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 $\mathbf{I_1}$

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

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

and similarly for $\overline{I}_1^c(a,b,T)$. However the subroutine INTEGI1 always returned $\overline{I}_1(a,b,T)$ or $\overline{I}_1^c(a,b,T)$ (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). $\overline{I}_1(a,b,T)$ and $\overline{I}_1^c(a,b,T)$ 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.