Geological Survey of Canada
Commission géologique du Canada

PAPER 87-12

WIENER PACK: A SUBROUTINE PACKAGE FOR COMPUTING PROBABILITIES ASSOCIATED WITH WIENER AND BROWNIAN BRIDGE PROCESSES

C.F. Chung

1987

Canada

GEOLOGICAL SURVEY OF CANADA
PAPER 87-12

WIENER PACK: A SUBROUTINE PACKAGE FOR COMPUTING PROBABILITIES ASSOCIATED WITH WIENER AND BROWNIAN BRIDGE PROCESSES

C.F. Chung

1987

Available in Canada through

authorized bookstore agents and other bookstores

or by mail from

Canadian Government Publishing Centre
Supply and Services Canada
Ottawa, Canada K1A 0S9

and from

Geological Survey of Canada offices:

601 Booth Street
Ottawa, Canada K1A 0E8

3303-33rd Street N.W.,
Calgary, Alberta T2L 2A7

100 West Pender Street
Vancouver, British Columbia V6B 1R8

A deposit copy of this publication is also available
for reference in public libraries across Canada

Cat. No. M44-87/12E Canada: $5.00
ISBN 0-660-12531-5 Other countries: $6.00

Price subject to change without notice

Critical Reader

F.P. Agterberg

Original manuscript received: 1985 - 10 - 10
Final version approved for publication: 1987 - 01 - 06

Contents

WIENER PACK: A SUBROUTINE PACKAGE FOR COMPUTING PROBABILITIES ASSOCIATED WITH WIENER AND BROWNIAN BRIDGE PROCESSES

Abstract

A subroutine package WIENER PACK written in FORTRAN-77 has been developed to compute the probabilities associated with Wiener and Brownian bridge processes of the following forms:

$P\{\ W(t) \le \gamma + \delta\, t\colon a \le t \le b\ \}$;
$P\{\ |W(t)| \le \gamma + \delta\, t\colon a \le t \le b\ \}$;
$P\{\ B(t) \le \gamma : a \le t \le b\ \}$;
$P\{\ |B(t)| \le \gamma : a \le t \le b\ \}$,

where W(.) *and* B(.) *are Wiener and Brownian bridge processes, respectively.*

Résumé

Un ensemble de sous-routiers, WIENER PACK, en FORTRAN-77, a été mis au point pour calculer les probabilités associées aux processus de liaison Wiener et Brown dont le symbolisation est la suivante:

$P\{\ W(t) \le \gamma + \delta\, t\colon a \le t \le b\ \}$;
$P\{\ |W(t)| \le \gamma + \delta\, t\colon a \le t \le b\ \}$;
$P\{\ B(t) \le \gamma : a \le t \le b\ \}$;
$P\{\ |B(t)| \le \gamma : a \le t \le b\ \}$,

Les symboles W(t) *et* B(z) *identifient les processus de liaison Wiener et Brown.*

INTRODUCTION

Weak convergence and strong approximation of empirical, empirical quantile, product-limit and Quantile-Quantile processes to Brownian bridge and Wiener processes are well known (Breslow and Crowley, 1974; Csörgö and Révész, 1978; Csörgö and Horváth, 1985, Aly et al., 1985; Aly and Bleuer, 1983). These theories can, in principle, be used to construct confidence bands for their statistical functions.

When these methodologies are applied to statistical problems, however, the computation of probabilities for functionals of interest for the approximating Gaussian processes plays a crucial role, and the required computations on functionals of appropriate Gaussian processes are frequently complicated. Without such computations, recent statistical advancements using weak convergence and strong approximation methodologies cannot be used in practice.

A subroutine package WIENER PACK written in FORTRAN-77 has been developed to compute the probabilities associated with Wiener and Brownian bridge processes of the following forms:

(A.1) $P\{\ W(t) \le \gamma : a\le t\le b\ \}$, $a\ge 0$, $b<\infty$, $\gamma>0$;

(A.2) $P\{\ |W(t)| \le \gamma : a\le t\le b\ \}$, $a\ge 0$, $b<\infty$, $\gamma>0$;

(A.3) $P\{\ W(t) \le \gamma+\delta t : a\le t\le b\ \}$, $a\ge 0$, $b<\infty$, $\gamma>0$, $\gamma+\delta b\ge 0$;

(A.4) $P\{\ |W(t)| \le \gamma+\delta t : a\le t\le b\ \}$, $a\ge 0$, $b<\infty$, $\gamma>0$, $\gamma+\delta b\ge 0$;

(A.5) $P\{\ B(t) \le \gamma : a\le t\le b\ \}$, $a\ge 0$, $b\le 1$, $\gamma>0$;

(A.6) $P\{\ |B(t)| \le \gamma : a\le t\le b\ \}$, $a\ge 0$, $b\le 1$, $\gamma>0$.

where W(.) and B(.) are Wiener and Brownian bridge processes, respectively.

In addition, for given probability p, the package enables the user to compute any one of the parameters a, b or γ in the formulae (A.1), (A.2), (A.5), and (A.6), by specifying the other two parameters.

Some special cases of the probabilities of (A.5) and (A.6) have been developed by Csörgö and Horváth (1981). Barr and Davidson (1973), Koziol and Byar (1975), and Schumacher (1984) among many others. A computer program package called WIENER PACK has been developed to compute the probabilities of the above forms for any given values of the parameters a, b, γ and δ. Inversely, the package also enables one to compute any one of the parameters – a, b and γ in (A.1), (A.2), (A.5) and (A.6) for a given probability level and given values for the other two of the parameters a, b and c.

Obviously, (A.1) and (A.2) are special cases of (A.3) and (A.4), respectively. It can also be shown that (A.5) and (A.6) are again special cases of (A.3) and (A.4), respectively. However, in actual computations, each of the formulae related to Wiener processes shown is programmed separately for computational efficiency.

In this paper, some of the available methods for computing probabilities associated with Wiener and Brownian bridge processes are summarized. More detailed discussions and proofs of the formulae quoted here are given in Chung (1986).

Acknowledgments

I would like to thank Dr. M. Csörgö, Carleton University for his valuable remarks and suggestions.

THEORETICAL BACKGROUND FOR WIENER PROCESS

Definition of Wiener process

A stochastic process $\{ W(t) : 0 \le t < \infty \}$ is called a Wiener process if :

(i) $W(t) - W(s) \sim N(0, t-s)$ for all $0 \le s \le t < \infty$ and $W(0)=0$;
(ii) W(t) is an independent increment process, that is $W(t_2)-W(t_1), W(t_4)-W(t_3), \ldots, W(t_{2i})-W(t_{2i-1})$ are independent random variables for all $0 \le t_1 < t_2 < \ldots < t_{2i-1} < t_{2i} < \infty$ (i = 2, 3, ...), and
(iii) the sample path function W(t) is continuous in t with probability 1.

Note that (i) and (ii) imply that the covariance function of a Wiener process is

$$E(W(s)W(t)) = s \wedge t. \tag{1}$$

Conversely, a continuous Gaussian process having the covariance function of (1) is a Wiener process.

Formula for $p = P\{ W(t) \le \gamma + \delta t : 0 \le t \le b \}$.

Case 1.

Let $b>0$, $\gamma>0$ and $\gamma+\delta b \ge 0$. Then

$$P\{ W(t) \le \gamma+\delta t : 0 \le t \le b \} = \Phi [(\gamma+\delta b)/\sqrt{b}] - e^{-2\gamma\delta} \Phi [(-\gamma+\delta b)/\sqrt{b}]. \tag{2}$$

where

$$\Phi (y) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{y} e^{-\frac{1}{2}h^2} dh.$$

When $\delta = 0$, then

$$P\{ W(t) \le \gamma : 0 \le t \le b \} = 2 \Phi (\gamma/\sqrt{b}) - 1. \tag{3}$$

Case 2.

If $\gamma > 0$ and $b = 0$, then $p = 1$.

Formula for $p = P\{ W(t) \le \gamma+\delta t : a \le t \le b \}$

Case 1.

Let $a > 0$, $\gamma>0$, $a<b$, $\gamma+\delta a \ge 0$ and $\gamma+\delta b \ge 0$.

Then we have

(4) $$P\{ W(t) \le \gamma+\delta t : a\le t\le b \} = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{(\gamma+\delta a)/\sqrt{a}} e^{-\frac{1}{2}x^2} \Big\{ \Phi [(\gamma+\delta b-\sqrt{a}x)/\sqrt{b-a}] - e^{-2(\gamma+\delta a-\sqrt{a}x)\delta} \Phi [(-\gamma-2a\delta+b\delta+\sqrt{a}x)/\sqrt{b-a}] \Big\} dx$$

Note that when $\delta = 0$, we have

(5) $$P\{ W(t) \le \gamma : a\le t\le b \} = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\gamma/\sqrt{a}} e^{-\frac{1}{2}x^2} \Big\{ 2\,\Phi [(\gamma-\sqrt{a}x)/\sqrt{b-a}] - 1 \Big\} dx.$$

Rényi (1953, (3.6) on page 208) and Csörgö (1967, (3.4) on page 554) have also given (5).

Case 2.

Let $a > 0$, $\gamma > 0$, $a = b$ and $\gamma+\delta a \ge 0$. Then,

(6) $$P\{ W(a) \le \gamma+\delta a \} = \Phi [(\gamma+\delta a)/\sqrt{a}].$$

Formula for $P\{ \gamma_2+\delta_2 t \le W(t) \le \gamma_1+\delta_1 t : 0\le t\le b \}$.

Case 1.

(Anderson, 1960, (4.32) on page 180). Let $\gamma_1 > 0$, $\gamma_2<0$, $b >0$, $\gamma_2+\delta_2 b \le \gamma_1+\delta_1 b$. Then we have

(7) $$P\{ \gamma_2+\delta_2 t \le W(t) \le \gamma_1+\delta_1 t : 0\le t\le b \} = 1 - \Phi (w/\sqrt{b}) - \Phi (-v/\sqrt{b}) + \sum_{k=1}^{\infty} \Big\{ e^{s_1}[\Phi (q_1)-\Phi(p_1)] + e^{s_2} [\Phi (p_2)-\Phi(q_2)] - e^{s_3} [\Phi(q_3)-\Phi(p_3)] - e^{s_4} [\Phi(p_4)-\Phi(q_4)] \Big\}$$

where

$\gamma' = \gamma_1-\gamma_2$, $\delta' = \delta_1-\delta_2$, $v = \gamma_1+\delta_1 b$, $w = \gamma_2+\delta_2 b$,

$s_1 = -2 [k^2\gamma'\delta' + k(\gamma_1\delta_2-\gamma_2\delta_1)]$,

$s_2 = -2 [k^2\gamma'\delta' - k (\gamma_1\delta_2-\gamma_2\delta_1)]$,

$s_3 = -2 [k^2\gamma'\delta' + k(\gamma'\delta_2+\delta'\gamma_2) + \gamma_2\delta_2]$.

$s_4 = -2 [k^2\gamma'\delta' - k(\gamma'\delta_1+\delta'\gamma_1) + \gamma_1\delta_1]$,

$p_1 = (-2k\gamma' + w) / \sqrt{b}$, $q_1 = (-2k\gamma' + v) / \sqrt{b}$,

$p_2 = (-2k\gamma' - w) / \sqrt{b}$, $q_2 = (-2k\gamma' - v) / \sqrt{b}$,

$p_3 = (-2k\gamma'+w-2\gamma_2) / \sqrt{b}$, $q_3 = (-2k\gamma'+v-2\gamma_2) / \sqrt{b}$,

$p_4 = (-2k\gamma'-w+2\gamma_1) / \sqrt{b}$, $q_4 = (-2k\gamma'-v+2\gamma_1) / \sqrt{b}$,

From (7), it can be derived that

From (7), it can be derived that

$$(8)\qquad P\{ \gamma_2+\delta_2 t \le W(t) \le \gamma_1+\delta_1 t : 0\le t\le b \}$$
$$= P\{ -\gamma_1-\delta_1 t \le W(t) \le -\gamma_2-\delta_2 t : 0\le t\le b \}.$$

Note that when $\gamma_1 = -\gamma_2 = \gamma\ (>0)$ and $\delta_1 = -\delta_2 = \delta\ (\ge 0)$, then

$$(9)\qquad P\{ |W(t)| \le \gamma+\delta t : 0\le t\le b \}$$
$$= 1 - 2\,\Phi(-u/\sqrt{b}) - 2\sum_{k=1}^{\infty} (-1)^{k+1} e^{-2k^2\gamma\delta} \left[\Phi[(-2k\gamma+u)/\sqrt{b}] - \Phi[(-2k\gamma-u)/\sqrt{b}] \right]$$

where $u = \gamma + \delta b$. Gillaspie and Fisher (1979, $P(\gamma,\delta,b)$ on page 921) have also discussed (9).

When $\delta_1 = \delta_2 = 0$ in (7), we have

$$(10)\qquad P\{ \gamma_2 \le W(t) \le \gamma_1 : 0\le t\le b \}$$
$$= 1 - 2\sum_{k=1}^{\infty} (-1)^{k+1} \left[\Phi[(-k\gamma'+\gamma_1)/\sqrt{b}] + \Phi[(-k\gamma'-\gamma_2)/\sqrt{b}] \right]$$

where $\gamma' = \gamma_1 - \gamma_2$. It was also denoted by Feller (1966, (5.8) on page 329) as $\lambda.(b, \gamma_1)$. In addition Feller (1966, (5.9) on p. 330) and M. Csörgö (1967, (2.5) on page 553) have shown an alternative formula for (10) as follows:

$$(11)\qquad P\{ \gamma_2 \le W(t) \le \gamma_1 : 0\le t\le b \}$$
$$= \frac{4}{\pi}\sum_{k=0}^{\infty} \frac{1}{2k+1} \exp\left[\frac{-(2k+1)^2\pi^2 b}{2(\gamma_1-\gamma_2)^2} \right] \sin\left[\frac{(2k+1)\,\pi\gamma_1}{\gamma_1-\gamma_2} \right].$$

Feller (1966) noted that the series in (10) converges quickly when b is small whereas in (11) it converges rapidly when b is large.

When $\gamma_1 = -\gamma_2 = \gamma$, (10) and (11) are reduced to

$$(12)\qquad P\{ |W(t)| \le \gamma : 0\le t\le b \}$$
$$= 1 - 4\sum_{k=1}^{\infty} (-1)^{k+1} \Phi[-(2k-1)\gamma/\sqrt{b}]$$
$$= \frac{4}{\pi}\sum_{k=0}^{\infty} \frac{(-1)^k}{2k+1} \exp\left[\frac{-(2k+1)^2\pi^2 b}{8\gamma^2} \right].$$

When $b = 1$ in (12), Feller (1966, (5.11) on page 330) has denoted it as $L(\gamma)$.

Case 2.

Let $\gamma_1 > 0$, $\gamma_2 < 0$, $b = 0$. Then the probability is simply 1.

Formula for $P\{ \gamma_2 + \delta_2 t \leq W(t) \leq \gamma_1 + \delta_1 t : a \leq t \leq b \}$.

Case 1.

Let $\gamma_1 > 0$, $\gamma_2 < 0$, $a > 0$, $\gamma_2 + \delta_2 b \leq \gamma_1 + \delta_1 b$.

Then we have

$$(13) \quad P\{ \gamma_2 + \delta_2 t \leq W(t) \leq \gamma_1 + \delta_1 t : a \leq t \leq b \}$$
$$= \frac{1}{\sqrt{2\pi}} \int_{q''/\sqrt{a}}^{q'/\sqrt{a}} \Gamma(q'' - \sqrt{a}h, \delta_2, q' - \sqrt{a}h, \delta_1, b - a)\, e^{-\frac{1}{2}h^2}\, dh,$$

where $q' = \gamma_1 + \delta_1 a$, $q'' = \gamma_2 + \delta_2 a$ and

$$(14) \quad \Gamma(u', v', u, v, w)$$
$$= P\{ u' + v't \leq W(t) \leq u + vt : 0 \leq t \leq w \},$$

which is discussed in (7).

When $\gamma_1 = -\gamma_2 = \gamma\ (> 0)$ and $\delta_1 = -\delta_2 = \delta\ (\geq 0)$, we have

$$(15) \quad P\{ |W(t)| \leq \gamma + \delta t : a \leq t \leq b \}$$
$$= \frac{2}{\sqrt{2\pi}} \int_{o}^{q/\sqrt{a}} \Gamma(-q - \sqrt{a}h, \delta, q - \sqrt{a}h, \delta, b - a)\, e^{-\frac{1}{2}h^2}\, dh,$$

where $q = \gamma + \delta a$. If $\delta = 0$ in (15), we obtain

$$(16) \quad P\{ |W(t)| \leq \gamma : a \leq t \leq b \}$$
$$= \frac{2}{\sqrt{2\pi}} \int_{o}^{\gamma/\sqrt{a}} \Gamma'(-\gamma - \sqrt{a}h, \gamma - \sqrt{a}h, b - a)\, e^{-\frac{1}{2}h^2}\, dh,$$

where

$$(17) \quad \Gamma'(u', u, w) = P\{ u' \leq W(t) \leq u : 0 \leq t \leq w \},$$

which is discussed in (10) and (11).

Case 2.

Let $\gamma_1 > 0$, $\gamma_2 < 0$, $a = b$ and $\gamma_2 + \delta_2 a \leq \gamma_1 + \delta_1 a$.

Then we have

$$(18) \quad P\{ \gamma_2 + \delta_2 a \leq W(a) \leq \gamma_1 + \delta_1 a \}$$
$$= \Phi[(\gamma_1 + \delta_1 b)/\sqrt{b}] - \Phi[(\gamma_2 + \delta_2 b)/\sqrt{b}].$$

THEORETICAL BACKGROUND FOR BROWNIAN BRIDGE PROCESS.

Definition of Brownian bridge process.

A stochastic process $\{ B(t); 0 \le t \le 1 \}$ is called a Brownian bridge if:

(i) the joint distribution of $B(t_1), \ldots, B(t_n)$, $0 \le t_1 \le \ldots \le t_n \le 1$, is Gaussian with $E(B(t)) = 0$; (ii) the covariance function is

(19) $\quad EB(t)B(s) = s \wedge t - st$; and

(iii) the sample path function B(t) is continuous in t with probability 1.

Note that (ii) implies that $B(0) = B(1) = 0$ amost surely. If $B(t) = W(t) - tW(1)$ $(0 \le t \le 1)$ where W(t) is a Wiener process, then B(t) is a Brownian bridge. On the other hand, if $W(t) = (t+1)\,B[\,t/(t+1)\,]$, $t \ge 0$, then W(t) is a Wiener process. Also $B(t) = (1-t)\,W[\,t/(1-t)]$, $0 \le t < 1$ is a Brownian bridge process.

Formula for $p = P\{ B(t) \le c : 0 \le t \le b \}$.

Case 1.

Let $0<b<1$ and $c \ge 0$. Then

(20) $$P\{ B(t) \le c : 0 \le t \le b \} = \Phi\left[c/\sqrt{b(1-b)} \right] - e^{-2c^2}\, \Phi\left[-c(1-2b)/\sqrt{b(1-b)} \right].$$

Koziol and Byar (1975, (2.3) on page 508) have derived (20) denoting it as $G_b(c)$, although a misprint occurs in formula (2.3) on page 508 of their paper. i.e., $(T\text{-}T^2)^{-1/2}$ should be printed as $(T\text{-}T^2)^{-1/2}d$ in the probability argument of their formula (2.3). Hall and Wellner (1980, (2.8) on page 136) have also derived (20) denoting it as as $G_b{}^+(c)$. However, their notation is confusing and their formula (2.8) is not correct.

Case 2.

Let $b = 1$. Then we have

(21) $$P\{ B(t) \le c : 0 \le t \le 1 \} = 1 - e^{-2c^2},$$

which is the limiting distribution function of the one-sided Kolmogorov-Smirnov statistic

$$\sqrt{n} \sup_{-\infty<x<\infty} (F_n(x) - F(x))$$

where $F_n(.)$ is the empirical distribution function of a continuous distribution function F(.) (cf. Doob, 1949).

Formula for $p = P\{ B(t) \le c : a \le t \le b \}$.

Case 1.

Let $0 < a < b < 1$ and $c \ge 0$. Then

(22) $$P\{ \sup_{a \le s \le b} B(s) \le c \}$$

$$= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{c/(1-a)} e^{-\frac{1}{2}h^2} \left\{ \Phi\left[\frac{c\sqrt{1-a} - (1\text{-}b)\sqrt{ah}}{\sqrt{1-b}\,\sqrt{b-a}} \right] \right.$$

$$\left. - e^{-2c[c-\sqrt{a(1-a)}\,h]/(1-a)}\, \Phi\left[\frac{c(2b-a-1)+(1-b)\sqrt{a(1-a)}\,h}{\sqrt{1-a}\,\sqrt{1-b}\,\sqrt{b-a}} \right] \right\} dh$$

Csáki (1981, (2.17) on page 257) has derived (22) as the limiting distribution function of a special empirical process of limited range and his formula has been quoted by Csörgö (1983, (4.2.10) on page 37).

Case 2.

Let $b=1$, $0<a<b$, $c\geq 0$. Then, by the symmetricity property of the Brownian bridge process, we have

$$(23) \qquad P\{\, B(t) \leq c : a\leq t\leq 1 \,\}$$
$$= P\{\, B(t) \leq c : 0\leq t\leq 1-a \,\}$$
$$= \Phi\, [\, c/\sqrt{a(1-a)}\,] \;-\; e^{-2c^2}\, \Phi\, [\, c(1-2a)/\sqrt{a(1-a)}\,].$$

Case 3.

Let $a = b$ ($\neq$ 0 or 1) and $c \geq 0$. Then

$$(24) \qquad P\{\, B(a) \leq c \,\} = \Phi\, [\, c/\sqrt{a(1-a)}\,].$$

Formula for $p = P\{\, |B(t)| \leq c : 0\leq t\leq b \,\}$.

Case 1.

Let $b < 1$ and $c \geq 0$. Then we have

$$(25) \qquad P\{\, |B(t)| \leq c : 0\leq t\leq b \,\}$$
$$= 1 - 2\,\Phi\left(-c/\sqrt{b(1-b)} \right) - 2 \sum_{k=1}^{\infty} (-1)^{k+1} e^{-2k^2c^2}$$
$$\left[\Phi\left(\frac{-c(2k-2kb-1)}{\sqrt{b(1-b)}} \right) - \Phi\left(\frac{-c(2k-2kb+1)}{\sqrt{b(1-b)}} \right) \right].$$

Hall and Wellner (1980, (2.9) on page 136) have also shown (3.3.1) denoting it as $G_b(c)$.

Case 2.

Let $b = 1$ and $c \geq 0$. Then we have

$$(26) \qquad P\{\, |B(t)| \leq c : 0\leq t\leq 1 \,\}$$
$$= 1 - 2 \sum_{k=1}^{\infty} (-1)^{k+1} e^{-2k^2c^2},$$

which is the limiting distribution function of the two-sided Kolmogorov-Smirnov statistic or the empirical process

$$\sqrt{n} \sup_{-\infty<x<\infty} |\, F_n(x) - F(x) \,|$$

where $F_n(.)$ is the empirical distribution function of a continuous distribution function $F(.)$ (cf. Doob, 1949).

Formula for $p = P\{\ |B(t)| \le c : a \le t \le b\ \}$.

Case 1.

Let $0 < a < b < 1$ and $c \ge 0$. Then

$$(27)\qquad P\{\ |B(t)| \le c : a \le t \le b\ \}$$

$$= \frac{1}{\sqrt{2\pi}} \int_0^{\mu/\sqrt{a}} \Gamma(\ -\mu - \sqrt{a'}h, -c, \mu - \sqrt{a'}h, \delta_1, b' - a'\ )\, e^{-\frac{1}{2}h^2}\, dh,$$

where $\mu = c(1+a')$, $a' = a/(1-a)$, $b' = b/(1-b)$ and $\Gamma(.)$ is defined in (14).

Anderson and Darling (1952, (5.9) on page 210) have also considered (27) and have shown that

$$(28)\qquad P\{\ |B(t)| \le c : a \le t \le b\ \}$$

$$= \sum_{k=-\infty}^{\infty} (-1)^k e^{-2c^2k^2} M(2kc\sqrt{a'}, 2kc\sqrt{b'}, c\sqrt{a'}a, c\sqrt{b'}/b, -\sqrt{a'b'})$$

where $M(u,v,u',v',\rho)$ is the volume under the bivariate normal distribution surface with zero means, unit variances and correlation ρ which is above the rectangle with vertices $x = u \pm u'$ and $y = v \pm v'$.

Case 2.

Let $a = b$ ($\ne 0$ or 1) and $c \ge 0$. Then

$$(29)\qquad P\{\ |B(a)| \le c\ \} = 2\,\Phi[\ c/\sqrt{a(1-a)}\ ] - 1.$$

GENERAL DESCRIPTION OF THE ALGORITHMS.

Overview.

The package contains algorithms for computing the probabilities associated with Wiener and Brownian bridge processes of the following forms:

$$(30)\qquad \begin{aligned}
&P\{\ W(t) \le \gamma : a \le t \le b\ \},\ a \ge 0,\ b < \infty,\ \gamma > 0;\\
&P\{\ |W(t)| \le \gamma : a \le t \le b\ \},\ a \ge 0,\ b < \infty,\ \gamma > 0;\\
&P\{\ \gamma' \le W(t) \le \gamma : a \le t \le b\ \},\ b < \infty,\ \gamma > 0,\ \gamma' < 0;\\
&P\{\ W(t) \le \gamma + \delta t : a \le t \le b\ \},\ a \ge 0,\ b < \infty,\ \gamma > 0,\ \gamma + \delta b \ge 0;\\
&P\{\ |W(t)| \le \gamma + \delta t : a \le t \le b\ \},\ a \ge 0,\ b < \infty,\ \gamma > 0,\ \gamma + \delta b \ge 0;\\
&P\{\ \gamma' + \delta' t \le W(t) \le \gamma + \delta t : 0 \le t \le b\ \},\ b < \infty,\ \gamma > 0,\ \gamma' < 0,\ \gamma' + \delta' b \le \gamma + \delta b;\\
&P\{\ B(t) \le c : a \le t \le b\ \},\ a \ge 0,\ b \le 1,\ c > 0;\\
&P\{\ |B(t)| \le c : a \le t \le b\ \},\ a \ge 0,\ b \le 1,\ c > 0,
\end{aligned}$$

where $W(.)$ and $B(.)$ are Wiener and Brownian bridge processes, respectively.

The relationships between the formulae in the previous sections and the corresponding subroutines in WIENER PACK are listed in Table 1.

From a theoretical point of view, the two routines GWPAB and GWPSAB supersede all other routines except for WIEN1 and WIENER. That is, all the other routines are special cases of these two routines. The relationships are shown in Figure 1.

However, in actual computations, each formula related to Wiener processes shown in Table 1 is programmed separately for computational efficiency.

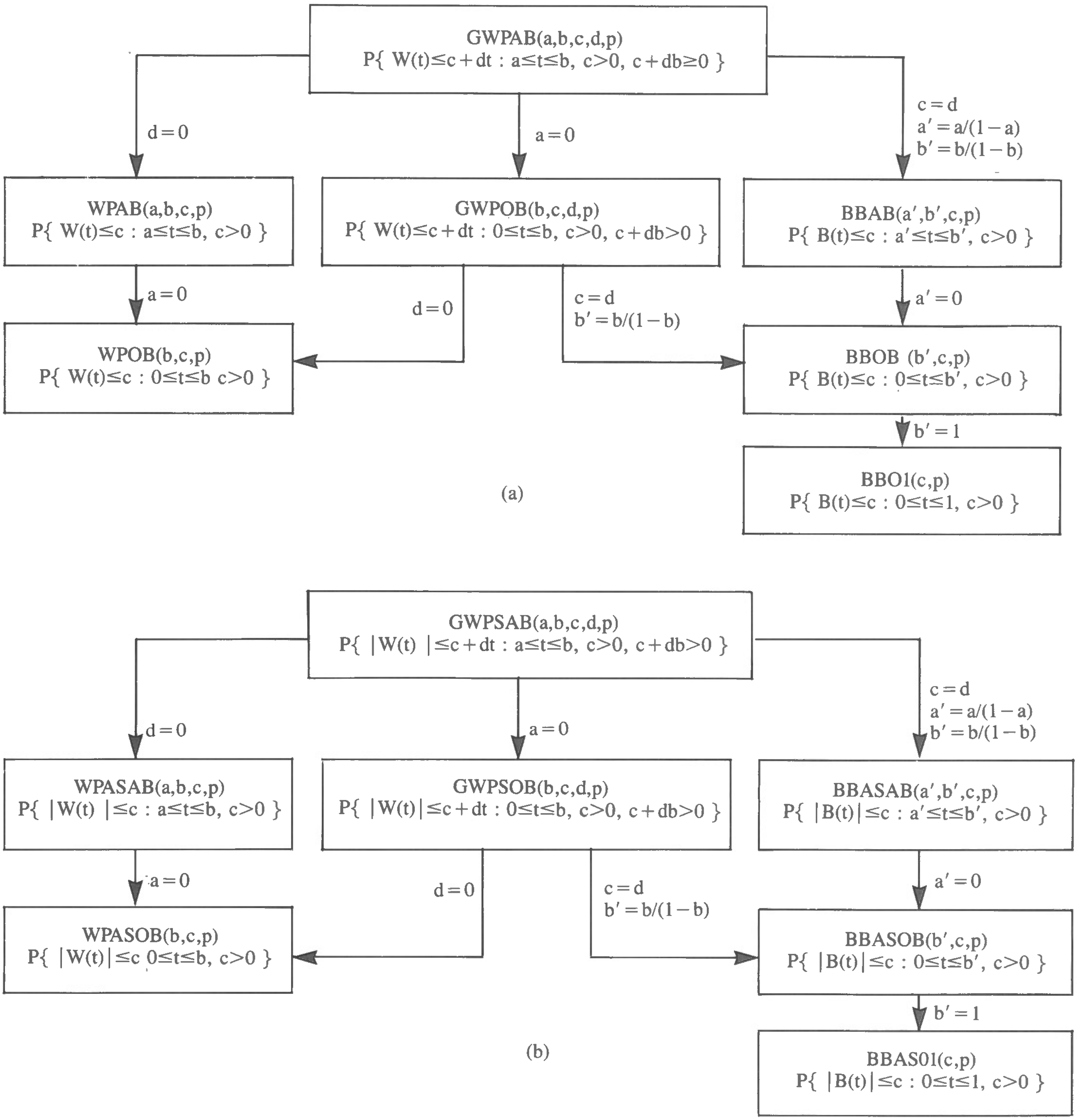
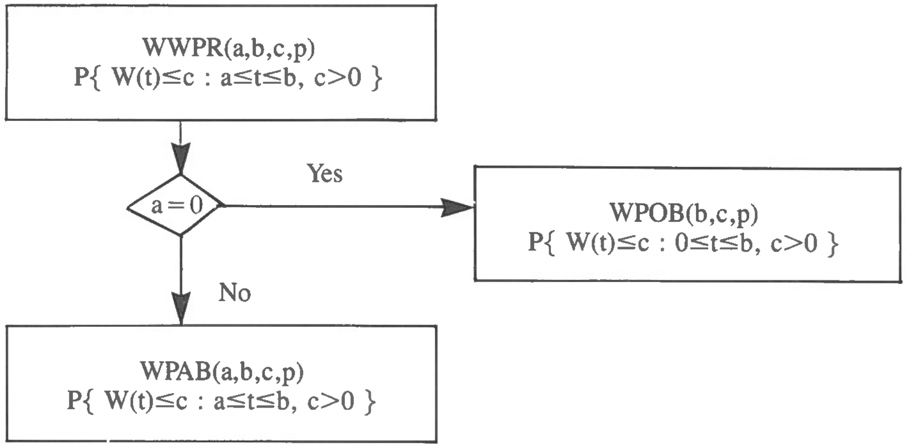
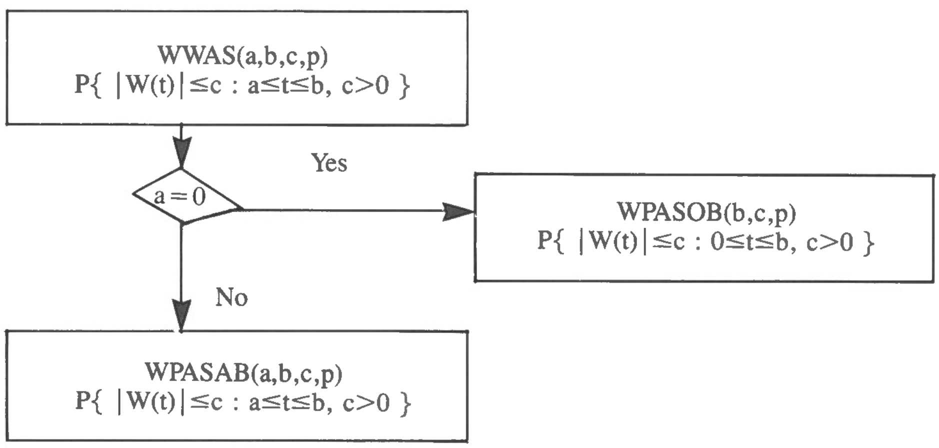


Figure 1. Interrelationships among the routines. The lower routines are the special cases of the original, upper routines and the equations beside the arrows indicate the relationships.



(a)

Figure 2. (a). General flowchart of WWPR which combines WPOB and WPAB. It calls WPOB or WPAB depending upon the input values a and b.



(b)

(b). General flowchart of WWAS which combines WPASOB and WPASAB. It calls WPASOB or WPASAB depending upon the input values a and b.

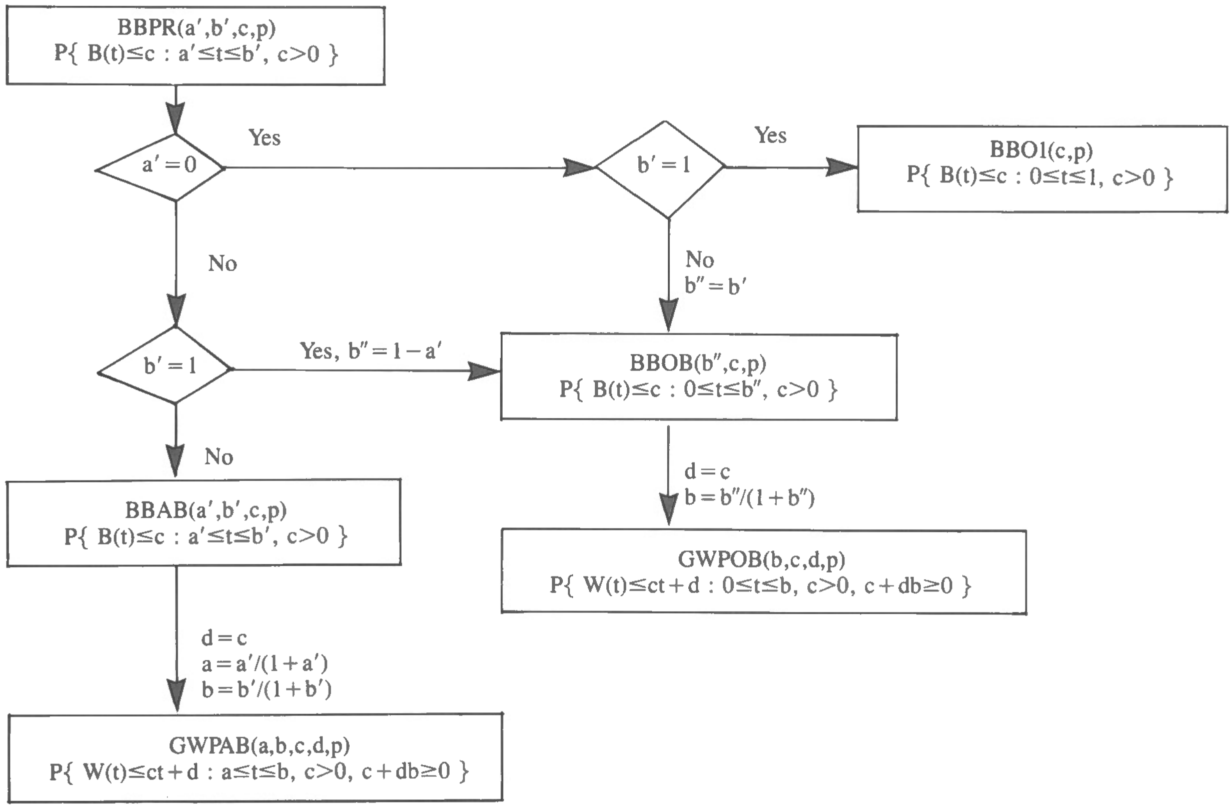


Figure 2(c). General flowchart of BBPR which combines BBO1, BBOB and BBAB. It calls BBO1, BBOB or BBAB depending upon the input value a′ and b′. It also illustrates how routines GWPOB and GWPAB are utilized for computing BBOB and BBAB, respectively.

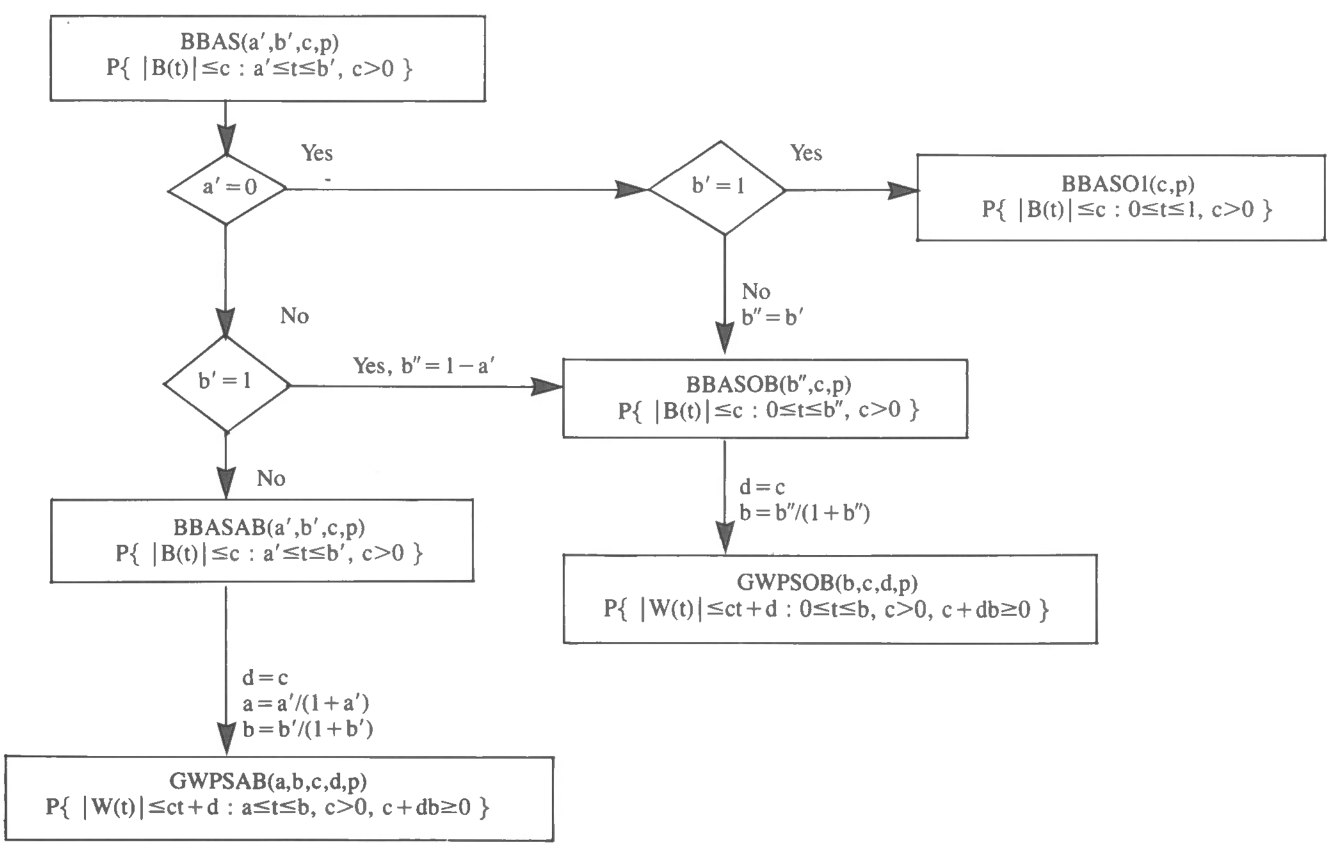


Figure 2(d). General flowchart of BBAS which combines BBASO1, BBASOB and BBASAB. It calls BBASO1, BBASOB or BBASAB depending upon the input value a′ and b′. It also illustrates how routines GWPSOB and GWPSAB are utilized for computing BBASOB and BBASAB, respectively.

Four routines were added to the package for easy access to the subroutines: WWPR for combining WWOB and WWAB, WWAS for WWASOB and WWASAB, BBPR for BBO1, BBOB and BBAB, and BBAS for BBASO1, BBASOB and BBASAB, and these relationships are illustrated in Figure 2.

For the numerical integrations required for the routines WPAB, WPASAB, GWPAB and GWPSAB in this package, subroutine QNG of QUADPACK (Piessens et al., 1983) has been used. See QUADPACK (p. 130-136) for the detailed description of QNG.

In addition, two routines INVWW and INVBB have been added to compute one of parameters selected for a given probability using the four routines WWPR, WWAS, BBPR and BBAS.

Consider p_1, p_2, p_3 and p_4 as functions of three parameters a, b and c in the following four formulae;

$$
\begin{aligned}
&p_1(a,b,c) = P\{ W(t) \le c : a \le t \le b \},\\
&p_2(a,b,c) = P\{ |W(t)| \le c : a \le t \le b \},\\
(31)\quad &p_3(a,b,c) = P\{ B(t) \le c : a \le t \le b \}\\
&p_4(a,b,c) = P\{ |B(t)| \le c : a \le t \le b \}.
\end{aligned}
$$

Chung (1986) have shown that p_1, p_2, p_3, and p_4 are monotonically non-decreasing and continuous functions with respect to c, a and $-b$. Subsequently the following functions can be defined:

(32) $q_k\ (a,b,p) := \{ c : p_k\ (a,b,c,) = p \}$,

(33) $r_k\ (b,c,p) := \{ a : p_k\ (a,b,c) = p \}$,

(34) $s_k\ (a,c,p) := \{ b : p_k\ (a,b,c) = p \}$,

for $k = 1, 2, 3$ and 4. Then using these monotonicity, and the routines WWPR, WWAS, BBPR and BBAS, we can iteratively compute the values of $q_k\ (a,b,p)$ for given values for a, b and p, the values of $r_k\ (b,c,p)$ for given values for b, c and p and the values of $s_k(a,c,p)$ for given values for a, c and p for $k = 1, 2, 3$ and 4. These iterative procedures are shown in Figures 3 and 4 for INVWW and INVBB – the former for Wiener process and the latter for Brownian bridge process.

Routines for Wiener process.

WPOB (b,γ,p): $p = P\{ W(t) \le \gamma : 0 \le t \le b \}$.

Given two parameters b and γ, this routine will compute and return the probability p.

WPAB (a,b,γ,p): $p = P\{ W(t) \le \gamma : a \le t \le b \}$.

Given the three parameters a, b and γ, this routine will compute and return the probability p. The integrand in (5) is computed by FUNCTION WFUNC1 and QNG is called the numerical integration.

WPASOB (b,γ,p): $p = P\{ |W(t)| \le \gamma : 0 \le t \le b \}$.

Given parameters b and γ, this routine will compute and return the probability p.

WIEN1 (γ′,γ,b,p): $p = P\{ \gamma' \le W(t) \le \gamma : 0 \le t \le b \}$.

Given three parameters b, γ and γ', this routine will compute and return probability p. If $\gamma = \gamma'$, then it does provide the same probability p as that obtained by WPAB (b,γ,p). This routine is used in the following routine WPASAB through FUNCTION WFUNC2.

WPASAB (a,b,γ,p): $p = P\{ |W(t)| \le \gamma : a \le t \le b \}$.

Given parameters a, b and γ, this routine will compute and return probability p. FUNCTION WFUNC2 evaluates the values for the integrand function in (16); the probability argument in the integrand is evaluated by WIEN1 and QNG is called the numerical integration.

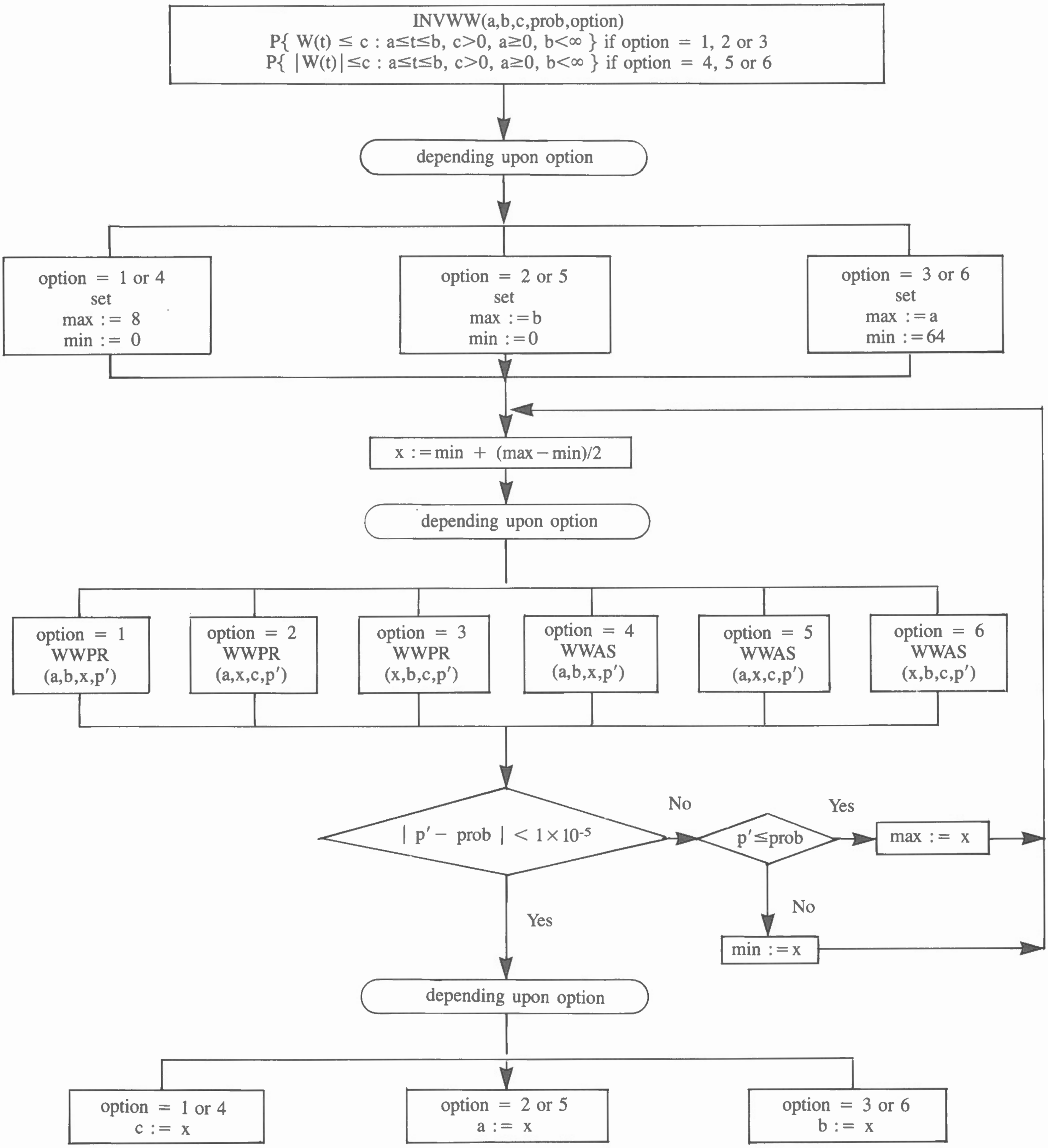


Figure 3. System flowchart for INVWW. It illustrates how the iterative procedure is used and the routines WWPR and WWAS are called for computations.

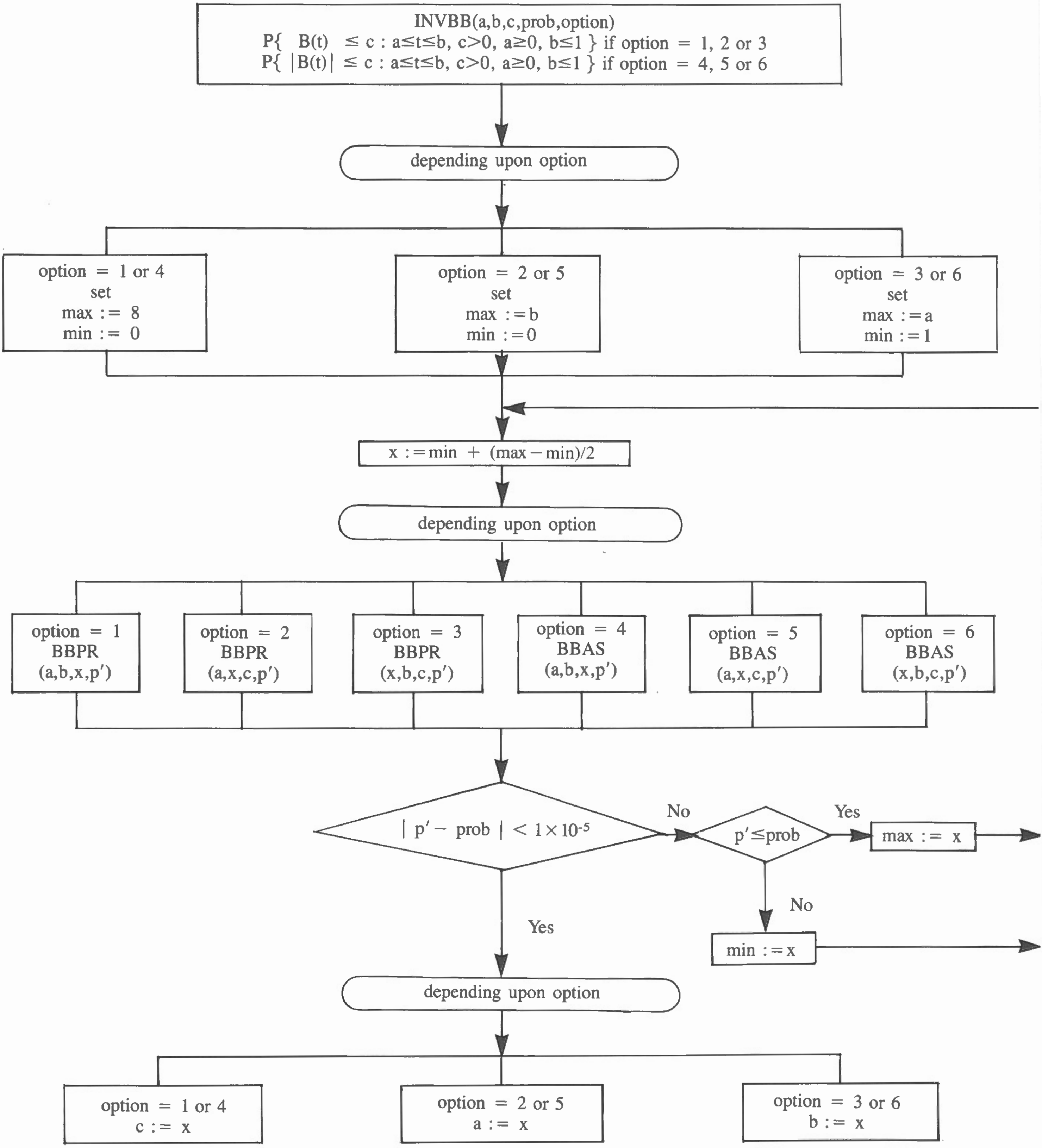


Figure 4. System flowchart for INVBB. It illustrates how the iterative procedure is used and the routines BBPR and BBAS are called for computations.

GWPOB (b, γ, δ, p): p= P{ W(t) $\leq \gamma+\delta t$: $0\leq t\leq b$ }.

Given parameters b, γ and δ, this routine will compute and return probability p. It is used by routines GWPAB and BBOB.

GWPAB (a,b,γ,δ,p): p =P{ W(t) $\leq \gamma+\delta t$: $a\leq t\leq b$ }.

Given parameters a, b, γ, and δ, this routine will compute and return probability p. FUNCTION WFUNC3 evaluates the values for the integrand function in (4), but the probability argument in the integrand is computed by GWPOB. QNG is called the numerical integration.

GWPSOB (b,γ,δ,p): P{ |W(t)| $\leq \gamma+\delta t$: $0\leq t\leq b$ }.

Given the three parameters b, γ and δ, this routine will compute and return probability p. It is used by BBASOB. It is also a special case of WIENER when $\gamma = -\gamma'$ and $\delta = -\delta'$ in WIENER.

WIENER (γ',δ',γ,δ,b,p): P{ $\gamma'+\delta' t \leq$ W(t) $\leq \gamma+\delta t$: $0\leq t\leq b$ }

Given five parameters (b, γ, δ, γ', and δ'), this routine will compute and return probability p. It has been utilized by GWPSAB.

GWPSAB (a,b,γ,δ,p): P{ |W(t)| $\leq \gamma+\delta t$: $a\leq t\leq b$ }.

With the four parameters a, b, γ and δ, this routine will compute and return probability p. FUNCTION WFUNC4 evaluates the values for the integrand function in (15); the probability argument in the integrand is evaluated by WIENER and QNG is called the numerical integration. It is used by BBASAB.

Routines for Brownian bridge process.

BBO1 (c,p): p = P{ B(t) $\leq$ c : $0\leq t\leq 1$ }.

For a given parameter c, this routine will compute and return probability p.

BBOB(b,c,p): p = P{ B(t) $\leq$ c : $0\leq t\leq b$ }.

For two parameters b and c, this routine will compute and return probability p. In this routine, we first set $b' := b/(1-b)$ and then evaluate GWPOB(b',c,c,p).

BBAB (a,b,c,p): p = P{ B(t) $\leq$ c : $a\leq t\leq b$ }.

Given the three parameters a, b and c, this routine will compute and return probability p. Similar to BBOB, we set $a' := a/(1-a)$, $b' = b/(1-b)$ and then evaluate GWPAB(a',b',c,c,p).

BBASO1(c,p): p= P{ |B(t)| $\leq$ c: $0\leq t\leq 1$ }.

For a given parameter c, this routine will compute and return probability p. It is the probability for the two-sided Kolmogorov-Smirnov statistics.

BBASOB(b,c,p): p = P{ |B(t)| $\leq$ c : $0\leq t\leq b$ }.

For the two parameters b and c, this routine will compute and return probability p. Set $b' := b/(1-b)$ and then call GWPSOB(b',c,c,p).

BBASAB(a,b,c,p): P{ |B(t)| $\leq$ c : $a\leq t\leq b$ }.

Given the three parameters a, b and c, this routine will compute and return the probability p. Set $a' := a/(1-a)$, $b' := b/(1-b)$ and then call GWPSAB(a',b',c,c,p).

Access routines: WWPR, WWAS, BBPR, BBAS.

These four routines have been added to permit easier access to the routines discussed in the preceding sections regardless of the values of the parameters a and b. In these routines, checks are made for simple errors such as $a>b$ and error code numbers are specified and returned to the user where errors occur.

Figure 2 presents the general flowcharts of WWPR, WWAS, BBPR and BBAS and illustrates how the routines in the previous sections are called in these routines.

WWPR(a,b,γ,p,error): $P\{ W(t) \leq \gamma : a \leq t \leq b \}$.

Depending upon the input value of a, either WPOB or WPAB is called in this routine. If $a < 0$ or $a > b$, error is set equal to 101.

If $a = b = 0$, then $p := 1$. If $a = b = 0$, the $p := \Phi(\gamma/\sqrt{b})$ where $\Phi(.)$ is the normal distribution function defined in section "Formulae for $p = P\{ W(t) \leq \gamma + \delta t : 0 \leq t \leq b \}$.

WWAS(a,b,γ,p,error): $P\{ |W(t)| \leq \gamma : a \leq t \leq b \}$.

Depending upon the input value of a, either WPASOB or WPASAB is called in this routine. If $a < 0$ or $a > b$, error is set equal to 101.

If $a = b = 0$, $p := 0$. If $a = b = 0$, then $p := 2\,\Phi(\gamma/\sqrt{b}) - 1$.

BBPR(a,b,c,p,error): $P\{ B(t) \leq c : a \leq t \leq b \}$.

Depending upon the values of a and b, it calls BBO1, BBOB or BBAB. Thus this routine can be used instead of BBO1, BBOB and BBAB by specifying the proper values of a and b. If $a < 0$, $b > 1$ or $a > b$, error is set equal to 102.

If $a=b=0$ or $a=b=1$, then $p := 1$. If $a=b$ ($= 0$ or 1), $p := \Phi(c/\sqrt{b(1-b)})$.

BBAS(a,b,c,p,error): $P\{ |B(t)| \leq c : a \leq t \leq b \}$.

Depending upon the values of a and b, this routine calls BBASO1, BBASOB or BBASAB, and thus it can be used instead of BBASO1, BBASOB and BBASAB by specifying the proper values for a and b.

If $a = b$ ($= 0$ or 1), then $p := 2\,\Phi\,(c/\sqrt{b(1-b)})$.

Inverse procedures: INVWW, INVBB.

In INVWW, depending upon the value of the option specified, one of the parameters a, b or γ in either

$p= P\{ W(t) \leq \gamma : a \leq t \leq b \}$ (if option = 1, 2, or 3) or

$p = P\{ |W(t)| \leq \gamma : a \leq t \leq b \}$ (if option = 4, 5, or 6)

is iteratively computed for a given p and the other two parameters (of a, b or γ) using the routines shown in Table 1 through WWPR and WWAS.

In INVBB, depending upon the value of the option specified, one of the parameters a, b or c in either

$p = P\{ B(t) \leq c : a \leq t \leq b \}$ (if option = 1, 2, or 3) or

$p = P\{ |B(t)| \leq c : a \leq t \leq b \}$ (if option = 4, 5, or 6)

is iteratively computed for a given p and the other two parameters (of a, b or c) using the routines shown in Table 1 through BBPR and BBAS.

INVWW(a,b,c,p,option).

As shown in Table 2, one of the parameters is computed in this routine depending upon the value specified for option for a given probability.

The maximum possible values for c and b are set equal to 8 and 64, respectively, instead of infinity. The iteration is stopped if the absolute difference between the probability specified and the probability computed using the estimated parameter is less than 0.00001.

Table 1. Names of routines in WIENER PACK with corresponding formulae and equation numbers used in text.

Routine name	Formula	Equation no.
WPOB	$P\{ W(t) \le c : 0\le t\le b,\ c>0,\ b<\infty \}$	3
WPAB	$P\{ W(t) \le c : a\le t\le b,\ c>0,\ a\ge 0,\ b<\infty \}$	5
WPASOB	$P\{ \lvert W(t)\rvert \le c : 0\le t\le b,\ c>0,\ b<\infty \}$	12
WPASAB	$P\{ \lvert W(t)\rvert \le c : a\le t\le b,\ c>0,\ a\ge 0,\ b<\infty \}$	16
GWPOB	$P\{ W(t) \le c+dt: 0\le t\le b,\ c>0,\ c+db\ge 0,\ b<\infty \}$	2
GWPAB	$P\{ W(t) \le c+dt: a\le t\le b,\ c>0,\ c+db\ge 0,\ a\ge 0,\ b<\infty \}$	4
GWPSOB	$P\{ \lvert W(t)\rvert \le c+dt: 0\le t\le b,\ c>0,\ c+db\ge 0,\ b<\infty \}$	9
GWPSAB	$P\{ \lvert W(t)\rvert \le c+dt: a\le t\le b,\ c>0,\ c+db\ge 0,\ a\ge 0,\ b<\infty \}$	15
WIEN1	$P\{ c' \le W(t) \le c: 0\le t\le b,\ c>0,\ c'<0,\ b<\infty \}$	10
WIENER	$P\{ c'+d't \le W(t) \le c+dt : 0\le t\le b,\ c>0,\ c+db\ge 0, c'<0,\ c'+d'b\le 0,\ b<\infty \{$	7
BB01	$P\{ B(t) \le c : 0\le t\le 1,\ c>0 \}$	21
BBOB	$P\{ B(t) \le c : 0\le t\le b,\ c>0,\ b\le 1 \}$	20
BBAB	$P\{ B(t) \le c : a\le t\le b,\ c>0,\ a\ge 0,\ b\le 1 \}$	22
BBASO1	$P\{ \lvert B(t)\rvert \le c : 0\le t\le 1,\ c>0 \}$	26
BBASOB	$P\{ \lvert B(t)\rvert \le c : 0\le t\le b,\ c>0,\ b\le 1 \}$	25
BBASAB	$P\{ \lvert B(t)\rvert \le c : a\le t\le b,\ c>0,\ a\ge 0,\ b\le 1 \}$	27

Table 2. Options for INVWW.

Option	formula	parameters specified	parameter computed
1	$P\{ W(t) \le c : a\le t\le b \}$	a, b	c
2	$P\{ W(t) \le c : a\le t\le b \}$	b, c	a
3	$P\{ W(t) \le c : a\le t\le b \}$	a, c	b
4	$P\{ \lvert W(t)\rvert \le c : a\le t\le b \}$	a, b	c
5	$P\{ \lvert W(t)\rvert \le c : a\le t\le b \}$	b, c	a
6	$P\{ \lvert W(t)\rvert \le c : a\le t\le b \}$	a, c	b

Table 3. Options for INVBB.

Option	formula	parameters specified	parameter computed
1	$P\{ B(t) \le c : a\le t\le b \}$	a, b	c
2	$P\{ B(t) \le c : a\le t\le b \}$	b, c	a
3	$P\{ B(t) \le c : a\le t\le b \}$	a, c	b
4	$P\{ \lvert B(t)\rvert \le c : a\le t\le b \}$	a, b	c
5	$P\{ \lvert B(t)\rvert \le c : a\le t\le b \}$	b, c	a
6	$P\{ \lvert B(t)\rvert \le c : a\le t\le b \}$	a, c	b

Through WWPR and WWAS, the routines discussed in section on the Wiener process are iteratively utilized for this routine.

INVBB.

As shown in Table 3, one of the parameters is computed in this routine depending upon the value specified for option for a given probability.

The maximum possible value for c is set equal to 8 instead of infinity. The iteration is stopped if the absolute difference between the probability specified and the probability computed using the estimated parameter is less than 0.00001.

Through BBPR and BBAS, the routines discussed in the section on the Brownian bridge process are iteratively utilized for this routine.

REFERENCES

Aly, E.-E. and Bleuer, S.
1983: Confidence bands for quantile-quantile plots; Carleton University, Technical Report 16, p. 23-46.

Aly, E.-E., Csörgö, M. and Horváth, L.
1985: Strong Approximations of the quantile process of the product-limit estimator; Journal of Multivariate Analysis, v. 16, no. 2, p. 185-210.

Anderson, T.W.
1960: A modification of the sequential probability ratio test to reduce the sample size; Annals of Mathematical Statistics, v. 31, p. 165-197.

Anderson, T.W. and Darling, D.A.
1952: Asymptotic theory of certain goodness of fit criteria based on stochastic processes; v. 23, p. 193-212.

Barr, D.R. and Davidson, T.
1973: A Kolomogorov-Smirnov test for censored samples; Technometrics, v. 15, no. 4, p. 737-757.

Breslow, N. and Crowley, J.
1974: A large sample study of the life table and product limit estimates under random censorship; Annals of Statistics, v. 2, p. 437-453.

Chung, C.F.
1986: Formulae for probabilities associated with Wiener and Brownian bridge processes; Carleton University, Technical report 78.

Csáki, E.
1981: Empirical distribution function; Selected Translations in mathematics, Statistics and Probability, v. 15, p. 229-317.

Csörgö, M.
1967: A new proof of some results of Rényi and the asymptotic distribution of the range of his Kolomogorov-Smirnov type random variables; Canadian Journal of Mathematics, v. 19, p. 550-558.

Csörgö, M.
1983: Quantile Processes with Statistical Applications; SIAM CBMS-NSF No. 42, 156 p.

Csörgö, M. and Révész, P.
1978: Strong approximations of the quantile process; Annals of Statistics, v. 4, p. 882-894.
1981: Strong Approximations in Probability and Statistics; Academic Press, 284p.

Csörgö, S. and Horváth L.
1981: On the Koziol-Green model for random censorship; Biometrika, v. 68, p. 391-401.
1985: Confidence bands from censored samples; Carleton University, Technical Report 44, p. 39-76.

Doob, J.L.
1949: A heuristic approach to the Kolmogorov-Smirnov theorems; Annals of Mathematical Statistics, v. 20, p. 393-403.

Feller, W.
1966: An Introduction to Probability Theory and its Applications, volume II; John Wiley, 626p.

Gillaspie, M.J. and Fisher, L.
1979: Confidence bands for the Kaplan-Meier survival curve estimate; Annals of Statistics, v. 7, no. 4, p. 920-924.

Hall, W.J. and Wellner J.A.
1980: Confidence bands for a survival curve from censored data; Biometrika v. 67. no. 1, p. 133-143.

Koziol, J.A. and Byar, D.P.
1975: Percentage points of the asymptotic distributions of one and two sample K-S statistics for truncated or censored data; Technometrics v. 17, p. 507-510.

Piessens, R., de Doncker-Kapenga, E., Uberhuber, C.W. and Kahaner, D.K.
1983: QUADPACK – A subroutine package for automatic integration; Springer-Verlag, 301p.

Rényi, A.
1953: On the theory of order statistics; Acta Mathematica [Budapest, Hungary], v. 4, p. 191-231.

Schumacher, M.
1984: Two-sample tests of Cramer-von-Mises- and Kolmogorov-Smirnov-type for randomly censored data; International Statistical Review, v. 52, no. 3, p. 263-281.

APPENDIX

Listing of the routines

In addition to the routines discussed in the text, a main program (MAIN) and four subroutines (WWIN, BBIN, INWW and INBB) are included in this listing. The main program and the four subroutines allow the user to access the routines discussed interactively.

An IBM-PC formatted diskette containing the source code of the routines is included in Geological Survey of Canada Open File 1426. A limited number of copies of this Open File can be purchased from

Geological Survey of Canada,
601 Booth Street, Ottawa, Ontario
Canada, K1A 0E8

```
      PROGRAM GSC
C
C              MAIN PROGRAM GSC
C
C    COMPUTING PROBABILITY FOR GIVEN PARAMETERS AND PARAMETERS
C    FOR GIVEN PROBABILITY OF WIENER PROCESS, BROWNIAN BRIDGE
C    PROCESS.
C
C               GSC PAPER 86-XX
C
      COMMON /FUNCT/DELTA(10)
C
      WRITE(*,999)' THIS PROGRAM HAS BEEN DEVELOPED AT THE '
      WRITE(*,999)'       GEOLOGICAL SURVEY OF CANADA.'
      WRITE(*,999)
10    WRITE(*,999)'  THE FOLLOWING OPTIONS ARE AVAILABLE:'
      WRITE(*,999)'  1 : COMPUTE PROBABILITY OF A WIENER  PROCESS'
      WRITE(*,999)'  2 : COMPUTE PROBABILITY OF A BROWNIAN BRIDGE'
      WRITE(*,999)'  3 : GIVEN PROBABILITY LEVEL, COMPUTE PARAMETER'
      WRITE(*,999)'            SELECTED OF A WIENER PROCESS'
      WRITE(*,999)'  4 : GIVEN PROBABILITY LEVEL, COMPUTE PARAMETER'
      WRITE(*,999)'            SELECTED OF A BROWNIAN BRIDGE'
      WRITE(*,999)'  5 : STOP.'
      WRITE(*,999)
20    WRITE(*,999)' ENTER OPTION (1-5)'
      READ(*,998)ISEL
999   FORMAT(A)
998   FORMAT(I1)
      IF(ISEL.LT.0.OR.ISEL.GT.5)GO TO 20
      GOTO (110,120,130,140,150),ISEL
110   CALL WWIN
      GO TO 10
120   CALL BBIN
      GO TO 10
130   CALL INWW
      GO TO 10
140   CALL INBB
      GO TO 10
150   STOP
      END
C=========================================================
C
      SUBROUTINE WWIN
C
C CALLING FOR COMPUTING PROBABILITY OF WIENER PROCESS
C
      CHARACTER IYES*1,ANSWER*1
      DATA IYES/'Y'/
10    WRITE(*,999)' ENTER 0 FOR P = PROB{    W(T)<C : A<T<B }'
      WRITE(*,999)'                      OR'
      WRITE(*,999)' ENTER 1 FOR P = PROB{ -C<W(T)<C : A<T<B }'
      READ(*,998)INN
      WRITE(*,999)' ENTER  C  ( > 0 ) , C = '
      READ(*,997)C
      WRITE(*,999)' ENTER  A  ( >=0 ) , A = '
      READ(*,997)A
      WRITE(*,999)' ENTER  B  ( > A ) , B = '
      READ(*,997)B
C
999   FORMAT(A)
998   FORMAT(I1)
997   FORMAT(F12.0)
996   FORMAT(A,I3)
995   FORMAT(A,F12.6)
994   FORMAT(A1)
      IF ( INN.EQ.0 ) THEN
        CALL WWPR( A,B,C,PROB,IER )
      ELSE
        CALL WWAS( A,B,C,PROB,IER )
      ENDIF
C
      IF ( IER.NE.0 ) THEN
        WRITE(*,996)' ERROR OCCURS, ERROR NUMBER = ',IER
      ELSE
        WRITE(*,995) ' GIVEN PARAMETERS, PROBABILITY = ',PROB
      ENDIF
      WRITE(*,999)' WOULD YOU LIKE TO TRY ANOTHER PARAMETERS?(Y OR N)'
      READ(*,994)ANSWER
      IF( ANSWER.EQ.IYES ) GO TO 10
      RETURN
      END
C=========================================================
C
      SUBROUTINE BBIN
C
C CALLING FOR COMPUTING PROBABILITY OF BROWIAN BRIDGE PROCESS
C
      CHARACTER IYES*1,ANSWER*1
      DATA IYES/'Y'/
10    WRITE(*,999)' ENTER 0 FOR P = PROB{    B(T)<C : A<T<B }'
      WRITE(*,999)'                      OR'
      WRITE(*,999)' ENTER 1 FOR P = PROB{ -C<B(T)<C : A<T<B }'
      READ(*,998)INN
      WRITE(*,999)' ENTER  C  (   > 0   ) , C = '
      READ(*,997)C
      WRITE(*,999)' ENTER  A  ( 0<=A< B ) , A = '
      READ(*,997)A
      WRITE(*,999)' ENTER  B  ( A< B<=1 ) , B = '
      READ(*,997)B
C
999   FORMAT(A)
998   FORMAT(I1)
997   FORMAT(F12.0)
996   FORMAT(A,I3)
```

```
995   FORMAT(A,F12.6)
994   FORMAT(A1)
      IF ( INN.EQ.0 ) THEN
        CALL BBPR( A,B,C,PROB,IER )
      ELSE
        CALL BBAS( A,B,C,PROB,IER )
      ENDIF
C
      IF ( IER.NE.0 ) THEN
        WRITE(*,996)' ERROR OCCURS, ERROR NUMBER = ',IER
      ELSE
        WRITE(*,995) ' GIVEN PARAMETERS, PROBABILITY = ',PROB
      ENDIF
      WRITE(*,999)' WOULD YOU LIKE TO TRY ANOTHER PARAMETERS?(Y OR N)'
      READ(*,995)ANSWER
      IF( ANSWER.EQ.IYES ) GO TO 10
      RETURN
      END
C=========================================================
C
      SUBROUTINE INWW
C
C CALLING FOR COMPUTING PARAMETER OF WIENER PROCESS, FOR
C GIVEN PROBABILITY LEVEL
C
      CHARACTER IYES*1,ANSWER*1
      DATA IYES/'Y'/
10    WRITE(*,999)' ENTER 0 FOR P = PROB{    W(T)<C : A<T<B }'
      WRITE(*,999)º                      OR'
      WRITE(*,999)' ENTER 1 FOR P = PROB{ -C<W(T)<C : A<T<B }'
      WRITE(*,999)º                      OR'
      WRITE(*,999)' ENTER 2 FOR P = PROB{ W(T)<C+DT : A<T<B }'
      WRITE(*,999)'                      OR'
      WRITE(*,999)' ENTER 3 FOR P = P{-C-DT<W(T)<C+DT:A<T<B }'
      READ(*,998)INN
C
999   FORMAT(A)
998   FORMAT(I1)
997   FORMAT(F12.0)
996   FORMAT(A,I3)
995   FORMAT(A,F12.6)
994   FORMAT(A1)
      IF ( INN.EQ.0 ) THEN
        WRITE(*,999)' TO CALCULATE  C , ENTER OPTION = 1'
        WRITE(*,999)'               A                  2'
        WRITE(*,999)'               B                  3'
        WRITE(*,999)'  OPTION = '
      ENDIF
      IF ( INN.EQ.1 ) THEN
        WRITE(*,999)' TO CALCULATE  C , ENTER OPTION = 4'
        WRITE(*,999)'               A                  5'
        WRITE(*,999)'               B                  6'
        WRITE(*,999)'  OPTION = '
      ENDIF
      IF ( INN.EQ.2 ) THEN
        WRITE(*,999)' TO CALCULATE  C , ENTER OPTION = 7'
        WRITE(*,999)'               D                  8'
        WRITE(*,999)'  OPTION = '
      ENDIF
      IF ( INN.EQ.3 ) THEN
        WRITE(*,999)' TO CALCULATE  C , ENTER OPTION = 9'
        WRITE(*,999)'               D                  10'
        WRITE(*,999)'  OPTION = '
      ENDIF
      READ(*,998)IOP
      WRITE(*,999)' ENTER  P  ( 0<P<1 ) , P = '
      READ(*,997)P
      IF ( IOP.EQ.1.OR.IOP.EQ.4.OR.IOP.GE.7 ) THEN
        WRITE(*,999)' ENTER  A  ( >=0 ) , A = '
        READ(*,997)A
        WRITE(*,999)' ENTER  B  ( > A ) , B = '
        READ(*,997)B
        IF (IOP.EQ.7.OR.IOP.EQ.9) THEN
          WRITE(*,999)' ENTER  D , D = '
          READ(*,997)D
        ELSE
          WRITE(*,999)' ENTER  C , C = '
          READ(*,997)C
        ENDIF
      ELSE
        IF ( IOP.EQ.2.OR.IOP.EQ.5 ) THEN
          WRITE(*,999)' ENTER  C  ( > 0 ) , C = '
          READ(*,997)C
          WRITE(*,999)' ENTER  B  ( > A ) , B = '
          READ(*,997)B
        ELSE
          WRITE(*,999)' ENTER  C  ( > 0 ) , C = '
          READ(*,997)C
          WRITE(*,999)' ENTER  A  ( >=0 ) , A = '
          READ(*,997)A
        ENDIF
      ENDIF
C
      CALL INVWW( A,B,C,D,P,IOP,IER )
C
      IF ( IER.NE.0 ) THEN
        WRITE(*,996)' ERROR OCCURS, ERROR NUMBER = ',IER
      ELSE
        IF ( IOP.EQ.1.OR.IOP.EQ.4.OR.IOP.EQ.7.OR.IOP.EQ.9 ) THEN
          PARA = C
        ELSE
          IF ( IOP.EQ.2.OR.IOP.EQ.5 ) THEN
            PARA = A
          ELSE
```

```
          IF ( IOP.EQ.3.OR.IOP.EQ.6 ) THEN
            PARA = B
          ELSE
            PARA = D
          ENDIF
        ENDIF
      ENDIF
      WRITE(*,995) ' GIVEN PROBABILITY, PARAMETER = ',PARA
    ENDIF
    WRITE(*,999)' WOULD YOU LIKE TO TRY ANOTHER PARAMETERS?(Y OR N)'
    READ(*,994)ANSWER
    IF( ANSWER.EQ.IYES ) GO TO 10
    RETURN
    END
C============================================================
C
      SUBROUTINE INBB
C
C CALLING FOR COMPUTING PARAMETER OF BROWNIAN BRIDGE PROCESS,
C FOR GIVEN PROBABILITY LEVEL
C
      CHARACTER IYES*1,ANSWER*1
      DATA IYES/'Y'/
10    WRITE(*,999)' ENTER 0 FOR P = PROB{    B(T)<C : A<T<B }'
      WRITE(*,999)'                 OR'
      WRITE(*,999)' ENTER 1 FOR P = PROB{ -C<B(T)<C : A<T<B }'
      READ(*,998)INN
C
999   FORMAT(A)
998   FORMAT(I1)
997   FORMAT(F12.0)
996   FORMAT(A,I3)
995   FORMAT(A,F12.6)
994   FORMAT(A1)
      IF ( INN.EQ.0 ) THEN
        WRITE(*,999)' TO CALCULATE  C , ENTER OPTION = 1'
        WRITE(*,999)'               A                  2'
        WRITE(*,999)'               B                  3'
        WRITE(*,999)'  OPTION = '
      ELSE
        WRITE(*,999)' TO CALCULATE  C , ENTER OPTION = 4'
        WRITE(*,999)'               A                  5'
        WRITE(*,999)'               B                  6'
        WRITE(*,999)'  OPTION = '
      ENDIF
      READ(*,998)IOP
      WRITE(*,999)' ENTER  P  ( 0<P<1 ) , P = '
      READ(*,997)P
      IF ( IOP.EQ.1.OR.IOP.EQ.4) THEN
        WRITE(*,999)' ENTER  A  ( >=0 ) , A = '
        READ(*,997)A
        WRITE(*,999)' ENTER  B  ( > A ) , B = '
        READ(*,997)B
      ELSE
        IF ( IOP.EQ.2.OR.IOP.EQ.5 ) THEN
          WRITE(*,999)' ENTER  C  ( > 0 ) , C = '
          READ(*,997)C
          WRITE(*,999)' ENTER  B  ( > A ) , B = '
          READ(*,997)B
        ELSE
          WRITE(*,999)' ENTER  C  ( > 0 ) , C = '
          READ(*,997)C
          WRITE(*,999)' ENTER  A  ( >=0 ) , A = '
          READ(*,997)A
        ENDIF
      ENDIF
C
      CALL INVBB( A,B,C,P,IOP,IER )
C
      IF ( IER.NE.0 ) THEN
        WRITE(*,996)' ERROR OCCURS, ERROR NUMBER = ',IER
      ELSE
        IF ( IOP.EQ.1.OR.IOP.EQ.4 ) THEN
          PARA = C
        ELSE
          IF ( IOP.EQ.2.OR.IOP.EQ.5 ) THEN
            PARA = A
          ELSE
            PARA = B
          ENDIF
        ENDIF
        WRITE(*,995) ' GIVEN PROBABILITY, PARAMETER = ',PARA
      ENDIF
      WRITE(*,999)' WOULD YOU LIKE TO TRY ANOTHER PARAMETERS?(Y OR N)'
      READ(*,994)ANSWER
      IF( ANSWER.EQ.IYES ) GO TO 10
      RETURN
      END
C============================================================
C
      SUBROUTINE FLOW ( UFLOW,OFLOW )
C************************************************************
C
C  THIS ROUTINE PROVIDES THE MACHINE CONSTANTS OF THE
C  SMALLEST AND LARGEST NUMBERS.
C
C  PARAMETERS:
C    OUT    UFLOW  :  THE SMALLEST POSITIVE NUMBER
C           OFLOW  :  THE LARGEST  POSITIVE NUMBER
C
C************************************************************
      UFLOW = 1.0E-37
      OFLOW = 1.0E+38
      RETURN
      END
C============================================================
C
      REAL FUNCTION PHI(X)
C************************************************************
C
C  1.  PHI   - FORTRAN 77 REAL FUNCTION
C  2.  IT COMPUTES THE PROBABILITY OF STANDARD NORMAL DISTRIBUTION
C      FUNCTION:
C      PHI(X) = PROB{ N < X } WHERE N ~ NORMAL( 0, 1 )
C  3.  PARAMETERS:
C        IN    X      :  SEE THE ABOVE FORMULA
C        OUT   PHI    :  PROBABILITY
C  4.  SUBROUTINES OR FUNCTIONS REQUIRED:
C        NONE
C  5.  REFERENCES:
C        ABRAMOWITZ AND STEGUN ( 1964, (26.2.17) OF P.932)
C
C************************************************************
      DIMENSION B(5)
      DATA B/ 0.319381530, -0.356563782,  1.781477937,
     C        -1.821255978,  1.330274429/
      DATA PP/0.231641900/, PH/3.14159265359/
C
      IN=0
      IF(X.LT.0.0)THEN
        IN=1
        X=-X
      ENDIF
      M=5
      CALL FLOW ( UFLOW, OFLOW )
      EUFLOW = -ALOG( UFLOW )
      XXM = X*X/2.0
      IF(XXM.LT.EUFLOW)THEN
        T=1.0/(1.0+PP*X)
        ZX=(EXP(-XXM))/SQRT(2.0*PH)
        TT=1.
        BT=0.
        DO 10 I=1,M
          TT=TT*T
          BT=BT+B(I)*TT
10      CONTINUE
      ELSE
        ZX=0.
        BT=0.
      ENDIF
      PHI=1.0-ZX*BT
      IF(IN.EQ.1)THEN
        PHI=1.0-PHI
        X=-X
      ENDIF
      RETURN
      END
C============================================================
C
       SUBROUTINE WWPR(A,B,C,PROB,IER)
C****************************************************************
C
C  1.  WWPR   - FORTRAN 77 SUBROUTINE
C
C  2.  THE ROUTINE CALCULATES AN APPROXIMATION RESULT OF THE
C      PROBABILITY OF THE WIENER PROCESS OF THE FOLLOWING FORM:
C
C      PROB = P { W ( T ) < C : A < T < B }
C
C  3.  PARAMETERS:
C        IN    A     :  SEE THE ABOVE FORMULA
C              B     :       "
C              C     :       "
C        OUT   PROB  :       "
C              IER   :  ERROR AND IER > 0, IF AN ERROR OCCURS.
C
C  4.  SUBROUTINES OR FUNCTIONS REQUIRED:
C        - WWOB - PHI
C        - WWAB - QNG - QMACO
C               - WFUNC1 - PHI
C
C****************************************************************
C
C * * * * * CHECK ON THE VALIDITY OF PARAMETERS
C
       IF ( A.LT.0.0 .OR. A.GT.B ) THEN
         IER=101
         PROB=0.0
       ELSE
         IER=0
         IF ( A.EQ.B ) THEN
C
C * * * * * SET PROB = PHI( C/SQRT(B) ), IF   A = B.
C
           IF ( B.EQ.0. ) THEN
             PROB = 1.0
           ELSE
             BB = SQRT( B )
             PROB = PHI ( C / BB )
           ENDIF
         ELSE
           IF ( A.EQ.0.0 ) THEN
C
C * * * * * CALL WPOB TO COMPUTE    P { W(T) < C : 0 < T < B }
C
             CALL WPOB( B,C,PROB )
           ELSE
C
```

```
C * * * * * CALL WPAB TO COMPUTE    P { W(T) < C : A < T < B }
C
            CALL WPAB( A,B,C,PROB )
          ENDIF
        ENDIF
        IF ( PROB.GE.100. ) THEN
          IER = PROB
          PROB = 0.
        ENDIF
      ENDIF
      RETURN
      END
C==============================================================
C
      SUBROUTINE WWAS(A,B,C,PROB,IER)
C**************************************************************
C
C  1.  WWAS   - FORTRAN 77 SUBROUTINE
C
C  2.  THE ROUTINE CALCULATES AN APPROXIMATION RESULT OF THE
C      PROBABILITY OF THE WIENER PROCESS OF THE FOLLOWING FORM:
C
C      PROB = P { -C < W ( T ) < C : A < T < B }
C
C  3.  PARAMETERS:
C        IN    A     :  SEE THE ABOVE FORMULA
C              B     :          "
C              C     :          "
C        OUT   PROB  :          "
C              IER   :  ERROR AND IER > 0, IF AN ERROR OCCURS.
C
C  4.  SUBROUTINES OR FUNCTIONS REQUIRED:
C        - WWASOB - FLOW
C        - WWASAB - QNG - QMACO
C                 - WFUNC2 - WIEN1 - PHI
C
C*******************************************************************
C
C * * * * * CHECK ON THE VALIDITY OF PARAMETERS
C
      IF ( A.LT.0.0 .OR. A.GT.B ) THEN
        IER=101
        PROB=0.0
      ELSE
        IER=0
        IF ( A.EQ.B ) THEN
C
C * * * * * SET PROB = 2*PHI( C/SQRT(B) ) - 1 , IF   A = B.
C
          IF ( B.EQ.0. ) THEN
            PROB = 1.0
            BB = SQRT( B )
            PROB = 2.0 * PHI( C/BB ) - 1.0
          ENDIF
        ELSE
          IF ( A.EQ.0.0 ) THEN
C
C * * * * * CALL WPASOB TO COMPUTE    P { -C < W(T) < C : 0 < T < B }
C
            CALL WPASOB( B,C,PROB )
          ELSE
C
C * * * * * CALL WPASAB TO COMPUTE    P { -C < W(T) < C : A < T < B }
C
            CALL WPASAB( A,B,C,PROB )
          ENDIF
        ENDIF
        IF ( PROB.GE.100. ) THEN
          IER = PROB
          PROB = 0.
        ENDIF
      ENDIF
      RETURN
      END
C==============================================================
C
      SUBROUTINE BBPR(A,B,C,PROB,IER)
C**************************************************************
C
C  1.  BBPR   - FORTRAN 77 SUBROUTINE
C
C  2.  THE ROUTINE CALCULATES AN APPROXIMATION RESULT OF THE
C      PROBABILITY OF THE BROWNIAN BRIDGE PROCESS OF THE FOLLOWING FORM:
C
C      PROB = P { B ( T ) < C : A < T < B }
C
C  3.  PARAMETERS:
C        IN    A     :  SEE THE ABOVE FORMULA
C              B     :          "
C              C     :          "
C        OUT   PROB  :          "
C              IER   :  ERROR AND IER > 0, IF AN ERROR OCCURS.
C
C  4.  SUBROUTINES OR FUNCTIONS REQUIRED:
C        - BB01
C        - BBOB - GWPOB - PHI
C        - BBAB - GWPAB - QNG - QMACO
C                       - WFUNC3 - GWPOB - PHI
C
C*******************************************************************
C
C * * * * * CHECK ON THE VALIDITY OF PARAMETERS
C
      IF ( A.LT.0.0 .OR. B.GT.1.0 .OR. A.GT.B ) THEN
        IER=102
        PROB=0.0
      ELSE
        IER=0
        IF ( A.EQ.B ) THEN
C
C * * * * * SET PROB = PHI( C/SQRT(B*(1-B)) ), IF   A = B.
C
          IF ( B.EQ.0.0.OR.B.EQ.1.0 ) THEN
            PROB = 1.0
          ELSE
            BB = SQRT( B*(1.0-B) )
            PROB = PHI( C/BB )
          ENDIF
        ELSE
          IF ( A.EQ.0.0 ) THEN
            IF ( B.EQ.1.0 ) THEN
C
C * * * * * CALL BB01 TO COMPUTE    P { B(T) < C : 0 < T < 1 }
C
              CALL BB01( C,PROB )
            ELSE
C
C * * * * * CALL BBOB TO COMPUTE    P { B(T) < C : 0 < T < B }
C
              CALL BBOB( B,C,PROB )
            ENDIF
          ELSE
            IF ( B.EQ.1.0 ) THEN
              BB = 1.0 - A
C
C * * * * * CALL BBOB TO COMPUTE    P { B(T) < C : A < T < 1 }
C
              CALL BBOB( BB,C,PROB )
            ELSE
C
C * * * * * CALL BBAB TO COMPUTE    P { B(T) < C : A < T < B }
C
              CALL BBAB( A,B,C,PROB )
            ENDIF
          ENDIF
        ENDIF
        IF ( PROB.GE.100. ) THEN
          IER = PROB
          PROB = 0.
        ENDIF
      ENDIF
      RETURN
      END
C==============================================================
C
      SUBROUTINE BBAS(A,B,C,PROB,IER)
C**************************************************************
C
C  1.  BBAS   - FORTRAN 77 SUBROUTINE
C
C  2.  THE ROUTINE CALCULATES AN APPROXIMATION RESULT OF THE
C      PROBABILITY OF THE BROWNIAN BRIDGE PROCESS OF THE FOLLOWING FORM:
C
C      PROB = P { -C < B ( T ) < C : A < T < B }
C
C  3.  PARAMETERS:
C        IN    A     :  SEE THE ABOVE FORMULA
C              B     :          "
C              C     :          "
C        OUT   PROB  :          "
C              IER   :  ERROR AND IER > 0, IF AN ERROR OCCURS.
C
C  4.  SUBROUTINES OR FUNCTIONS REQUIRED:
C        - BBAS01 - FLOW
C        - BBASOB - GWPSOB - FLOW
C                          - PHI
C        - BBASAB - GWPSAB - QNG - QMACO
C                          - WFUNC4 - WIENER - FLOW
C                                            - PHI
C
C*******************************************************************
C
C * * * * * CHECK ON THE VALIDITY OF PARAMETERS
C
      IF ( A.LT.0.0 .OR. B.GT.1.0 .OR. A.GT.B ) THEN
        IER=102
        PROB=0.0
      ELSE
        IER=0
        IF ( A.EQ.B ) THEN
C
C * * * * * SET PROB = 2*PHI ( C/SQRT(B*(1-B)) ) - 1 , IF   A = B.
C
          IF ( B.EQ.0.0.OR.B.EQ.1.0 ) THEN
            PROB = 1.0
          ELSE
            BB = SQRT( B*(1.0-B) )
            PROB = 2.0*PHI( C/BB ) - 1.0
          ENDIF
        ELSE
          IF ( A.EQ.0.0 ) THEN
            IF ( B.EQ.1.0 ) THEN
C
C * * * * * CALL BBAS01 TO COMPUTE    P { -C < B(T) < C : 0 < T < 1 }
C
              CALL BBAS01( C,PROB )
            ELSE
```

```
C
C * * * * * CALL BBASOB TO COMPUTE    P { -C < B(T) < C : 0 < T < B }
C
              CALL BBASOB( B,C,PROB )
            ENDIF
          ELSE
            IF ( B.EQ.1.0 ) THEN
              BB = 1.0 - A
C
C * * * * * CALL BBASOB TO COMPUTE    P { -C < B(T) < C : A < T < 1 }
C
              CALL BBASOB( BB,C,PROB )
            ELSE
C
C * * * * * CALL BBASAB TO COMPUTE    P { -C < B(T) < C : A < T < B }
C
              CALL BBASAB( A,B,C,PROB )
            ENDIF
          ENDIF
        ENDIF
        IF ( PROB.GE.100. ) THEN
          IER = PROB
          PROB = 0.
        ENDIF
      ENDIF
      RETURN
      END
C============================================================
C
      SUBROUTINE INVWW( A,B,C,D,P,IOPT,IER )
C************************************************************
C
C  1.  INVWW   - FORTRAN 77 SUBROUTINE
C
C  2.  THE ROUTINE CALCULATES AN APPROXIMATION RESULT OF THE PARAMETER
C      SELECTED, DEPENDING UPON THE INPUT PARAMETER IOPT, OF THE WIENER
C      PROCESS.  WHEN IOPT = 1, 2 OR 3, FOR GIVEN PROBABILITY P, FOR,
C
C      P = PROB { W ( T ) < C : A < T < B }
C
C      IT COMPUTES   C  , IF  IOPT = 1
C         "          A  ,    "    = 2
C         "          B  ,    "    = 3
C
C      WHEN IOPT = 4, 5 OR 6, FOR,
C
C      P = PROB { -C < W ( T ) < C : A < T < B }
C
C      IT COMPUTES   C  ,  IF IOPT = 4
C         "          A  ,    "    = 5
C         "          B  ,    "    = 6
C
C      WHEN IOPT = 7 OR 8, FOR,
C
C      P = PROB { W ( T ) < C+DT : A < T < B }
C
C      IT COMPUTES   C  , IF  IOPT = 7
C         "          D  ,    "    = 8
C
C      WHEN IOPT = 9 OR 10, FOR,
C
C      P = PROB { -C-DT < W ( T ) < C+DT : A < T < B }
C
C      IT COMPUTES   C  ,  IF IOPT = 9
C         "          D  ,    "    = 10
C
C  3.  PARAMETERS:
C        IN    IOPT      :  SEE THE ABOVE
C              P         :  GIVEN PROBABILITY LEVEL
C              C         :  IF IOPT =    2, 3,    5, 6,    8,    10
C              D         :           =                  7,    9
C              A         :           = 1,    3, 4,    6, 7, 8, 9, 10
C              B         :           = 1, 2,    4, 5,    7, 8, 9, 10
C        OUT   C         :  IF IOPT = 1, 4, 7, 9
C              D         :           = 8, 10
C              A         :           = 2, 5
C              B         :           = 3, 6
C              IER       :  ERROR CODE AND IER > 0, IF AN ERROR OCCURS.
C
C  4.  SUBROUTINES OR FUNCTIONS REQUIRED:
C        - WWPR - WWOB - PHI
C               - WWAB - QNG - QMACO
C                      - WFUNC1 - PHI
C        - WWAS - WWASOB - FLOW
C               - WWASAB - QNG - QMACO
C                        - WFUNC2 - WIEN1 - PHI
C
C************************************************************
      DATA XMAX/8.0/, TOL/0.000001/
C
C * * * * * CHECK ON THE VALIDITY OF PARAMETERS
C
      IF ( IOPT.LT.1 .OR. IOPT.GT.10 ) THEN
        IER=111
        RETURN
      ENDIF
      GOTO ( 10,20,30,40,50,60,70,75,80,85 ),IOPT
C
C * * * * * SET UP APPROPRIATE INITIAL VALUES
C
10      VMAX=XMAX
        VMIN=0.
        PROB=0.0
        CALL WWPR( A,B,VMAX,PROBX,IER )
        GO TO 90
20      VMAX = B
        VMIN=0.0
        CALL WWPR( VMIN,B,C,PROB,IER )
        CALL WWPR( B,B,C,PROBX,IER )
        GO TO 90
30      VMAX=A
        VMIN=64.
        CALL WWPR( A,VMIN,C,PROB,IER )
        CALL WWPR( A,A,C,PROBX,IER )
        GO TO 90
40      VMAX=XMAX
        VMIN=0.
        PROB=0.
        CALL WWAS( A,B,VMAX,PROBX,IER )
        GO TO 90
50      VMAX=B
        VMIN=0.
        CALL WWAS( VMIN,B,C,PROB,IER )
        CALL WWAS( B,B,C,PROBX,IER )
        GO TO 90
60      VMAX=A
        VMIN=128.
        CALL WWAS( A,VMIN,C,PROB,IER )
        CALL WWAS( A,A,C,PROBX,IER )
70      VMAX=XMAX
        VMIN=0.
        IF ( A.EQ.0.0 ) THEN
          CALL GWPOB( B,VMIN,D,PROB )
          CALL GWPOB( B,VMAX,D,PROBX )
        ELSE
          CALL GWPAB( A,B,VMIN,D,PROB )
          CALL GWPAB( A,B,VMAX,D,PROBX )
        ENDIF
        GO TO 90
75      VMAX=XMAX
        VMIN=0.
        IF ( A.EQ.0.0 ) THEN
          CALL GWPOB( B,C,VMIN,PROB )
          CALL GWPOB( B,C,VMAX,PROBX )
        ELSE
          CALL GWPAB( A,B,C,VMIN,PROB )
          CALL GWPAB( A,B,C,VMAX,PROBX )
        ENDIF
        GO TO 90
80      VMAX=XMAX
        VMIN=0.
        IF ( A.EQ.0.0 ) THEN
          CALL GWPSOB( B,VMIN,D,PROB )
          CALL GWPSOB( B,VMAX,D,PROBX )
        ELSE
          CALL GWPSAB( A,B,VMIN,D,PROB )
          CALL GWPSAB( A,B,VMAX,D,PROBX )
        ENDIF
        GO TO 90
85      VMAX=XMAX
        VMIN=0.
        IF ( A.EQ.0.0 ) THEN
          CALL GWPSOB( B,C,VMIN,PROB )
          CALL GWPSOB( B,C,VMAX,PROBX )
        ELSE
          CALL GWPSAB( A,B,C,VMIN,PROB )
          CALL GWPSAB( A,B,C,VMAX,PROBX )
        ENDIF
90    IF ( IER.NE.0 ) RETURN
C
      IF ( ABS(PROB-P).LT.TOL) THEN
         GOTO ( 310,320,330,310,320,330,310,340,310,340 ), IOPT
310         C = VMIN
            GO TO 390
320         A = VMIN
            GO TO 390
330         B = VMIN
            GO TO 390
340         D = VMIN
            GO TO 390
      ELSE
         IF (  ABS(PROBX-P).LT.TOL) THEN
            GOTO ( 360,370,380,360,370,380,360,385,360,385), IOPT
360            C = VMAX
               GO TO 390
370            A = VMAX
               GO TO 390
380            B = VMAX
               GO TO 390
385            D = VMAX
               GO TO 390
         ENDIF
      ENDIF
C
      IF ( PROB.GT.P.OR.PROBX.LT.P ) THEN
        IER=131
        RETURN
      ENDIF
C
C * * * * * ITERATIVE PROCEDURE STARTS FROM HERE
C
      ITER = 1
100   VNEW = VMIN + (VMAX-VMIN)/2.0
      WRITE(*,992)ITER
992   FORMAT('+ THE COMPUTATIONS ARE IN PROGRESS, ITERATION NO. =',I5)
      ITER = ITER + 1
C
      GOTO ( 110,120,130,140,150,160,170,175,180,185 ), IOPT
```

```
C
110     CALL WWPR( A,B,VNEW,PROBN,IER )
        GO TO 190
120     CALL WWPR( VNEW,B,C,PROBN,IER )
        GO TO 190
130     CALL WWPR( A,VNEW,C,PROBN,IER )
        GO TO 190
140     CALL WWAS( A,B,VNEW,PROBN,IER )
        GO TO 190
150     CALL WWAS( VNEW,B,C,PROBN,IER )
        GO TO 190
160     CALL WWAS( A,VNEW,C,PROBN,IER )
        GO TO 190
170     IF ( A.EQ.0.0 ) THEN
          CALL GWPOB( B,VNEW,D,PROBN )
        ELSE
          CALL GWPAB( A,B,VNEW,D,PROBN )
        ENDIF
        GO TO 190
175     IF ( A.EQ.0.0 ) THEN
          CALL GWPOB( B,C,VNEW,PROBN )
        ELSE
          CALL GWPAB( A,B,C,VNEW,PROBN )
        ENDIF
        GO TO 190
180     IF ( A.EQ.0.0 ) THEN
          CALL GWPSOB( B,VNEW,D,PROBN )
        ELSE
          CALL GWPSAB( A,B,VNEW,D,PROBN )
        ENDIF
        GO TO 190
185     IF ( A.EQ.0.0 ) THEN
          CALL GWPSOB( B,C,VNEW,PROBN )
        ELSE
          CALL GWPSAB( A,B,C,VNEW,PROBN )
        ENDIF
190   IF ( IER.NE.0 ) RETURN
C
C * * * * * IF THE DIFFERENCE IS LESS THAN THE GIVEN TORELANCE
C                 THEN STOP THE ITERATIONS
C
      IF ( ABS(PROBN-P).GT.TOL ) THEN
        IF ( P.GT.PROBN ) THEN
          VMIN = VNEW
        ELSE
          VMAX = VNEW
        ENDIF
        GO TO 100
      ELSE
        GOTO ( 210,220,230,210,220,230,210,240,210,240 ), IOPT
210       C = VNEW
          GO TO 290
220       A = VNEW
          GO TO 290
230       B = VNEW
          GO TO 290
240       D = VNEW
290     CONTINUE
      ENDIF
390   RETURN
      END
C==========================================================
C
      SUBROUTINE INVBB( A,B,C,P,IOPT,IER )
C**********************************************************
C
C  1.  INVWW   -FORTRAN 77 SUBROUTINE
C
C  2.  THE ROUTINE CALCULATES AN APPROXIMATION RESULT OF THE PARAMETER
C      SELECTED, DEPENDING UPON THE INPUT PARAMETER IOPT, OF THE
C      BROWNIAN BRIDGE PROCESS.
C      WHEN IOPT = 1, 2 OR 3, FOR GIVEN PROBABILITY P,
C
C      P = PROB { B ( T ) < C : A < T < B }
C
C      IT COMPUTES   C  , IF  IOPT = 1
C        "           A  ,     "    = 2
C        "           B  ,     "    = 3
C
C      WHEN IOPT = 4, 5 OR 6, FOR,
C
C      P = PROB { -C < B ( T ) < C : A < T < B }
C
C      IT COMPUTES   C  ,  IF IOPT = 4
C        "           A  ,     "    = 5
C        "           B  ,     "    = 6
C
C  3.  PARAMETERS:
C        IN    IOPT     :  SEE THE ABOVE
C              P        :  GIVEN PROBABILITY LEVEL
C              C        :  IF IOPT =    2, 3,    5 OR 6
C              A        :            = 1,    3, 4   OR 6
C              B        :            = 1, 2,     4 OR 5
C        OUT   C        :  IF IOPT = 1 OR 4
C              A        :            = 2 OR 5
C              B        :            = 3 OR 6
C              IER      :  ERROR CODE AND IER > 0, IF AN ERROR OCCURS.
C
C  4.  SUBROUTINES OR FUNCTIONS REQUIRED:
C        - BBPR - BB01
C               - BBOB - GWPOB - PHI
C               - BBAB - GWPAB - QNG - QMACO
C                              - WFUNC3 - GWPOB - PHI
C        - BBAS - BBAS01 - FLOW
C               - BBASOB - GWPSOB - FLOW
C                                 - PHI
C               - BBASAB - GWPSAB - QNG - QMACO
C                                 - WFUNC4 - WIENER - FLOW
C                                                   - PHI
C******************************************************************
      DATA XMAX/8.0/, TOL/0.000001/
C
C * * * * * CHECK ON THE VALIDITY OF PARAMETERS
C
      IF ( IOPT.LT.1 .OR. IOPT.GT.6 ) THEN
        IER=112
        RETURN
      ENDIF
      GOTO ( 10,20,30,40,50,60 ),IOPT
C
C * * * * * SET UP APPROPRIATE INITIAL VALUES
C
10      VMAX=XMAX
        VMIN=0.
        PROB=0.
        CALL BBPR( A,B,VMAX,PROBX,IER )
        GO TO 90
20      VMAX=B
        VMIN=0.
        CALL BBPR( VMIN,B,C,PROB,IER )
        CALL BBPR( B,B,C,PROBX,IER )
        GO TO 90
30      VMAX=A
        VMIN=1.
        CALL BBPR( A,VMIN,C,PROB,IER )
        CALL BBPR( A,A,C,PROBX,IER )
        GO TO 90
40      VMAX=XMAX
        VMIN=0.
        PROB=0.
        CALL BBAS( A,B,VMAX,PROBX,IER )
        GO TO 90
50      VMAX=B
        VMIN=0.
        CALL BBAS( VMIN,B,C,PROB,IER )
        CALL BBAS( B,B,C,PROBX,IER )
        GO TO 90
60      VMAX=A
        VMIN=1.
        CALL BBAS( A,VMIN,C,PROB,IER )
        CALL BBAS( A,A,C,PROBX,IER )
90    IF ( IER.NE.0 ) RETURN
C
      IF ( ABS(PROB-P).LT.TOL) THEN
         GOTO ( 310,320,330,310,320,330), IOPT
310         C = VMIN
            GO TO 390
320         A = VMIN
            GO TO 390
330         B = VMIN
            GO TO 390
      ELSE
         IF (  ABS(PROBX-P).LT.TOL) THEN
            GOTO ( 360,370,380,360,370,380), IOPT
360            C = VMAX
               GO TO 390
370            A = VMAX
               GO TO 390
380            B = VMAX
               GO TO 390
         ENDIF
      ENDIF
C
      IF ( PROB.GT.P.OR.PROBX.LT.P ) THEN
        IER=132
        RETURN
      ENDIF
C
C * * * * * ITERATIVE PROCEDURE STARTS FROM HERE
C
      ITER = 1
100   VNEW = VMIN + (VMAX-VMIN)/2.0
      WRITE(*,992)ITER
992   FORMAT('+ THE COMPUTATIONS ARE IN PROGRESS, ITERATION NO. =',I5)
      ITER = ITER + 1
C
      GOTO ( 110,120,130,140,150,160 ), IOPT
C
110     CALL BBPR( A,B,VNEW,PROBN,IER )
        GO TO 190
120     CALL BBPR( VNEW,B,C,PROBN,IER )
        GO TO 190
130     CALL BBPR( A,VNEW,C,PROBN,IER )
        GO TO 190
140     CALL BBAS( A,B,VNEW,PROBN,IER )
        GO TO 190
150     CALL BBAS( VNEW,B,C,PROBN,IER )
        GO TO 190
160     CALL BBAS( A,VNEW,C,PROBN,IER )
190   IF ( IER.NE.0 ) RETURN
C
C * * * * * IF THE DIFFERENCE IS LESS THAN THE GIVEN TORELANCE
C                 THEN STOP THE ITERATIONS
C
      IF ( ABS(PROBN-P).GT.TOL ) THEN
        IF ( P.GT.PROBN ) THEN
          VMIN = VNEW
        ELSE
```

```
          VMAX = VNEW
        ENDIF
        GO TO 100
      ELSE
        GOTO ( 210,220,230,210,220,230 ), IOPT
210       C = VNEW
          GO TO 290
220       A = VNEW
          GO TO 290
230       B = VNEW
290     CONTINUE
      ENDIF
390   RETURN
      END
C===========================================================
C
      SUBROUTINE WPOB( B,C,PROB )
C************************************************************
C + + + + + SPECIAL CASE OF WPAB(A,B,C,PROB)  WHEN A = 0
C + + + + + SPECIAL CASE OF GWPOB(B,C,D,PROB) WHEN D = 0
C************************************************************
C
C  1.  WPOB   - FORTRAN 77 SUBROUTINE
C
C  2.  THE ROUTINE CALCULATES AN APPROXIMATION RESULT OF THE
C      PROBABILITY OF THE WIENER PROCESS OF THE FOLLOWING FORM:
C
C      PROB = P { W ( T ) < C : 0 < T < B }
C
C  3.  PARAMETERS:
C        IN    B     :   SEE THE ABOVE FORMULA
C              C     :             "
C        OUT   PROB  :             "
C
C  4.  SUBROUTINES OR FUNCTIONS REQUIRED:
C        - PHI
C
C  5.  REFERENCES
C        - CHUNG( 1986, (3.2))
C
C************************************************************
      BB = SQRT( B )
      PROB = 2.0 * PHI( C/BB ) - 1.0
      RETURN
      END
C===========================================================
C
      REAL FUNCTION WFUNC1 ( U )
C************************************************************
C
C  1.  WFUNC1     - FORTRAN 77 REAL FUNCTION
C
C  2.  THIS FUNCTION CALCULATES THE FOLLOWING FORMULA
C
C      WFUNC1=EXP(-U*U/2)*[2*PHI(CC/BB)-1]/SQRT(2*PH)
C             WHERE CC = C-SQRT(A)*U AND BB = SQRT(B-A)
C
C  3.  PARAMETERS:
C        -INFINITE < U < C/SQRT(A)
C
C  4.  SUBROUTINES AND FUNCTIONS REQUIRED:
C        - PHI
C
C************************************************************
      COMMON /FUNCT/A,B,C,DUM(7)
      DATA PH/3.14159265359/
C
      BB = SQRT( B - A )
      CC = C - SQRT( A ) * U
      PP = 2.0 * PHI( CC/BB ) - 1.0
      CALL FLOW ( UFLOW, OFLOW )
      EUFLOW = -ALOG( UFLOW )
      UUM = U*U/2.0
      IF ( UUM.GT.EUFLOW ) THEN
         UUM = EUFLOW
      ENDIF
      WFUNC1 = PP * EXP( -UUM ) / SQRT(2.0*PH )
      RETURN
      END
C===========================================================
C
      SUBROUTINE WPAB ( A,B,C,PROB )
C************************************************************
C + + + + + SPECIAL CASE OF GWPAB(A,B,C,D,PROB) WHEN D = 0
C*************************************************************
C
C  1.  WPAB   - FORTRAN 77 SUBROUTINE
C
C  2.  THE ROUTINE CALCULATES AN APPROXIMATION RESULT OF THE
C      PROBABILITY OF THE WIENER PROCESS OF THE FOLLOWING FORM:
C
C      PROB = P { W ( T ) < C : A < T < B }
C
C  3.  PARAMETERS:
C        IN    A     :   SEE THE ABOVE FORMULA
C              B     :             "
C              C     :             "
C        OUT   PROB  :             "
C
C  4.  SUBROUTINES OR FUNCTIONS REQUIRED:
C        - WFUNC1 - PHI
C        - QNG - QMACO
C
C  5.  REFERENCES
C        - CHUNG ( 1986, (3.9))
C        - CSORGO ( 1976, (3.4) OF P. 554)
C        - QUADPACK ( 1983, SUBROUTINE QNG( P.130-136))
C        - RENYI ( 1953, (3.6) OF P. 208)
C
C************************************************************
      COMMON /FUNCT/A1,B1,C1,DUM(7)
      EXTERNAL WFUNC1
      DATA EPSABS/0.000005/, EPSREL/0.000005/, XMIN/-8.0/
C
      A1 = A
      B1 = B
      C1 = C
      XMAX = C/SQRT(A)
C
C * * * * * CALL QNG FOR NUMERICAL INTEGRATIONS WITH FUNCTION WFUNC1
C
      CALL QNG(WFUNC1,XMIN,XMAX,EPSABS,EPSREL,RESULT,ABSE,NEVAL,IER)
      IF ( IER.NE.0 ) THEN
        PROB = 121.
      ELSE
        PROB = RESULT
      ENDIF
      RETURN
      END
C===========================================================
C
      SUBROUTINE WPASOB ( B,C,PROB )
C************************************************************
C + + + + + SPECIAL CASE OF WPASAB(A,B,C,PROB)  WHEN A = 0
C + + + + + SPECIAL CASE OF WIEN1(C2,C1,B,PROB) WHEN C1 = C2 = C
C************************************************************
C
C  1.  WPASOB   - FORTRAN 77 SUBROUTINE
C
C  2.  THE ROUTINE CALCULATES AN APPROXIMATION RESULT OF THE
C      PROBABILITY OF THE WIENER PROCESS OF THE FOLLOWING FORM:
C
C      PROB = P { -C < W ( T ) < C : 0 < T < B }
C
C  3.  PARAMETERS:
C        IN    B     :   SEE THE ABOVE FORMULA
C              C     :             "
C        OUT   PROB  :             "
C
C  4.  SUBROUTINES OR FUNCTIONS REQUIRED:
C        - FLOW
C
C  5.  REFERENCES
C        - CHUNG( 1986, (4.4))
C        - FELLER( 1966, (5.11) OF P. 330)
C
C************************************************************
      DATA IMAX/100/, PH/3.14159265359/
C
      CALL FLOW( UFLOW,OFLOW )
      EUFLOW = -ALOG( UFLOW )
      CC = PH * PH * B / ( 8.0 * C * C )
      SIGN = 1.0
      SUM = EXP( -CC )
      DO 10 K = 1, IMAX
        KK = 2 * K + 1
        XK = FLOAT( KK )
        ARG = XK * XK * CC
        IF ( ARG .GT. EUFLOW ) THEN
          GO TO 20
        ELSE
          SIGN = -1.0 * SIGN
          SUM = SUM + SIGN * EXP( -ARG ) / XK
        ENDIF
10    CONTINUE
20    PROB = SUM * 4.0 / PH
      RETURN
      END
C===========================================================
C
      REAL FUNCTION WFUNC2 ( U )
C************************************************************
C
C  1.  WFUNC2     - FORTRAN 77 REAL FUNCTION
C
C  2.  THIS FUNCTION CALCULATES THE FOLLOWING FORMULA
C
C      WFUNC2=EXP(-U*U/2)*PROB{-C-U*AA<W(T)<C-U*AA:0<T<B-A}/SQRT(2*PH)
C        WHERE  AA = SQRT(A)
C
C  3.  PARAMETERS:
C        0 < U < C/SQRT(A)
C
C  4.  SUBROUTINES AND FUNCTIONS REQUIRED:
C        - WFUNC2 - WIEN1 - PHI
C        - QNG - QMACO
C
C************************************************************
      COMMON /FUNCT/ A,B,C,DUM(7)
      DATA PH/3.14159265359/
C
      UU = SQRT(A) * U
      BB = B - A
C
C * * * * * CALL WIEN1 TO COMPUTE PROB = PROB{-C-UU<W(T)<C-UU : 0<T<B-A}
C
      CALL WIEN1 ( -C-UU, C-UU, BB, PROB )
      CALL FLOW ( UFLOW, OFLOW )
```

```
      EUFLOW = -ALOG( UFLOW )
      UUM = U*U/2.0
      IF ( UUM.GT.EUFLOW ) THEN
        UUM = EUFLOW
      ENDIF
      WFUNC2 = EXP ( -UUM ) * PROB / SQRT ( 2.*PH )
      RETURN
      END
C=====================================================================
C
      SUBROUTINE WPASAB ( A,B,C,PROB )
C*********************************************************************
C + + + + + SPECIAL CASE OF GWPSAB(A,B,C,D,PROB) WHEN D = 0
C*********************************************************************
C
C  1.  WPASAB   - FORTRAN 77 SUBROUTINE
C
C  2.  THE ROUTINE CALCULATES AN APPROXIMATION RESULT OF THE
C      PROBABILITY OF THE WIENER PROCESS OF THE FOLLOWING FORM:
C
C      PROB = P { -C < W ( T ) < C : A < T < B }
C
C  3.  PARAMETERS:
C        IN    A     :   SEE THE ABOVE FORMULA
C              B     :          "
C              C     :          "
C        OUT   PROB  :          "
C
C  4.  SUBROUTINES OR FUNCTIONS REQUIRED:
C        - WFUNC2 - WIEN1 - PHI
C        - QNG - QMACO
C
C  5.  REFERENCES
C        - CHUNG( 1986, (4.11))
C        - QUADPACK( 1983, SUBROUTINE QNG( P.130-136))
C
C*********************************************************************
      COMMON /FUNCT/ A1,B1,C1,DUM(7)
      EXTERNAL WFUNC2
      DATA EPSABS/0.000005/, EPSREL/0.000005/, XMIN/0.0/
C
      A1 = A
      B1 = B
      C1 = C
      P = C / SQRT( A )
C
C * * * * * CALL QNG FOR NUMERICAL INTEGRATIONS WITH FUNCTION WFUNC1
C
      CALL QNG(WFUNC2,XMIN,P,EPSABS,EPSREL,RESULT,ABSE,NEVAL,IER)
      IF ( IER.NE.0 ) THEN
        PROB = 122.0
      ELSE
        PROB = RESULT * 2.0
      ENDIF
      RETURN
      END
C=====================================================================
C
      SUBROUTINE WIEN1 ( C2,C1,B,PROB )
C*********************************************************************
C + + + + + SPECIAL CASE OF WIENER(C2,D2,C1,D1,B,PROB) WHEN D2=D1=0
C*********************************************************************
C
C  1.  WIEN1    - FORTRAN 77 SUBROUTINE
C
C  2.  THE ROUTINE CALCULATES AN APPROXIMATION RESULT OF THE
C      PROBABILITY OF THE WIENER PROCESS OF THE FOLLOWING FORM:
C
C      PROB = P { C2 < W ( T ) < C1 : 0 < T < B }
C
C      IT IS A SPECIAL CASE OF THE ROUTINE WIENER WHEN D2 = D1 = 0
C
C  3.  PARAMETERS:
C        IN    B     :   SEE THE ABOVE FORMULA
C              C2    :          "                AND C2 < 0
C              C1    :          "                    C1 > 0
C        OUT   PROB  :          "
C
C  4.  SUBROUTINES OR FUNCTIONS REQUIRED:
C        - PHI
C
C  5.  REFERENCES
C        - CHUNG( 1986, (4.2) AND (4.3))
C        - CSORGO( 1967, (2.5) OF P. 553)
C        - FELLER( 1966, (5.8) OF P. 329 AND (5.9) OF P. 330)
C
C*********************************************************************
      DATA IMAX/100/, TOL/1.0E-9/
C
      BB = SQRT( B )
      CB1 = C1 / BB
      CB2 = C2 / BB
      CC = CB1 - CB2
      SIGN = 1.0
      SUM = 1.0
      DO 10 K = 1, IMAX
        CBK = -FLOAT( K ) * CC
        ADD = PHI( CBK + CB1 ) + PHI( CBK - CB2 )
        IF ( ABS(ADD).LT.TOL ) GO TO 20
        SIGN = -1.0 * SIGN
        SUM = SUM + 2.0 * ADD * SIGN
10    CONTINUE
20    PROB = SUM
      RETURN
      END
C=====================================================================
C
      SUBROUTINE BB01( C,PROB )
C*********************************************************************
C + + + + + SPECIAL CASE OF BB0B(B,C,PROB) WHEN B = 1
C*********************************************************************
C
C  1.  BB01   - FORTRAN 77 SUBROUTINE
C
C  2.  THE ROUTINE CALCULATES AN APPROXIMATION RESULT OF THE
C      PROBABILITY OF THE BROWNIAN BRIDGE PROCESS OF THE FOLLOWING FORM:
C
C      PROB = P { B ( T ) < C : 0 < T < 1 }
C
C  3.  PARAMETERS:
C        IN    C     :   SEE THE ABOVE FORMULA
C        OUT   PROB  :          "
C
C  4.  SUBROUTINES OR FUNCTIONS REQUIRED:
C        NONE
C
C*********************************************************************
      PROB = 1.0 - EXP ( -2.0*C*C )
      RETURN
      END
C=====================================================================
C
      SUBROUTINE BB0B ( B,C,PROB )
C*********************************************************************
C + + + + + SPECIAL CASE OF BBAB(A,B,C,PROB)   WHEN A = 0
C + + + + + SPECIAL CASE OF GWP0B(BB,C,D,PROB) WHEN D=C, BB=B/(1-B)
C*********************************************************************
C
C  1.  BBAS   - FORTRAN 77 SUBROUTINE
C
C  2.  THE ROUTINE CALCULATES AN APPROXIMATION RESULT OF THE
C      PROBABILITY OF THE BROWNIAN BRIDGE PROCESS OF THE FOLLOWING FORM:
C
C      PROB = P { B ( T ) < C : 0 < T < B }
C
C  3.  PARAMETERS:
C        IN    B     :   SEE THE ABOVE FORMULA
C              C     :          "
C        OUT   PROB  :          "
C
C  4.  SUBROUTINES OR FUNCTIONS REQUIRED:
C        - GWP0B - PHI
C
C  5.  REFERENCES:
C        - CHUNG ( 1986, (3.3))
C        - HALL AND WELLNER( 1980, (2.8) OF P. 136)
C        - KOZIOL AND BYAR( 1975, (2.3) OF P.508)
C
C*********************************************************************
      BB = B / ( 1. - B )
      CALL GWP0B ( BB,C,C,PROB )
      RETURN
      END
C=====================================================================
C
      SUBROUTINE BBAB ( A,B,C,PROB )
C*********************************************************************
C + + + + + SPECIAL CASE OF GWPAB(AA,BB,C,D,PROB) WHEN
C               AA = A/(1-A) , BB = B/(1-B) AND C = D
C*********************************************************************
C
C  1.  BBAB   - FORTRAN 77 SUBROUTINE
C
C  2.  THE ROUTINE CALCULATES AN APPROXIMATION RESULT OF THE
C      PROBABILITY OF THE BROWNIAN BRIDGE PROCESS OF THE FOLLOWING FORM:
C
C      PROB = P { B ( T ) < C : A < T < B }
C
C  3.  PARAMETERS:
C        IN    A     :   SEE THE ABOVE FORMULA
C              B     :          "
C              C     :          "
C        OUT   PROB  :          "
C
C  4.  SUBROUTINES OR FUNCTIONS REQUIRED:
C        - GWPAB - QNG - QMACO
C               - WFUNC3 - GWP0B - PHI
C
C  5.  REFERENCES:
C        - CHUNG ( 1986, (3.10))
C        - CSAKI ( 1981, (2.17) OF P. 259)
C        - CSORGO ( 1983, (4.2.10) OF P.37)
C
C*********************************************************************
      AA = A / ( 1. - A )
      BB = B / ( 1. - B )
      CALL GWPAB( AA,BB,C,C,PROB )
      RETURN
      END
C=====================================================================
C
      SUBROUTINE BBAS01( C,PROB )
C*********************************************************************
C + + + + + SPECIAL CASE OF BBASOB(B,C,PROB) WHEN B = 1
C*********************************************************************
C
C  1.  BBAS01   - FORTRAN 77 SUBROUTINE
C
```

```
C  2.  THE ROUTINE CALCULATES AN APPROXIMATION RESULT OF THE
C      PROBABILITY OF THE BROWNIAN BRIDGE PROCESS OF THE FOLLOWING FORM:
C
C      PROB = P { -C < B ( T ) < C : 0 < T < 1 }
C
C  3.  PARAMETERS:
C        IN    C     :  SEE THE ABOVE FORMULA
C        OUT   PROB  :           "
C
C  4.  SUBROUTINES OR FUNCTIONS REQUIRED:
C        - FLOW
C
C*****************************************************************
      DATA IMAX/100/
      CALL FLOW( UFLOW,OFLOW )
      EUFLOW = -ALOG( UFLOW )
      CC = 2. * C * C
      SIGN = 1.0
      SUM = 0.
      DO 10 K = 1, IMAX
        UP = CC * K * K
        IF ( UP.GE.EUFLOW ) THEN
          GO TO 20
        ELSE
          SIGN = SIGN * (-1)
          SUM = SUM + SIGN * EXP ( -UP )
        ENDIF
10    CONTINUE
20    PROB = 1.0 + 2.0 * SUM
      RETURN
      END
C==========================================================
C
      SUBROUTINE BBASOB ( B,C,PROB )
C***********************************************************
C + + + + + SPECIAL CASE OF BBASAB(A,B,C,PROB)  WHEN A = 0
C + + + + + SPECIAL CASE OF GWPSOB(BB,C,D,PROB) WHEN C=D, BB=B/(1-B)
C***********************************************************
C
C  1.  BBASOB   - FORTRAN 77 SUBROUTINE
C
C  2.  THE ROUTINE CALCULATES AN APPROXIMATION RESULT OF THE
C      PROBABILITY OF THE BROWNIAN BRIDGE PROCESS OF THE FOLLOWING FORM:
C
C      PROB = P { -C < B ( T ) < C : 0 < T < B }
C
C  3.  PARAMETERS:
C        IN    B     :  SEE THE ABOVE FORMULA
C              C     :           "
C        OUT   PROB  :           "
C
C  4.  SUBROUTINES OR FUNCTIONS REQUIRED:
C        - GWPSOB - FLOW
C                 - PHI
C
C  5.  REFERNCES:
C        - CHUNG ( 1986, (4.5))
C        - HALL AND WELLNER ( 1980, (2.9) OF P. 136)
C
C*****************************************************************
      BB = B / ( 1. - B )
      CALL GWPSOB(BB,C,C,PROB)
      RETURN
      END
C==========================================================
C
      SUBROUTINE BBASAB( A,B,C,PROB )
C***********************************************************
C + + + + + SPECIAL CASE OF GWPSAB(AA,BB,C,D,PROB) WHEN
C            AA=A/(1-A), BB=B/(1-B) AND C=D
C***********************************************************
C
C  1.  BBASAB   - FORTRAN 77 SUBROUTINE
C
C  2.  THE ROUTINE CALCULATES AN APPROXIMATION RESULT OF THE
C      PROBABILITY OF THE BROWNIAN BRIDGE PROCESS OF THE FOLLOWING FORM:
C
C      PROB = P { -C < B ( T ) < C : A < T < B }
C
C  3.  PARAMETERS:
C        IN    A     :  SEE THE ABOVE FORMULA
C              B     :           "
C              C     :           "
C        OUT   PROB  :           "
C
C  4.  SUBROUTINES OR FUNCTIONS REQUIRED:
C        - GWPSAB - QNG - QMACO
C                 - WFUNC4 - WIENER - FLOW
C                                   - PHI
C
C  5.  REFERNCES:
C        - CHUNG ( 1986, (4.12))
C
C*****************************************************************
      AA = A / ( 1. - A )
      BB = B / ( 1. - B )
      CALL GWPSAB( AA,BB,C,C,PROB )
      RETURN
      END
C==========================================================
C
      SUBROUTINE GWPOB ( B,C,D,PROB )
C***********************************************************
C + + + + + SPECIAL CASE OF GWPAB(A,B,C,D,PROB) WHEN A = 0
C***********************************************************
C
C  1.  GWPOB    - FORTRAN 77 SUBROUTINE
C
C  2.  THE ROUTINE CALCULATES AN APPROXIMATION RESULT OF THE
C      PROBABILITY OF THE WIENER PROCESS OF THE FOLLOWING FORM:
C
C      PROB = P { W ( T ) < C + D*T : 0 < T < B }
C
C  3.  PARAMETERS:
C        IN    B     :  SEE THE ABOVE FORMULA
C              C     :           "
C              D     :           "
C        OUT   PROB  :           "
C
C  4.  SUBROUTINES OR FUNCTIONS REQUIRED:
C        - PHI
C
C  5.  REFERENCES:
C        - CHUNG ( 1986, (3.1))
C
C*****************************************************************
      BB = SQRT(B)
      C1 = ( C + D*B )/BB
      C2 = (-C + D*B )/BB
      PROB = PHI( C1 ) - EXP( -2.0*C*D ) * PHI( C2 )
      RETURN
      END
C==========================================================
C
      REAL FUNCTION WFUNC3 ( U )
C***********************************************************
C
C  1.  WFUNC3   - FORTRAN 77 REAL FUNCTION
C
C  2.  IT CALCULATES THE FOLLOWING FORMULA
C
C      WFUNC3=EXP(-U*U/2)*PROB{W(T)<Q-U*AA+D*T:0<T<B-A}/SQRT(2*PH)
C             WHERE Q=C+D*A AND AA=SQRT(A)
C
C  3.  PARAMETERS:
C        -INFINITE < U < C + D*A
C
C  4.  SUBROUTINES OR FUNCTIONS REQUIRED:
C        - GWPOB - PHI
C
C  5.  REFERENCES:
C        - CHUNG (1986, (3.8))
C
C*****************************************************************
      COMMON /FUNCT/A,B,C,D,DUM(6)
      DATA PH/3.14159265359/
      BB = B - A
      CC = C + D*A - SQRT(A)*U
      CALL GWPOB( BB,CC,D,PROB )
      CALL FLOW ( UFLOW, OFLOW )
      EUFLOW = -ALOG( UFLOW )
      UUM = U*U/2.0
      IF ( UUM.GT.EUFLOW ) THEN
        UUM = EUFLOW
      ENDIF
      WFUNC3 = EXP( -UUM ) * PROB / SQRT(2.*PH)
      RETURN
      END
C==========================================================
C
      SUBROUTINE GWPAB ( A,B,C,D,PROB )
C***********************************************************
C
C  1.  GWPAB    - FORTRAN 77 SUBROUTINE
C
C  2.  THE ROUTINE CALCULATES AN APPROXIMATION RESULT OF THE
C      PROBABILITY OF THE WIENER PROCESS OF THE FOLLOWING FORM:
C
C      PROB = P { W ( T ) < C + D*T : A < T < B }
C
C  3.  PARAMETERS:
C        IN    A     :  SEE THE ABOVE FORMULA
C              B     :           "
C              C     :           "
C              D     :           "
C        OUT   PROB  :           "
C
C  4.  SUBROUTINES OR FUNCTIONS REQUIRED:
C        - QNG - QMACO
C        - WFUNC3 - GWPOB - PHI
C
C  5.  REFERNCES:
C        - CHUNG ( 1986, (3.5))
C
C*****************************************************************
      COMMON /FUNCT/A1,B1,C1,D1,DUM(6)
      EXTERNAL WFUNC3
      DATA EPSABS/0.000005/, EPSREL/0.000005/, XMIN/-8.0/
C
      A1 = A
      B1 = B
      C1 = C
      D1 = D
      XMAX = ( C + D*A ) / SQRT(A)
C
C * * * * * CALL QNG FOR NUMERICAL INTEGRATION WITH WFUNC3
C
      CALL QNG( WFUNC3,XMIN,XMAX,EPSABS,EPSREL,RESULT,ABSE,NEVAL,IER )
```

```
      IF ( IER.NE.0 ) THEN
        PROB = 123.
      ELSE
        PROB = RESULT
      ENDIF
      RETURN
      END
C===============================================================
C
      SUBROUTINE GWPS0B ( B,C,D,PROB )
C**************************************************************
C + + + + + SPECIAL CASE OF GWPSAB(A,B,C,D,PROB) WHEN A = 0
C**************************************************************
C
C  1.  GWPS0B  - FORTRAN 77 SUBROUTINE
C
C  2.  THE ROUTINE CALCULATES AN APPROXIMATION RESULT OF THE
C      PROBABILITY OF THE WIENER PROCESS OF THE FOLLOWING FORM:
C
C      PROB = P { -C - D*T < W ( T ) < C + D*T : 0 < T < B }
C
C  3.  PARAMETERS:
C        IN   B    :  SEE THE ABOVE FORMULA
C             C    :         "
C             D    :         "
C       OUT   PROB :         "
C
C  4.  SUBROUTINES OR FUNCTIONS REQUIRED:
C        - FLOW
C        - PHI
C
C  5.  REFERENCES:
C        - CHUNG ( 1986, (4.1))
C        - GILLASPIE AND FISHER ( 1979, P. 921)
C
C*****************************************************************
      DATA IMAX/100/
C
      CALL FLOW( UFLOW,OFLOW )
      EUFLOW = -ALOG( UFLOW )
      BB = SQRT( B )
      BC = ( C + D*B ) / BB
      CD = 2.0 * C * D
      SIGN = 1.0
      SUM = 0.
      DO 10 K = 1, IMAX
        XK = FLOAT( K )
        UP = CD * XK * XK
        IF ( UP.GE.EUFLOW ) THEN
          GO TO 20
        ELSE
          AD = -2.0 * XK * C / BB
          SIGN = SIGN * (-1.0)
          TERM = PHI( AD + BC ) - PHI( AD - BC )
          SUM = SUM + SIGN * EXP(-UP) * TERM
        ENDIF
10    CONTINUE
20    PROB = 1.0 - 2.0 * PHI(-BC) + 2.0 * SUM
      RETURN
      END
C===============================================================
C
      REAL FUNCTION WFUNC4( U )
C**************************************************************
C
C  1.  WFUNC4   - FORTRAN 77 REAL FUNCTION
C
C  2.  IT CALCULATES THE FOLLOWING FORMULA
C
C      WFUNC4=EXP(-U*U/2)*
C                PROB{-Q-U*AA-D*T<W(T)<Q-U*AA+D*T:0<T<B-A}/SQRT(2*PH)
C        WHERE Q=C+D*A AND AA=SQRT(A)
C
C  3.  PARAMETERS:
C        -C -D*A < U < C + D*A
C
C  4.  SUBROUTINES OR FUNCTIONS REQUIRED:
C        - WIENER - FLOW
C                 - PHI
C
C  5.  REFERENCES:
C
C        - CHUNG (1986, (4.7))
C
C*****************************************************************
      COMMON /FUNCT/A,B,C,D,DUM(6)
      DATA PH/3.14159265359/
C
      CD = C + D*A
      UU = SQRT( A ) * U
      CALL WIENER ( -CD-UU, -D, CD-UU, D, B-A, PROB )
      CALL FLOW ( UFLOW, OFLOW )
      EUFLOW = -ALOG( UFLOW )
      UUM = U*U/2.0
      IF ( UUM.GT.EUFLOW ) THEN
         UUM = EUFLOW
      ENDIF
      WFUNC4 = EXP( -UUM ) * PROB / SQRT(2.0*PH)
      RETURN
      END
C===============================================================
C
      SUBROUTINE GWPSAB ( A,B,C,D,PROB )
C**************************************************************
C
C  1.  GWPSAB  - FORTRAN 77 SUBROUTINE
C
C  2.  THE ROUTINE CALCULATES AN APPROXIMATION RESULT OF THE
C      PROBABILITY OF THE WIENER PROCESS OF THE FOLLOWING FORM:
C
C      PROB = P { -C - D*T < W ( T ) < C + D*T : A < T < B }
C
C  3.  PARAMETERS:
C        IN   A    :  SEE THE ABOVE FORMULA
C             B    :         "
C             C    :         "
C             D    :         "
C       OUT   PROB :         "
C
C  4.  SUBROUTINES OR FUNCTIONS REQUIRED:
C        - QNG - QMACO
C        - WFUNC4 - WIENER - FLOW
C                          - PHI
C
C  5.  REFERENCES:
C        - CHUNG ( 1986, (4.7))
C
C*****************************************************************
      COMMON /FUNCT/A1,B1,C1,D1,DUM(6)
      EXTERNAL WFUNC4
      DATA EPSABS/0.000005/, EPSREL/0.000005/, XMIN/0.0/
C
      A1 = A
      B1 = B
      C1 = C
      D1 = D
      CD = ( C + D*A ) / SQRT(A)
C
C * * * * * CALL QNG FOR NUMERICAL INTEGRATION WITH WFUNC4
C
      CALL QNG( WFUNC4,XMIN,CD,EPSABS,EPSREL,RESULT,ABSE,NEVAL,IER )
      IF ( IER.NE.0 ) THEN
        PROB = 124.
      ELSE
        PROB = RESULT * 2.0
      ENDIF
      RETURN
      END
C===============================================================
C
      SUBROUTINE WIENER ( C2,D2,C1,D1,S,PROB )
C**************************************************************
C
C  1.  WIENER   - FORTRAN 77 SUBROUTINE
C
C  2.  THE ROUTINE CALCULATES AN APPROXIMATION RESULT OF THE
C      PROBABILITY OF THE WIENER PROCESS OF THE FOLLOWING FORM:
C
C      PROB = P { C2 + D2*T < W ( T ) < C1 + D1*T : 0 < T < S }
C
C  3.  PARAMETERS:
C        IN   C2   :  SEE THE ABOVE FORMULA
C             D2   :         "
C             C1   :         "
C             D1   :         "
C             S    :         "
C       OUT   PROB :         "
C
C  4.  SUBROUTINES OR FUNCTIONS REQUIRED:
C        - FLOW
C        - PHI
C
C  5.  REFERENCES:
C
C        - ANDERSON (1961, (4.32) OF P.180)
C        - CHUNG (1986, (2.9))
C
C*****************************************************************
      DATA IMAX/100/
C
C * * * * * SETTING UP ALL CONSTANT TERMS FOR COMPUTATIONS
C
      SS   = SQRT ( S )
      V    = ( C1 + D1*S ) / SS
      W    = ( C2 + D2*S ) / SS
      CC   = C1 - C2
      DD   = D1 - D2
      CD1  = CC * D1 + DD * C1
      CD2  = CC * D2 + DD * C2
      C1D1 = C1 * D1
      C2D2 = C2 * D2
      CDCD = C1 * D2 - D1 * C2
      CD   = CC * DD
      CCSS = -2.0 * CC / SS
      C1S  = 2.0 * C1 / SS
      C2S  = 2.0 * C2 / SS
C
C * * * * * CALL FLOW TO OBTAIN MINIMUM AND MAXIMUM ALLOWED NO.
C
      CALL FLOW ( UFLOW, OFLOW )
      EUFLOW = -ALOG( UFLOW )
      SUM = 0.0
      DO 50 K = 1, IMAX
        X = FLOAT( K )
        ISTOP = 0
        CSK = CCSS * X
        CDK = CD * X * X
        P1 = CSK + W
```

```
SUBROUTINE  QNG(F,A,B,EPSABS,EPSREL,RESULT,ABSERR,NEVAL,IER)
```

This subroutine has been published by Piessens et al. 1983, p. 130-136.

```
SUBROUTINE QMACO(EPMACH,UFLOW,OFLOW)
```

This subroutine has been published by Piessens et al. 1983, p. 294.

Energy, Mines and Resources Canada Énergie, Mines et Ressources Canada