------------------------------------------
        KODE=1
        JL=1
        IL=31
        PRINT *,"           FRESNEL INTEGRALS TABLE 7.7"
        PRINT *,"      X                         C(X)         S(X)"
        DO 70 J=1,JL
          Y=DBLE(FLOAT(J-1))/5.0D0
        DO 80 I=1,IL
          X=DBLE(FLOAT(I-1))/10.0D0
          COEFR=0.5D0
          COEFI=0.5D0
          ZTR=HRPI*X
          ZTI=-HRPI*X
          CALL ZERF(ZTR,ZTI,KODE,CYR,CYI,IERR)
          CALL ZMLT(COEFR,COEFI,CYR,CYI,CYR,CYI)
          ZR=X
          ZI=0.0D0
          KOUNT=KOUNT+1
          IF(MOD(KOUNT,20).EQ.0) THEN
            PAUSE
            PRINT *,"           FRESNEL INTEGRALS TABLE 7.7"
            PRINT *,"      X                         C(X)         S(X)"
          ENDIF
          PRINT 900,ZR,ZI,CYR,CYI
   80   CONTINUE
   70   CONTINUE
      ENDIF
      IF(ICASE.EQ.6) THEN
C------------------------------------------
C     ICASE=6     EXP(-Z*Z)*ERFC(-I*Z)
C------------------------------------------
```

```
        KODE=2
        JL=7
        IL=21
        PRINT *,"              COMPLEX ERROR FUNCTION TABLE 7.9"
        PRINT *,"               Z                        RESULT"
        DO 90 I=1,IL
          X=DBLE(FLOAT(I-1))/5.0D0
        DO 100 J=1,JL
          Y=DBLE(FLOAT(J-1))/2.0D0
          ZR=X
          ZI=Y
          COEF=1.0D0
          ZTR=Y
          ZTI=-X
          CALL ZERF(ZTR,ZTI,KODE,CYR,CYI,IERR)
          KOUNT=KOUNT+1
          IF(MOD(KOUNT,20).EQ.0) THEN
            PAUSE
            PRINT *,"              COMPLEX ERROR FUNCTION TABLE 7.9"
            PRINT *,"               Z                        RESULT"
          ENDIF
          PRINT 900,ZR,ZI,CYR,CYI
  100   CONTINUE
   90   CONTINUE
      ENDIF
      IF(ICASE.EQ.7) THEN
C-------------------------------------------------
C     ICASE=7     CHECK W(Z) AGAINST DP QUADRATURE
C-------------------------------------------------
C       FORMULAE 7.4.13 & 7.4.14 REQUIRE Y.GT.0.0D0. THEREFORE THETA=0
C       AND THETA=PI ARE EXCLUDED
        TPI=PI+PI
        JL=14
        IL=39
        PRINT *,"    W(Z) COMPARED TO DP QUADRATURE OF A&S EQNS. 7.4.13
     && 7.4.14"
        PRINT *,"      R             THETA       REL ERR0R            RES
     &ULT"
        DO 150 J=1,JL
          R=DBLE(FLOAT(J))/2.0D0
        DO 160 I=1,IL
          IF(I.EQ.20) GOTO 160
          THET=TPI*DBLE(FLOAT(I))/40.0D0
          X=R*DCOS(THET)
          Y=R*DSIN(THET)
          ZR=X
          ZI=Y
          KODE=2
          ZTR=Y
          ZTI=-X
          CALL ZERF(ZTR,ZTI,KODE,CYR,CYI,IERR)
          IF(IERR.EQ.2) THEN
            PRINT *, 'OVERFLOW FOR Z = ',ZTR,ZTI
            GOTO 160
          ENDIF
          DX=X
          DY=Y
          CALL ZW(DX,DY,WANSR,WANSI)
          DERRR=CYR-WANSR
          DERRI=CYI-WANSI
C          CERR=CMPLX(SNGL(DERRR),SNGL(DERRI))
          ERR=ZABS(DERRR,DERRI)
C          WANS=CMPLX(SNGL(WANSR),SNGL(WANSI))
          AZ=ZABS(WANSR,WANSI)
          IF(AZ.NE.0.0D0) THEN
            ERR=ERR/AZ
          ENDIF
```

```
          KOUNT=KOUNT+1
          IF(MOD(KOUNT,10).EQ.0) THEN
            PAUSE
            PRINT *,"    W(Z) COMPARED TO DP QUADRATURE OF A&S EQNS. 7.4
     &.13 & 7.4.14"
            PRINT *,"      R            THETA       REL ERR0R
     & RESULT"
          ENDIF
          PRINT 905,R,THET,ERR,WANSR,WANSI
C         PRINT 906,CYR,CYI
  905     FORMAT(5E14.6)
C 906     FORMAT(42X,2E14.6)
  160   CONTINUE
  150   CONTINUE
      ENDIF
      IF (ICASE.EQ.8) THEN
C-------------------------------------------------
C     ICASE=8     CHECK FOR CONJUGACY, NEGATIVE X
C-------------------------------------------------
        KODE=2
        JL=15
        IL=15
        PRINT *,"   CONJUGACY CHECK FOR COMPLEX ERROR FUNCTION"
        PRINT *,"                Z                         RESULT"
        DO 110 J=1,JL
          Y=DBLE(FLOAT(J-1))/2.0D0
        DO 120 I=1,IL
          X=DBLE(FLOAT(I-1))/2.0D0
          ZR=-X
          ZI=-Y
          ZTR=ZR
          ZTI=ZI
          CALL ZERF(ZTR,ZTI,KODE,CY6R,CY6I,IERR)
          CY6I=-CY6I
          ZTI=-ZTI
          CALL ZERF(ZTR,ZTI,KODE,CYR,CYI,IERR)
          CYR=CYR-CY6R
          CYI=CYI-CY6I
          XX=ZABS(CY6R,CY6I)
          IF (XX.NE.0.0) THEN
            CYR=CYI/XX
            CYI=CYI/XX
          ENDIF
          KOUNT=KOUNT+1
          IF(MOD(KOUNT,20).EQ.0) THEN
            PAUSE
            PRINT *,"   CONJUGACY CHECK FOR COMPLEX ERROR FUNCTION"
            PRINT *,"                Z                         RESULT"
          ENDIF
          PRINT 900,ZR,ZI,CYR,CYI
  120   CONTINUE
  110   CONTINUE
      ENDIF
      GO TO 5
    6 CONTINUE
      END
      SUBROUTINE ZW(DX,DY,WANSR,WANSI)
C
C     PLAZMA DISPERSION FUNCTION W(Z)=DEXP(-Z*Z)*ERFC(-I*Z) BY
C     DOUBLE PRECISION QUADRATURE, Z=X+I*Y.
C
C     DX AND DY ARE DOUBLE PRECISION, Z=DX+I*DY
C     WANSR=DOUBLE PRECISION REAL PART OF W(Z)
C     WANSI=DOUBLE PRECISION IMAGINARY PART OF W(Z)
C-----------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      COMMON/CDQWZ/PN,CX,CY,IR
```

```
      DATA PI /3.14159265358979D0/
      EXTERNAL DQWZ
      X=DX
      Y=DY
      IF(DY.LT.0.0D0) THEN
        X=-DX
        Y=-DY
      ENDIF
      CX=X
      CY=Y
      INIT=0
      X1=0.0D0
      SIG=4.0D0
      REL=1.0D-15
      PN=1.0D0
      IR=0
      CALL DQUAD8(DQWZ,INIT,X1,SIG,REL,X2,QANSP,IERR)
      CALL DQUAD8(DQWZ,INIT,X1,SIG,REL,X2,QANSP,IERR)
      INIT=0
      X1=0.0D0
      SIG=4.0D0
      REL=1.0D-15
      PN=-1.0D0
      IR=0
      CALL DQUAD8(DQWZ,INIT,X1,SIG,REL,X2,QANSN,IERR)
      CALL DQUAD8(DQWZ,INIT,X1,SIG,REL,X2,QANSN,IERR)
      WANSR=(QANSP+QANSN)/PI
      INIT=0
      X1=0.0D0
      SIG=4.0D0
      REL=1.0D-15
      PN=1.0D0
      IR=1
      CALL DQUAD8(DQWZ,INIT,X1,SIG,REL,X2,QANSP,IERR)
      CALL DQUAD8(DQWZ,INIT,X1,SIG,REL,X2,QANSP,IERR)
      INIT=0
      X1=0.0D0
      SIG=4.0D0
      REL=1.0D-15
      PN=-1.0D0
      IR=1
      CALL DQUAD8(DQWZ,INIT,X1,SIG,REL,X2,QANSN,IERR)
      CALL DQUAD8(DQWZ,INIT,X1,SIG,REL,X2,QANSN,IERR)
      WANSI=(QANSP+QANSN)/PI
      IF(DY.LT.0.0D0) THEN
        DZ2R=DX*DX-DY*DY
        DEX=DEXP(-DZ2R)
        DTXY=2.0D0*DX*DY
        DSXY=DSIN(DTXY)
        DCXY=DCOS(DTXY)
        DTRMR=DEX*DCXY
        DTRMI=-DEX*DSXY
        WANSR=DTRMR+DTRMR-WANSR
        WANSI=DTRMI+DTRMI-WANSI
      ENDIF
      RETURN
      END
      DOUBLE PRECISION FUNCTION DQWZ(T)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      COMMON/CDQWZ/PN,CX,CY,IR
      A=CX-PN*T
      IF(IR.EQ.0) THEN
        B=CY
      ELSE
        B=A
      ENDIF
      TEMP=B/(A*A+CY*CY)
```

```
      DQWZ=TEMP*DEXP(-T*T)
      RETURN
      END
      PROGRAM ZBNDRYCF
C     TESTS THE ACCURACY OF THE DP CONTINUED FRACTION FOR
C     EXP(Z*Z)*ERFC(Z) WITH N=INT(C/AZ)+5 AGAINST DP QUADRATURE FOR
C     W(I*Z). C=75.0 SUFFICES FOR ERRORS O(10**(-15)) ALONG THE BOUNDARY
C     X=1.5D0, Y>=0.0D0, AZ=CABS(Z).
C
C     USES ZMATH AND AMOSSUBS
C-----------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      ICASE=7
      IF(ICASE.EQ.7) THEN
C       FORMULAE 7.4.13 & 7.4.14 REQUIRE Y.GT.0.0E0.
        JL=31
        IL=101
        PRINT *,"CF COMPARED TO DP QUADRATURE OF W(I*Z) A&S EQNS. 7.4.13
     & & 7.4.14"
            PRINT *,"      X               Y          REL ERR0R
     & RESULT"
        EPS=1.0D-14
        DO 10 K=1,2
        DO 150 J=1,JL
          X=-EPS+1.5D0+DBLE(FLOAT(J-1))/20.0D0
        DO 160 I=1,IL
          Y=6.0D0*DBLE(FLOAT(I-1))/100.0D0
C         Z=CMPLX(X,Y)
          ZR=X
          ZI=Y
          KODE=2
          CALL CONTFRAC(ZR,ZI,KODE,CYR,CYI,IERR)
          IF(IERR.EQ.2) THEN
            PRINT *, 'OVERFLOW FOR Z = ',ZR,ZI
            GOTO 160
          ENDIF
C         Z IS REPLACED BY I*Z IN W TO GET EXP(Z*Z)*ERFC(Z)
C         I*Z=-Y + I*X
          DX=-Y
          DY=X
          CALL ZW(DX,DY,WANSR,WANSI)
          CANSR=CYR
          CANSI=CYI
          DERRR=CANSR-WANSR
          DERRI=CANSI-WANSI
          ERR=ZABS(DERRR,DERRI)
          IF(ZABS(WANSR,WANSI).NE.0.0D0) THEN
            ERR=ERR/ZABS(WANSR,WANSI)
          ENDIF
          KOUNT=KOUNT+1
          IF(MOD(KOUNT,10).EQ.0) THEN
            PAUSE
        PRINT *,"CF COMPARED TO DP QUADRATURE OF W(I*Z) A&S EQNS. 7.4.13
     & & 7.4.14"
            PRINT *,"      X               Y          REL ERR0R
     & RESULT"
          ENDIF
          PRINT 905,ZR,ZI,ERR,WANSR,WANSI
          PRINT 906,CYR,CYI
          IF(I.EQ.1) THEN
            KKODE=3
            YY=DRERF(ZR,KKODE,NZ)
            XX=0.0D0
            PRINT 907,ZR,YY,XX
  907       FORMAT(D25.16,17X,2D14.6)
          ENDIF
  905     FORMAT(5D14.6)
```

```
  906     FORMAT(42X,2D14.6)
  160   CONTINUE
  150   CONTINUE
        EPS=-EPS
   10   CONTINUE
      ENDIF
      END
      SUBROUTINE CONTFRAC(ZR,ZI,KODE,CANSR,CANSI,IERR)
C-----------------------------------------------------------------------
C     CONTINUED FRACTION FOR ERFC(Z), X.GT.1.5 AND CABS(Z).LE.6.0.
C     N=INT(C/AZ+5.0) FOR ACCURACIES TOL=MAX(UNIT ROUND OFF,1.E-15)
C-----------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DATA FNORM,GNORM /1.12837916709551D0,  0.564189583547756D0/
      IERR=0
      AZ=ZABS(ZR,ZI)
      N=INT(SNGL((75.0D0/AZ+5.0D0)))
      FN=DBLE(FLOAT(N))
      CALL ZMLT(ZR,ZI,ZR,ZI,CZR,CZI)
C      TERM=CZ
      TERMR=CZR
      TERMI=CZI
      DO 30 I=1,N
        FNH=FN-0.5D0
        TEMPR=FNH*TERMR
        TEMPI=FNH*TERMI
        DENR=FN+TERMR
        DENI=TERMI
        CALL ZDIV(TEMPR,TEMPI,DENR,DENI,ANSR,ANSI)
        TERMR=CZR+ANSR
        TERMI=CZI+ANSI
C        TERM=CZ+(FNH*TERM)/(FN+TERM)
        FN=FN-1.0D0
   30 CONTINUE
      IF(KODE.EQ.2) GO TO 35
C      CZ=-CZ
C      ER=REAL(CZ)
C      IF(ABS(ER).GT.ELIM) THEN
C        IF(ER.GT.0.0E0) THEN
C          IERR=2
C          CANS=CMPLX(0.0E0,0.0E0)
C          RETURN
C        ELSE
C          RER=0.0E0
C        ENDIF
C      ELSE
C        RER=EXP(ER)
C      ENDIF
C      EI=AIMAG(CZ)
C      CEI=COS(EI)
C      SEI=SIN(EI)
C      CZ=CMPLX(RER,0.0E0)*CMPLX(CEI,SEI)
C      SUM=ZP*CMPLX(GNORM,0.0E0)/TERM
C      TERM=-CZ*SUM
   35 CONTINUE
      TEMPI=0.0D0
      CALL ZDIV(GNORM,TEMPI,TERMR,TERMI,ANSR,ANSI)
      CALL ZMLT(ZR,ZI,ANSR,ANSI,CANSR,CANSI)
C      CANS=ZP*CMPLX(GNORM,0.0E0)/TERM
      IF(ZR.GE.0.0D0) RETURN
C      ER=REAL(CZ)
C      IF(ABS(ER).GT.ELIM) THEN
C        IF(ER.GT.0.0E0) THEN
C          IERR=2
C          CANS=CMPLX(0.0E0,0.0E0)
C          RETURN
C        ELSE
```

```
C          RER=0.0E0
C        ENDIF
C      ELSE
C        RER=EXP(ER)
C      ENDIF
C      EI=AIMAG(CZ)
C      CEI=COS(EI)
C      SEI=SIN(EI)
C      CZ=CMPLX(RER,0.0E0)*CMPLX(CEI,SEI)
C      CANS=CZ+CZ-CANS
      RETURN
      END
      SUBROUTINE ZW(DX,DY,WANSR,WANSI)
C
C     PLAZMA DISPERSION FUNCTION W(Z)=DEXP(-Z*Z)*ERFC(-I*Z) BY
C     DOUBLE PRECISION QUADRATURE, Z=X+I*Y.
C
C     DX AND DY ARE DOUBLE PRECISION, Z=DX+I*DY
C     WANSR=DOUBLE PRECISION REAL PART OF W(Z)
C     WANSI=DOUBLE PRECISION IMAGINARY PART OF W(Z)
C-----------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      COMMON/CDQWZ/PN,CX,CY,IR
      DATA PI /3.14159265358979D0/
      EXTERNAL DQWZ
      X=DX
      Y=DY
      IF(DY.LT.0.0D0) THEN
        X=-DX
        Y=-DY
      ENDIF
      CX=X
      CY=Y
      INIT=0
      X1=0.0D0
      SIG=4.0D0
      REL=1.0D-15
      PN=1.0D0
      IR=0
      CALL DQUAD8(DQWZ,INIT,X1,SIG,REL,X2,QANSP,IERR)
      CALL DQUAD8(DQWZ,INIT,X1,SIG,REL,X2,QANSP,IERR)
      INIT=0
      X1=0.0D0
      SIG=4.0D0
      REL=1.0D-15
      PN=-1.0D0
      IR=0
      CALL DQUAD8(DQWZ,INIT,X1,SIG,REL,X2,QANSN,IERR)
      CALL DQUAD8(DQWZ,INIT,X1,SIG,REL,X2,QANSN,IERR)
      WANSR=(QANSP+QANSN)/PI
      INIT=0
      X1=0.0D0
      SIG=4.0D0
      REL=1.0D-15
      PN=1.0D0
      IR=1
      CALL DQUAD8(DQWZ,INIT,X1,SIG,REL,X2,QANSP,IERR)
      CALL DQUAD8(DQWZ,INIT,X1,SIG,REL,X2,QANSP,IERR)
      INIT=0
      X1=0.0D0
      SIG=4.0D0
      REL=1.0D-15
      PN=-1.0D0
      IR=1
      CALL DQUAD8(DQWZ,INIT,X1,SIG,REL,X2,QANSN,IERR)
      CALL DQUAD8(DQWZ,INIT,X1,SIG,REL,X2,QANSN,IERR)
      WANSI=(QANSP+QANSN)/PI
```

```
      IF(DY.LT.0.0D0) THEN
        DZ2R=DX*DX-DY*DY
        DEX=DEXP(-DZ2R)
        DTXY=2.0D0*DX*DY
        DSXY=DSIN(DTXY)
        DCXY=DCOS(DTXY)
        DTRMR=DEX*DCXY
        DTRMI=-DEX*DSXY
        WANSR=DTRMR+DTRMR-WANSR
        WANSI=DTRMI+DTRMI-WANSI
      ENDIF
      RETURN
      END
      DOUBLE PRECISION FUNCTION DQWZ(T)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      COMMON/CDQWZ/PN,CX,CY,IR
      A=CX-PN*T
      IF(IR.EQ.0) THEN
        B=CY
      ELSE
        B=A
      ENDIF
      TEMP=B/(A*A+CY*CY)
      DQWZ=TEMP*DEXP(-T*T)
      RETURN
      END
```