Journal of Statistical Software
June 2008, Volume 26, Issue 3. http://www.jstatsoft.org/

Arbitrary Precision Mathematica Functions to Evaluate the One-Sided One Sample K-S Cumulative Sampling Distribution

J. Randall Brown
Kent State University

Milton E. Harvey
Kent State University

Abstract

Efficient rational arithmetic methods that can exactly evaluate the cumulative sampling distribution of the one-sided one sample Kolmogorov-Smirnov (K-S) test have been developed by Brown and Harvey (2007) for sample sizes n up to fifty thousand. This paper implements in arbitrary precision the same 13 formulae to evaluate the one-sided one sample K-S cumulative sampling distribution. Computational experience identifies the fastest implementation which is then used to calculate confidence interval bandwidths and p values for sample sizes up to ten million.

Keywords: K-S cumulative sampling distributions, K-S one-sided one sample probabilities, K-S confidence bands, arbitrary precision arithmetic.

1. Introduction

In a recent paper, Brown and Harvey (2007) evaluated 13 formulae for calculating the exact p values of the one-sided one sample Kolmogorov-Smirnov (K-S) test. They used rational arithmetic so that the p values were calculated exactly and the implementation of all 13 formulae yielded the same p values. From comparisons of the computational times which increased with increasing sample size, increasing number of test statistic digits, and decreasing value of the p value, the formulae were ranked from the fastest to the slowest. For $\rho = 3$ digits in the test statistic and a p value of 0.001, the computer time for the fastest formula was 1.406 seconds for a sample size of $n = 10,000$ and 81.047 seconds for a sample size of $n = 50,000$ on a Pentium IV running at 2.4 GHz. In contrast, for $\rho = 6$ digits in the test statistic and the same p value of 0.001, the computer time for the fastest formula was 23.657 seconds for a sample size of $n = 10,000$ and 1213.360 seconds for a sample size of $n = 50,000$.

Rational arithmetic stores every number as a ratio of two integers (a rational number) where each integer can have as many decimal digits as needed to express the number exactly. However, even the fastest rational arithmetic method only calculated p values for sample sizes up to fifty thousand. This paper develops an alternate computational environment that is faster than all the rational arithmetic implementations. Arbitrary precision methods are used to calculate one-sided one sample K-S p values where the accuracy of the resultant p value is specified by the user. Comparative analysis of the rational arithmetic implementations in Brown and Harvey (2007) with the arbitrary precision implementations in this paper show that the arbitrary precision methods are over ten times faster than the rational arithmetic methods. Although arbitrary precision methods are faster, a major difficulty is determining the accuracy of any p value produced by arbitrary precision methods. The p values calculated by rational arithmetic in Brown and Harvey (2007) will be used to check the accuracy of this paper's arbitrary precision calculations.

Arbitrary precision uses floating point quantities where the number of decimal digits (precision) is specified by the user. Every arbitrary precision quantity has a fixed number of digits but unlike rational arithmetic the precision of the calculated quantities can decrease as calculations are made. However, Mathematica in its arbitrary precision package automatically keeps track of every number's precision. To quote from the Mathematica Book (Wolfram 2003) (page 731), "When you do a computation, Mathematica keeps track of which digits in your result could be affected by unknown digits in your input. It sets the precision of your result so that no affected digits are ever included. This procedure ensures that all digits returned by Mathematica are correct, whatever the values of the unknown digits may be." In other words, the user specifies the internal precision ip to evaluate an expression and then Mathematica evaluates the expression so that the resulting precision rp (the precision of the quantity found by calculating the value of the expression) is equal to ip if at all possible. If this is not possible, then Mathematica produces a result that maximizes rp given the limitations imposed by the precision of the inputs and the computations.

Since the internal precision ip must be specified before computations begin, determining the relationship between the internal precision ip and the resulting precision rp makes arbitrary precision methods more difficult to implement than rational arithmetic methods. By finding functions that predict the internal precision ip needed to produce a p value with a resulting precision of at least rp, this paper develops arbitrary precision methods to calculate one-sided one sample K-S p values to any desired accuracy for sample sizes up to ten million, $n \leq 10,000,000$.

Mathematica 5 was used to develop all the code in this paper. However, the code was tested in Mathematica 6 and will work in both Mathematica 5 and 6.

2. K-S cumulative sampling distribution formulae

The one-sided one sample K-S test uses the maximum distance between the hypothesized continuous cumulative distribution $F(x)$ and the empirical cumulative distribution $F_n(x)$. There are two one-sided one sample random variables: the one-sided upper random variable $D_n^+ = \sup_{-\infty<x<\infty}\{F_n(x) - F(x)\}$ and the one-sided lower random variable $D_n^- = \sup_{-\infty<x<\infty}\{F(x) - F_n(x)\}$. Since by symmetry D_n^+ and D_n^- have the same cumulative sampling distribution, D_n^+ is used to represent both cases. The cumulative sampling distri-

Type	Formula to compute $P\left[D_n^+ \geq d^+\right]$ for $0 < d^+ \leq 1$
SmirnovD	$d^+ \sum_{j=0}^{\lfloor n(1-d^+)\rfloor} \binom{n}{j} \left(1 - \frac{j}{n} - d^+\right)^{n-j} \left(\frac{j}{n} + d^+\right)^{j-1}$
DwassD	$1 - d^+ \sum_{j=0}^{\lfloor nd^+\rfloor} \binom{n}{j} \left(1 - \frac{j}{n} + d^+\right)^{n-j-1} \left(\frac{j}{n} - d^+\right)^{j}$
SmirnovAltD	$\frac{d^+}{n^{n-1}} \sum_{j=0}^{\lfloor n(1-d^+)\rfloor} \binom{n}{j} \left(n - d^+n - j\right)^{n-j} \left(d^+n + j\right)^{j-1}$
DwassAltD	$1 - \frac{d^+}{n^{n-1}} \sum_{j=0}^{\lfloor nd^+\rfloor} \binom{n}{j} \left(n - j + d^+n\right)^{n-j-1} \left(j - d^+n\right)^{j}$

$\lfloor n(1-d^+)\rfloor$ is the greatest integer less than or equal to $n(1-d^+)$

Table 1: K-S one-sided one sample direct formulae.

bution is used to calculate the p value $P(D^+ \geq d^+)$ for test statistic d^+. Brown and Harvey (2007) found or developed 13 formulae (four direct formulae, four iterative formulae, and five recursion formulae) that can be used to evaluate the one-sided one sample K-S cumulative sampling distribution. This section summarizes the 13 formulae.

2.1. Direct formulae

A closed form expression of the one-sided one sample K-S cumulative sampling distribution was developed by Smirnov (1944) and verified by many scholars including Feller (1948) and Birnbaum and Tingey (1951). For $0 < d^+ \leq 1$ and sample size n, Smirnov's distribution, SmirnovD, is shown in the first row of Table 1 where $\lfloor n(1-d^+)\rfloor$ is the greatest integer less than or equal to $n(1-d^+)$. Dwass (1959) derived a different formula, DwassD, that is also shown in Table 1. Second forms of the Smirnov distribution, SmirnovAltD, and the Dwass distribution, DwassAltD, are derived by factoring $1/n^{n-1}$ out of their respective formulae.

2.2. Iterative formulae

Brown and Harvey (2007) transformed each of the four formulae in Table 1 into an iterative version which are presented in Table 2.

2.3. Daniels' recursion formula

Daniels (1945) derived a difference equation that can be solved for the following formula.

$$Q_0(1) = 1$$

Type	Iterative formula
Smirnov	$x_j = \frac{(n-j+1)\,(d^+n+j)}{j\,(n-d^+n-j)} \times \left[1-\frac{1}{n-d^+n-j+1}\right]^{n-j+1} \times \left[1+\frac{1}{d^+n+j-1}\right]^{j-2}$
Dwass	$y_j = \frac{(n-j+1)(j-d^+n)}{j(n-j+1+d^+n)} \times \left[1-\frac{1}{n-j+1+d^+n}\right]^{n-j-1} \times \left[1+\frac{1}{j-1-d^+n}\right]^{j-1}$

Name	Initial value	Iteration	$P\left[D_n^+ \geq d^+\right]$
SmirnovI	$\gamma_0 = (1-d^+)^n$	$\gamma_j = x_j\gamma_{j-1}$	$\sum_{j=0}^{\lfloor n(1-d^+)\rfloor} \gamma_j$
SmirnovAltI	$\gamma_0 = n^n(1-d^+)^n$	$\gamma_j = x_j\gamma_{j-1}$	$n^n \sum_{j=0}^{\lfloor n(1-d^+)\rfloor} \gamma_j$
DwassI	$\gamma_0 = d^+(1+d^+)^{n-1}$	$\gamma_j = y_j\gamma_{j-1}$	$1-\sum_{j=0}^{\lfloor nd^+\rfloor} \gamma_j$
DwassAltI	$\gamma_0 = d^+(n+d^+n)^{n-1}$	$\gamma_j = y_j\gamma_{j-1}$	$1-\left[\left(\sum_{j=0}^{\lfloor nd^+\rfloor} \gamma_j\right)/n^{n-1}\right]$

$\lfloor n(1-d^+)\rfloor$ is the greatest integer less than or equal to $n(1-d^+)$

Table 2: K-S one-sided one sample iterative formulae.

$$\begin{aligned} Q_i(1) &= -\sum_{k=0}^{i-1}\binom{i}{k}Q_k(1)\left[\max\left(\frac{i-t}{n},0\right)-1\right]^{i-k} \qquad \text{for } i=1,2,\cdots,n \\ P\left(D_n^+ \geq \frac{t}{n}\right) &= 1-Q_n(1) \end{aligned}$$

2.4. Noe and Vandewiele recursion formula

Since the Daniels recursion formula has both positive and negative terms, Noe and Vandewiele (1968) derived an alternate recursion formula that has only non-negative terms. Noe (1972) later added a correction to this recursion formula. The particular form of the recursion formula listed below containing Noe's correction is taken from Shorack and Wellner (1986, formulae 24 through 28 on page 363) and is denoted by Noe in the rest of the paper.

$$
\begin{aligned}
Q_0(0) &= 1 \\
Q_m(m) &= 0 \qquad \text{for } 1 \le m \le n+1 \\
Q_i(m) &= \sum_{k=0}^{i} \binom{i}{k} Q_k(m-1) \left[\max\left(\frac{m-t}{n}, 0\right) - \max\left(\frac{m-t-1}{n}, 0\right) \right]^{i-k} \\
&\qquad \text{for } 0 \le i \le m-1, 1 \le m \le n+1 \\
P\left(D_n^+ \ge \frac{t}{n}\right) &= 1 - Q_n(n+1)
\end{aligned}
$$

2.5. Steck recursion formula

Steck (1969) derived the recursion formula shown below that was later listed in Shorack and Wellner (1986).

$$
\begin{aligned}
b_j &= \min\left(\frac{j-1+t}{n}, 1\right) \qquad \text{for } j = 1, 2, \cdots, n \\
P_0 &= 1 \\
P_1 &= b_1 \\
P_i &= b_i^i - \sum_{m=0}^{i-2} \binom{i}{m} [b_i - b_{m+1}]^{i-m} P_m \qquad \text{for } i = 2, 3, \cdots, n \\
P\left(D_n^+ \ge \frac{t}{n}\right) &= 1 - P_n
\end{aligned}
$$

2.6. Conover recursion formula

Conover (1972) derived a recursion formula that Brown and Harvey (2007) simplified to the following form for a hypothesized continuous cumulative distribution $F(x)$.

$$
\begin{aligned}
e_0 &= 1 \\
e_k &= 1 - \sum_{j=0}^{k-1} \binom{k}{j} \left(1 - \frac{j}{n} - \frac{t}{n}\right)^{k-j} e_j \qquad \text{for } k = 1, 2, \cdots, \lfloor n - t \rfloor \\
P\left(D_n^+ \ge \frac{t}{n}\right) &= \sum_{j=0}^{\lfloor n-t \rfloor} \binom{n}{j} \left(1 - \frac{j}{n} - \frac{t}{n}\right)^{n-j} e_j
\end{aligned}
$$

2.7. Bolshev recursion formula

Kotelnikov and Chmaladze (1983) used the recursion formula shown below that was later called the Bolshev recursion in Shorack and Wellner (1986).

$$
\begin{aligned}
b_j &= \min\left(\frac{j-1+t}{n}, 1\right) && \text{for } j = 1, 2, \cdots, n \\
P_0 &= 1 \\
P_i &= 1 - \sum_{m=1}^{i} \binom{i}{m} [1 - b_{i-m+1}]^m \, P_{i-m} && \text{for } i = 1, 2, \cdots, n \\
P\left(D_n^+ \geq \frac{t}{n}\right) &= 1 - P_n
\end{aligned}
$$

3. Calculation error and computation time

Rounding and catastrophic cancellation are the two sources of calculation error. For rounding error, the size of the error grows with the number of calculations. In contrast, catastrophic cancellation can only occur when a negative number is added to a positive number and the size of the error is dependent on how close the absolute value of the negative number is to the positive number. Like possible rounding error, computation time also increases with the number of calculations.

In implementing the 13 formulae using arbitrary precision, the internal precision ip must be specified at the beginning of the calculations. At the end of the calculations, Mathematica gives both the value representing the result of the calculations and the value's resulting precision rp (number of decimal digits of precision). However, as the error increases, the resulting precision rp decreases for the same internal precision ip.

Before studying the relationship between ip and rp for the 13 formulae, the difficulty in evaluating the 13 formulae can be seen by looking at the number of terms as well as the smallest and largest terms in each formula. In each formula, the terms of the formula are added together to produce the p value $P\left(D_n^+ \geq d^+ = t/n\right)$. Table 3 defines what is a term for each of the 13 formulae.

For a specific test statistic d^+ and sample size n, the Mathematica functions implementing each of the 13 formulae in rational arithmetic from Brown and Harvey (2007) were modified to find the number of positive terms, the number of negative terms, the smallest negative term, the largest negative term, the smallest positive term, and the largest positive term. Table 4 provides a listing of these Mathematica functions which are contained in the approximately 630 kilobyte file `KS1SidedOneSampleLargestSmallestTermsRational.nb`.

In Section 14, file `KS1SidedOneSampleLargestSmallestTermsRational.nb` also contains the Mathematica function `LargeSmallTermsToFileKS1SidedOneSample` that uses all the Mathematica functions listed in Table 4 to produce an output file containing all the results. For the sample size $n = 200$ and the corresponding test statistics d^+ that produces a p value near 0.001 and 0.9, Table 5 lists the number of positive terms, the number of negative terms, the smallest negative term, the largest negative term, the smallest positive term, and the largest positive term for each formula. Table 5 shows that only the DwassD, DwassAltD, DwassI, DwassAltI, and Daniels formulae (the negative/positive term formulae) contain negative terms and thus are the only formulae that can have cancellation error. Since the SmirnovD, SmirnovAltD, SmirnovI, SmirnovAltI, Noe, Steck, Conover, and Bolshev formulae contain only non-negative

Formula	A term
SmirnovD	$\binom{n}{j}\left(1-\frac{j}{n}-d^+\right)^{n-j}\left(\frac{j}{n}+d^+\right)^{j-1}$
DwassD	$\binom{n}{j}\left(1-\frac{j}{n}+d^+\right)^{n-j-1}\left(\frac{j}{n}-d^+\right)^{j}$
SmirnovAltD	$\binom{n}{j}\left(n-d^+n-j\right)^{n-j}\left(d^+n+j\right)^{j-1}$
DwassAltD	$\binom{n}{j}\left(n-j+d^+n\right)^{n-j-1}\left(j-d^+n\right)^{j}$
SmirnovI	$\gamma_j = x_j\gamma_{j-1}$
DwassI	$\gamma_j = y_j\gamma_{j-1}$
SmirnovAltI	$\gamma_j = x_j\gamma_{j-1}$
DwassAltI	$\gamma_j = y_j\gamma_{j-1}$
Daniels	$\binom{i}{k}Q_k(m-1)\left[\max\left(\frac{m-t}{n},0\right)-\max\left(\frac{m-t-1}{n},0\right)\right]^{i-k}$
Noe	$\binom{i}{k}Q_k(m-1)\left[\max\left(\frac{m-t}{n},0\right)-\max\left(\frac{m-t-1}{n},0\right)\right]^{i-k}$
Steck	$\binom{i}{m}\left[b_i-b_{m+1}\right]^{i-m}P_m$
Conover	$\binom{k}{j}\left(1-\frac{j}{n}-\frac{t}{n}\right)^{k-j}e_j$
Bolshev	$\binom{i}{m}\left[1-b_{i-m+1}\right]^{m}P_{i-m}$

$$x_j = \frac{(n-j+1)\,(d^+n+j)}{j\,(n-d^+n-j)} \times \left[1-\frac{1}{n-d^+n-j+1}\right]^{n-j+1} \times \left[1+\frac{1}{d^+n+j-1}\right]^{j-2}$$

$$y_j = \frac{(n-j+1)(j-d^+n)}{j(n-j+1+d^+n)} \times \left[1-\frac{1}{n-j+1+d^+n}\right]^{n-j-1} \times \left[1+\frac{1}{j-1-d^+n}\right]^{j-1}$$

Table 3: Definition of a term for each formula.

terms, the only source of calculation error is rounding error. In contrast, the the negative/positive term formulae (DwassD, DwassAltD, DwassI, DwassAltI, and Daniels formulae) can have both rounding error and cancellation error. Thus, we would expect that the internal precision ip needed to produce a specific resulting precision rp would be much higher for the the negative/positive term formulae than for the non-negative term formulae (the SmirnovD, SmirnovAltD, SmirnovI, SmirnovAltI, Noe, Steck, Conover, and Bolshev formulae). All other

Formula name	Type formula	Mathematica function name	Listed in section
SmirnovD	Direct	`SmirnovDKS1SidedRTProbRationalLargeSmallTerms`	1
DwassD	Direct	`DwassDKS1SidedRTProbRationalLargeSmallTerms`	2
SmirnovAltD	Direct	`SmirnovAltDKS1SidedRTProbRationalLargeSmallTerms`	3
DwassAltD	Direct	`DwassAltDKS1SidedRTProbRationalLargeSmallTerms`	4
SmirnovI	Iterative	`SmirnovIKS1SidedRTProbRationalLargeSmallTerms`	5
DwassI	Iterative	`DwassIKS1SidedRTProbRationalLargeSmallTerms`	6
SmirnovAltI	Iterative	`SmirnovAltIKS1SidedRTProbRationalLargeSmallTerms`	7
DwassAltI	Iterative	`DwassAltIKS1SidedRTProbRationalLargeSmallTerms`	8
Daniels	Recursion	`DanielsKS1SidedRTProbRationalLargeSmallTerms`	9
Noe	Recursion	`NoeKS1SidedRTProbRationalLargeSmallTerms`	10
Steck	Recursion	`SteckKS1SidedRTProbRationalLargeSmallTerms`	11
Conover	Recursion	`ConoverKS1SidedRTProbRationalLargeSmallTerms`	12
Bolshev	Recursion	`BolshevKS1SidedRTProbRationalLargeSmallTerms`	13

Table 4: Mathematica function name to calculate the smallest and largest terms listed in file `KS1SidedOneSampleLargestSmallestTermsRational.nb`.

things being equal, a larger internal precision ip takes more computer time. So just considering the internal precision ip, the negative/positive term formulae would take more computer time than the non-negative term formulae.

Another factor that greatly affects computer time is the number of terms. Using the number of terms tabulated in Table 5, the formulae ranked from the smallest to the largest number of terms are DwassD, DwassAltD, DwassI, DwassAltI, SmirnovD, SmirnovAltD, SmirnovI, SmirnovAltI, Conover, Steck, Bolshev, Daniels, and Noe. In general, the Dwass-based formulae (DwassD, DwassAltD, DwassI, DwassAltI) have the fewest number of terms, the Smirnov base formulae (SmirnovD, SmirnovAltD, SmirnovI, SmirnovAltI) have the second fewest number of terms, and the recursion formulae (Conover, Steck, Bolshev, Daniels, Noe) have by far the largest number of terms. For rational arithmetic implementation of the 13 formulae, Brown and Harvey (2007) found the Dwass-based formulae were faster than the Smirnov-based formulae which were much faster than the recursion formulae. This difference in speed is undoubtedly due to the differences in the number of terms between the three groups.

The relative magnitude of the terms can also affect the error. For example, with an internal precision $ip = 20$, adding or subtracting 1.23×10^1 to 4.56×10^{23} has no effect as the result is 4.56×10^{23}. If in later calculations, catastrophic cancellation reduces the value to 1.11×10^2, then the contribution of 1.23×10^1 is lost and the correct value of $1.23 \times 10^1 + 1.11 \times 10^2 = 1.233 \times 10^2$ is not realized. Table 5 shows that the terms in each formula can have a large relative magnitude.

The computational tradeoff between the 13 formulae depends on three factors: the number

Formula	Number negative terms	Number positive terms	Smallest negative term	Largest negative term	Smallest positive term	Largest positive term
Sample size $n = 200$, $P[D_n^+ \geq d^+ = 0.1305] = 0.00098991393$ to 8 digits of precision						
SmirnovD	0	174			3.912×10^{-31}	1.186×10^{-4}
DwassD	13	14	-3.792×10^{-25}	-9.349×10^{12}	4.945×10^{-54}	9.033×10^{12}
SmirnovAltD	0	174			3.143×10^{427}	9.531×10^{453}
DwassAltD	13	14	-3.047×10^{433}	-7.511×10^{470}	3.973×10^{404}	7.258×10^{470}
SmirnovI	0	174			5.105×10^{-32}	1.548×10^{-5}
DwassI	13	14	-4.948×10^{-26}	-1.22×10^{12}	6.453×10^{-55}	1.179×10^{12}
SmirnovAltI	0	174			8.203×10^{428}	2.488×10^{455}
DwassAltI	13	14	-3.976×10^{432}	-9.802×10^{469}	5.185×10^{403}	9.472×10^{469}
Daniels	10,000	10,100	-1.325×10^{-177}	-1.227×10^{21}	2.03×10^{-174}	1.217×10^{21}
Noe	0	1,369,926			1.245×10^{-458}	$1.\times10^{0}$
Steck	0	19,900			4.071×10^{-33}	6.735×10^{-1}
Conover	0	15,224			7.144×10^{-13}	$10.\times10^{-1}$
Bolshev	0	19,749			5.105×10^{-32}	$10.\times10^{-1}$
Sample size $n = 200$, $P[D_n^+ \geq d^+ = 0.01545] = 0.89972290$ to 8 digits of precision						
SmirnovD	0	197			1.139×10^{-2}	3.276×10^{0}
DwassD	2	2	-1.307×10^{-4}	-1.637×10^{1}	1.724×10^{0}	2.114×10^{1}
SmirnovAltD	0	197			9.154×10^{455}	2.632×10^{458}
DwassAltD	2	2	-1.05×10^{454}	-1.315×10^{459}	1.386×10^{458}	1.698×10^{459}
SmirnovI	0	197			1.76×10^{-4}	5.061×10^{-2}
DwassI	2	2	-2.02×10^{-6}	-2.529×10^{-1}	2.664×10^{-2}	3.266×10^{-1}
SmirnovAltI	0	197			2.829×10^{456}	8.133×10^{458}
DwassAltI	2	2	-1.623×10^{452}	-2.032×10^{457}	2.141×10^{456}	2.624×10^{457}
Daniels	10,000	10,100	-6.105×10^{-363}	-1.452×10^{17}	7.903×10^{-359}	1.468×10^{17}
Noe	0	1,373,491			1.245×10^{-458}	$1.\times10^{0}$
Steck	0	19,900			4.071×10^{-33}	9.564×10^{-1}
Conover	0	19,502			4.442×10^{-2}	$10.\times10^{-1}$
Bolshev	0	20,094			4.248×10^{-31}	$10.\times10^{-1}$

Table 5: Number of terms, largest term, and smallest term for $n = 200$.

of terms, the relative magnitude of the terms, and whether negative terms are present. How these three factors will interact is unclear and can only be determined by implementing each formula in arbitrary precision and then comparing the computational time of all formulae.

Using the rational arithmetic implementations of all 13 formulae, Brown and Harvey (2007) found that the DwassAltD formula was the fastest. In developing techniques to implement the 13 formulae in arbitrary precision, the fastest rational arithmetic formula, DwassAltD, will be developed first. This provides a methodology that will be used to develop arbitrary precision implementations for the other formulae.

4. Arbitrary precision implementation of DwassAltD

Utilizing the rational arithmetic version of DwassAltD programmed by Brown and Harvey (2007), an arbitrary precision version is produced by inputing the internal precision ip to be used in all calculations and then replacing the rational arithmetic calculations with arbitrary precision calculations employing the inputed internal precision ip. The Mathematica function `DwassAltDKS1SidedOneSampleRTArbPrecision` contained in Section 1 of the `KS1SidedOneSampleDwassFormulae.nb` file calculates the right tail p value with the DwassAltD formula for test statistic $d^+ =$ `dIn` and sample size $n =$ `sampleSizeIn` using arbitrary precision arithmetic with internal precision $ip =$ `internalPrecisionIn` digits of precision. The inputed test statistic $d^+ =$ `dIn` is converted to a rational arithmetic number $d^+ =$ `d` so that Mathematica will consider the test statistic an exact number. For the test statistic $d^+ = 0.105632$, sample size $n = 100$, and internal precision $ip = 30$, the Mathematica function `DwassAltDKS1SidedOneSampleRTArbPrecision` produces a right tail probability of $P\left[D_{100}^+ \geq d^+ = 0.105632\right] = 0.09997990380077079963347$ which has a resulting precision of $rp = 22$. If the test statistic $d^+ = 0.105632$ is not converted to a rational number, the Mathematica function `DwassAltDKS1SidedOneSampleRTArbPrecisionNonRationalTestStatistic` in Section 1 of the `KS1SidedOneSampleDwassFormulae.nb` file produces a right tail probability of $P\left[D_{100}^+ \geq d^+ = 0.105632\right] = 0.0999799$ which has a resulting precision of $rp = 6$. The reason for this decrease in precision is that Mathematica considers the test statistic $d^+ = 0.105632$ a machine precision number and does all the computations in machine precision. Not converting the test statistic d^+ to a rational number can have very bad consequences. For the test statistic $d^+ = 0.0199760$, sample size $n = 2,000$, and internal precision $ip = 60$, the Mathematica function `DwassAltDKS1SidedOneSampleRTArbPrecision` produces a right tail probability of $P\left[D_{2000}^+ \geq d^+ = 0.0199760\right] = 0.199998648779404946563706459535437273$ which has a resulting precision of $rp = 36$. However, if the test statistic $d^+ = 0.0199760$ is not converted to a rational number, the Mathematica function `DwassAltDKS1SidedOneSampleRTArbPrecisionNonRationalTestStatistic` produces a right tail probability of $P\left[D_{2000}^+ \geq d^+ = 0.0199760\right] = -1.52071 \times 10^6$. These examples also show that the DwassAltD formula has a lot of catastrophic cancellation.

To deal with catastrophic cancellation in DwassAltD, the following six steps are needed to implement DwassAltD in arbitrary precision. First, implement DwassAltD in arbitrary precision where the internal precision ip is inputed. Second, develop a procedure to determine the minimum internal precision mp needed to produce a result with a desired precision dp. Third, determine an upper limit on the sample size n so that the computation time needed to find a p value for the sample size upper limit is around 100 seconds. Fourth, specify a representative set of sample sizes n, p values, and desired precisions dp to generate the resulting minimum precisions mp. Fifth, using the data generated in Step 4, fit a function that will predict the minimum precision mp needed for a particular n, p value, and dp. Sixth, using the fitted function found in Step 5 to predict the internal precision ip, modify the program in Step 1 so that the desired precision dp is inputed instead of ip.

4.1. Minimum precision

In order to get a resulting precision of rp, ip may have to be set to a value that exceeds rp but the user does not know by how much. Thus, the relationship between ip and rp needs

to be investigated so realistic internal precisions ip can be set. Define $rp(F, d^+, n, ip)$ as the resulting precision rp for formula F, sample size n, test statistic d^+, and internal precision ip. The resulting precision $rp(F, d^+, n, ip)$ can be found by running the arbitrary precision method for formula F and then using the Mathematica function `Precision` on the calculated right tail p value $P\left[D_n^+ \geq d^+\right]$ to determine $rp(F, d^+, n, ip)$. Since the Mathematica function `Precision` will often return a non-integer value, the result will be truncated so that resulting precision $rp(F, d^+, n, ip)$ used in this paper will always be an integer.

One way to study the resulting precision function $rp(F, d^+, n, ip)$ is to specify a desired precision dp for the resulting precision rp and then find the minimum internal precision ip needed to produce $rp \geq dp$. Specifically, the user specifies F, d^+, n, and dp and then the minimum internal precision $mp(F, d^+, n, dp)$ is found such that $rp\left(F, d^+, n, mp(F, d^+, n, dp)\right) = dp$ and $rp\left(F, d^+, n, mp(F, d^+, n, dp) - 1\right) < dp$. For example, $mp(DwassAltD, d^+ = 0.0185679, n = 10000, dp = 20) = 128$ because $rp\left(DwassAltD, d^+ = 0.0185679, n = 10000, ip = 127\right) = 19$ and $rp\left(DwassAltD, d^+ = 0.0185679, n = 10000, ip = 128\right) = 20$.

Similarly, $mp(DwassAltD, d^+ = 0.0185679, n = 10000, dp = 100) = 208$ because $rp\left(DwassAltD, d^+ = 0.0185679, n = 10000, ip = 207\right) = 99$ and $rp\left(DwassAltD, d^+ = 0.0185679, n = 10000, ip = 208\right) = 100$. In these two examples, the difference between the minimum internal precision $mp(F, d^+, n, dp)$ and the desired precision dp is the same, $mp(DwassAltD, d^+ = 0.0185679, n = 10000, dp = 20) - dp = 128 - 20 = 108$ and $mp(DwassAltD, d^+ = 0.0185679, n = 10000, dp = 100) - dp = 208 - 100 = 108$. Therefore, the minimum internal precision minus the desired precision function is defined as $mpdp(F, d^+, n, dp) = mp(F, d^+, n, dp) - dp$.

For any d^+, n, dp, and formula F, a search procedure can be used to find the minimum internal precision $mp(F, d^+, n, dp)$. The procedure to find $mp(F, d^+, n, dp)$ has two distinct steps. First, find a lower bound l and an upper bound u so that $l \leq mp(F, d^+, n, dp) \leq u$. Second, use bisection search to find the minimum internal precision $mp(F, d^+, n, dp)$. As seen from the example in the last paragraph, the minimum internal precision $mp(F, d^+, n, dp)$ for DwassAltD can be much greater than the desired precision so an initial estimate of $mp(DwassAltD, d^+, n, dp)$ could significantly reduce the number of iterations. For the DwassAltD formula, preliminary work shows that $\sqrt{n}$ is a good initial estimate.

For DwassAltD, the Mathematica function `DwassAltDKS1SidedOneSampleRTArbPrecision` contained in Section 1 of the `KS1SidedOneSampleDwassFormulae.nb` file will be used to calculate the resulting precision $rp(F, d^+, n, ip)$. The following search procedure to determine the DwassAltD minimum precision $mp(DwassAltD, d^+, n, dp)$ summarizes the Mathematica function `MinPrecisionMinusDesiredPrecisionDwassAltD` contained in Section 2 of the `KS1SidedOneSampleDwassFormulae.nb` file. The function returns the number of iterations `numberTries` and the minimum precision minus the desired precision $mpdp(DwassAltD, d^+, n, dp) = mp(DwassAltD, d^+, n, dp) - dp$.

Procedure to find the DwassAltD minimum precision $mp(DwassAltD, d^+, n, dp)$

Step 1 (initial potential bounds): Set the lower limit l to the desired precision dp, $l = dp$. Set the upper limit and internal precision $u = ip = \lfloor \sqrt{n} + dp \rfloor$ and then run the arbitrary precision DwassAltD method to determine $rp(DwassAltD, d^+, n, ip)$. If $rp(DwassAltD, d^+, n, ip) \geq dp$, go to Step 4. Otherwise, set $l = u$ and go to Step 2.

Step 2 (new potential upper bound): At this point, $mp(DwassAltD, d^+, n, dp) > l$. Set the upper limit u and internal precision ip to the lower limit plus an increment, $u = ip = l + dp - rp(DwassAltD, d^+, n, l)$.

Step 3 (test potential upper bound): Run the arbitrary precision DwassAltD method to determine $rp(DwassAltD, n, nt, ip)$. If $rp(DwassAltD, d^+, n, ip) \geq dp$, go to Step 4. Otherwise, set $l = u$ and go to Step 2.

Step 4 (test bounds): At this point, $l \leq mp(DwassAltD, d^+, n, dp) \leq u$. If $l \geq u - 1$, go to Step 6.

Step 5 (binary search): Set $ip = \left\lfloor \frac{l+u}{2} \right\rfloor$ and run the arbitrary precision DwassAltD method to determine $rp(DwassAltD, d^+, n, ip)$. If $rp(DwassAltD, d^+, n, ip) \geq dp$, set $u = ip$ and go to Step 4. Otherwise, set $l = ip$ and go to Step 4.

Step 6 (determine minimum precision): Run the arbitrary precision DwassAltD method to determine $rp(DwassAltD, n, nt, l)$. If $rp(DwassAltD, d^+, n, l) \geq dp$, set $mp(DwassAltD, d^+, n, dp) = l$ and terminate the procedure. Otherwise, set $mp(DwassAltD, d^+, n, dp) = u$ and terminate the procedure.

For example, the procedure above needed 6 iterations to find the minimum precision $mp\,(DwassAltD, d^+ = 0.0185679, n = 10000, dp = 100) = 208$ and 9 iterations to find $mp\,(DwassAltD, d^+ = 0.00227855, n = 10000, dp = 100) = 114$.

For an arbitrary precision formula F, the minimum precision minus the desired precision $mpdp(F, d^+, n, dp)$ will be found for a representative set of desired precisions dp, sample sizes n, and test statistics d^+. Using this data, a function will be fit to predict the minimum precision minus the desired precision $mpdp(F, d^+, n, dp)$ needed for a specific value of dp, n, and d^+. This function will then be used in the arbitrary precision routine to initially set the internal precision ip.

In considering what would be a good representative set of desired precisions dp, Mathematica uses machine precision rather than arbitrary precision if the internal precision is less than the precision of 16 needed for machine precision numbers. Since Mathematica needs to be forced to use arbitrary precision, the smallest desired precision will be set to $dp = 20$. Using the desired precisions $dp = 20, 40, 100$, the analyses in the rest of the paper found that the minimum precision minus the desired precision $mpdp(F, d^+, n, dp)$ did not vary for the same formula F, the same test statistic d^+, and the same sample size n. In other words, $mpdp(F, d^+, n, dp = 20) = mpdp(F, d^+, n, dp = 40) = mpdp(F, d^+, n, dp = 100)$. Thus, we will not report the minimum precision minus the desired precision $mpdp(F, d^+, n, dp)$ for different values of the desired precision dp.

4.2. Generating test statistics

The representative set of the test statistic d^+ must take into account the fact that for a fixed value of d^+, the p value changes with the sample size n. For $d^+ = 293/5000$, Brown and Harvey (2007) show that for $n = 1,000$, $P[D^+_{1,000} \geq 293/5000] \simeq 0.001$ while for $n = 5,000$, $P[D^+_{5,000} \geq 293/5000] \simeq 1.14493 \times 10^{-15}$. Given the p value for the same test statistic d^+ changes dramatically with n, a different method of determining the representative set for d^+

other than just arbitrarily fixing their values should be used. One method is to set the test statistic d^+ equal to a value that would give a p value close to a specified p value. The approximation of Maag and Dicaire (1971) can be used to find a test statistic d^+ for a sample size n that will yield a p value close to the specified p value α_{MD}. Specifically, this is accomplished by solving the approximation $\alpha_{MD} \simeq \exp\left(\frac{-\left[6nd^+ + 1\right]^2}{18n}\right)$ for d^+ yielding $d^+ \simeq d^+_{MD}(n, \alpha_{MD}) = \sqrt{\frac{\ln(\alpha_{MD})}{-2n}} - \frac{1}{6n}$. The Mathematica code for the Maag and Dicaire (1971) approximation is found in the Mathematica function `KS1SidedOneSampleTestStatisticByMaagDicaire` contained in Section 3 of the `KS1SidedOneSampleDwassFormulae.nb` file. The representative set of test statistics d^+ is generated by using the Maag and Dicaire approximation with the p value representative set $\alpha_{MD} = 0.001, 0.1, 0.5, 0.9$. Table 6 contains the Maag and Dicaire approximation of the test statistic d^+ to six digits of precision for the representative set $\alpha_{MD} = 0.001, 0.1, 0.5, 0.9$ and various sample sizes.

For the Maag and Dicaire test statistic $d^+_{MD}(n, \alpha_{MD})$, let $mp(F, n, dp, \alpha_{MD})$ denote the minimum precision for Formula F, sample size n, desired precision dp, and test statistic $d^+_{MD}(n, \alpha_{MD})$. Then let $mpdp(F, n, dp, \alpha_{MD}) = mp(F, n, dp, \alpha_{MD}) - dp$ denote the minimum precision minus desired precision. After the minimum precision minus the desired precision $mpdp(F, n, dp, \alpha_{MD})$ is determined for the representative set and a particular formula F, a function is fit to the data that predicts the minimum precision mp for any desired precision dp, sample size n, and p value α_{MD}. This function is then put into the arbitrary precision routine to set the internal precision.

4.3. Representative set for DwassAltD

Since the minium precision minus the desired precision $mpdp(F, n, dp, \alpha_{MD})$ does not vary with the desired precision dp, the representative set for dp will be set to $dp = 20$. Section 4.2 specified the representative set of test statistics d^+ as those generated by using the Maag and Dicaire approximation with the p value representative set $\alpha_{MD} = 0.001, 0.1, 0.5, 0.9$. In this section, the representative set of sample sizes n used to compute the minimum precision minus desired precision $mpdp(F, n, dp, \alpha_{MD})$ will be determined differently for each formula F because, as we will see, the computation time varies considerably from one formula to another. For each formula F, a reasonable upper limit on the sample size n will be determined experimentally so that the computation time will not exceed 100 to 200 seconds. Using this upper limit, a representative set of approximately twenty sample sizes will be specified.

To specify a representative set of sample sizes n for DwassAltD, we need to be able to determine the time in seconds needed to calculate the right tail p value for various sample sizes. The Mathematica function `TimingDwassAltDKS1SidedOneSampleRTArbPrecision` contained in Section 4 of the `KS1SidedOneSampleDwassFormulae.nb` file first determines the test statistic $d^+_{MD}(n, \alpha_{MD})$ corresponding to Maag and Dicaire p value α_{MD}. The minimum precision $mp(DwassAltD, n, dp, \alpha_{MD})$ for sample size n, test statistic $d^+_{MD}(n, \alpha_{MD})$, and desired precision dp is calculated. Using test statistic $d^+_{MD}(n, \alpha_{MD})$ and internal precision $ip = mp(DwassAltD, n, dp, \alpha_{MD})$, the time in seconds to calculate the right tail p value for the arbitrary precision DwassAltD formula is found. The sample output in Section 4.1 of the `KS1SidedOneSampleDwassFormulae.nb` file is summarized in Table 7. Since a sample size of $n = 10,000,000$ can have a computation time that exceeds 100 seconds but is less than

Sample size n	Maag and Dicaire test statistic approximation, $d^+_{MD}(n, \alpha_{MD})$ $\alpha_{MD} = 0.001$	$\alpha_{MD} = 0.1$	$\alpha_{MD} = 0.5$	$\alpha_{MD} = 0.9$
100	0.184179	0.105632	0.0572038	0.0212855
300	0.106743	0.0613931	0.0334333	0.0126959
600	0.0755936	0.0435266	0.0237560	0.00909241
1,000	0.0586030	0.0337640	0.0184498	0.00709145
3,000	0.0338751	0.0195343	0.0106927	0.00413492
6,000	0.0239649	0.0138244	0.00757237	0.00293534
10,000	0.0185679	0.0107132	0.00587038	0.00227855
30,000	0.0107243	0.00618931	0.00339333	0.00131959
60,000	0.00758436	0.00437766	0.00240060	0.000934241
100,000	0.00587530	0.00339140	0.00185998	0.000724145
300,000	0.00339251	0.00195843	0.00107427	0.000418492
600,000	0.00239899	0.00138494	0.000759737	0.000296034
1,000,000	0.00185829	0.00107282	0.000588538	0.000229355
2,000,000	0.00131405	0.00075863	0.000416194	0.000162213
3,000,000	0.00107293	0.000619431	0.000339833	0.000132459
4,000,000	0.000929189	0.000536450	0.000294311	0.000114719
5,000,000	0.000831096	0.000479819	0.000263244	0.000102612
6,000,000	0.000758686	0.000438016	0.000240310	0.0000936741
7,000,000	0.000702408	0.000405526	0.000222486	0.0000867273
8,000,000	0.000657044	0.000379336	0.000208118	0.0000811274
9,000,000	0.000619469	0.000357642	0.000196216	0.0000764887
10,000,000	0.000587680	0.000339290	0.000186148	0.0000725645

Table 6: Maag and Dicaire test statistic approximation $d^+_{MD}(n, \alpha_{MD})$ to six digits of precision.

200 seconds, we will set $n = 10,000,000$ as the sample size upper limit for the DwassAltD formula. The sample size representative set will then be the twenty-two sample sizes listed in Table 6.

4.4. Minimum precision minus desired precision DwassAltD data

The minimum precision minus desired precisions in Tables 8 and 9 were produced by the Mathematica function `MinPrecisionMinusDesiredPrecisionToFileDwassAltD` contained in Section 5 of the `KS1SidedOneSampleDwassFormulae.nb` file which writes the minimum pre-

Sample size n	$\alpha_{MD} = 0.001$, $dp = 20$ Minimum precision	Time in seconds	$\alpha_{MD} = 0.001$, $dp = 100$ Minimum precision	Time in seconds	$\alpha_{MD} = 0.9$, $dp = 20$ Minimum precision	Time in seconds
1,000,000	1,058	3.828	1,138	4.062	150	1.219
5,000,000	2,334	42.985	2,414	44.453	308	8.938
10,000,000	3,289	110.938	3,369	110.390	426	22.640

Minimum precision is $mp(DwassAltD, n, dp, \alpha_{MD})$

Computational times on a 3.4 GHz Pentium IV

Table 7: DwassAltD time in seconds to compute a p value.

cision minus desired precisions $mpdp(DwassAltD, n, dp, \alpha_{MD})$ to an output file for a set of specified sample sizes n, a set of desired precisions dp, and a set of Maag and Dicaire approximate test statistics $d^{+}_{MD}(n, \alpha_{MD})$ with p values α_{MD}.

4.5. Predicting the minimum precision for DwassAltD

In constructing a fitted function to predict $mpdp(DwassAltD, n, dp, \alpha_{MD})$, a tradeoff exists between how close the predicted value is to the actual value and whether the predicted value is less than the actual value. If the predicted value is less than the actual value, then the internal precision will produce a p value whose resulting precision rp is less than the specified desired precision dp. When $rp < dp$, the internal precision ip must be increased and the p value recalculated. The fitted function can be modified to insure that almost no predicted values are less than the actual values by adding a constant value to the original fitted function. Thus, there is a tradeoff between living with the chance of having to calculate the p value more than once or adding a constant value to the original fitted function to almost eliminate the possibility of that chance. To study this tradeoff, we need to know the relative computer times needed to calculate the p value again versus the computer time needed to calculate the p value with a larger internal precision ip. Table 8 shows this tradeoff by tabulating the computer time needed for DwassAltD to calculate p values for $\alpha_{MD} = 0001$ and various internal precisions, $ip = mpdp(DwassAltD, n, dp, \alpha_{MD} = 0.001) + dp$. This table also shows that the computer time needed to calculate a p value is much greater than the additional time needed for a slightly larger internal precision (ip increased by 2 to 3). Thus, in producing a function to predict $mpdp(F, n, dp, \alpha_{MD})$, an original fitted function is first created and then a constant value is added to the original fitted function to create the final fitted function so that a predicted value produced by the final fitted function will almost always be greater than or equal to the actual value.

4.6. Predicting internal precision for the DwassAltD formula

To investigate how $mpdp(DwassAltD, n, dp, \alpha_{MD})$ changes with the p value α_{MD}, define the proportion $P001mpdp(DwassAltD, n, dp, \alpha_{MD})$ shown in Equation (1) below.

DwassAltD sample size n	$mpdp$	Time in seconds for DwassAltD to calculate an $\alpha_{MD} = 0.001$ p value using internal precision $ip = mpdp + dp$ $dp = 20$	$dp = 40$	$dp = 60$	$dp = 80$	$dp = 100$	$dp = 200$
$10,000$	108	0.016	0.015	0.016	0.016	0.016	0.031
$100,000$	332	0.187	0.204	0.218	0.235	0.218	0.297
$1,000,000$	1,038	3.828	3.984	4.093	4.063	4.000	4.516
$10,000,000$	3,269	72.468	93.438	98.781	109.844	110.172	113.234

$mpdp$ is the mimimum precision minus desired precision

$$mpdp = mpdp(DwassAltD, n, dp, \alpha_{MD} = 0.001)$$

Note: timings on a Pentium IV running at 3.4 GHz.

Table 8: Time in seconds for the DwasssAltD formula to calculate a p value for various desired precisions dp.

$$P001mpdp(DwassAltD, n, dp, \alpha_{MD}) = \frac{mpdp(DwassAltD, n, dp, \alpha_{MD})}{mpdp(DwassAltD, n, dp, \alpha_{MD} = 0.001)}. \quad (1)$$

Table 9 shows that proportion $P001mpdp(DwassAltD, n, dp, \alpha_{MD})$ does not change much as the sample size n varies. This suggests a two stage process for fitting a function to predict $mpdp(DwassAltD, n, dp, \alpha_{MD})$. First, $mpdp(DwassAltD, n, dp, \alpha_{MD} = 0.001)$ is predicted by a fitted function using n as the independent variable. Second, fit a function to predict the proportion $P001mpdp(DwassAltD, n, dp, \alpha_{MD})$ for some n (say $n = 1,000,000$) by using α_{MD} as the independent variable. Then, multiply the two fitted functions to obtain a function to predict $mpdp(DwassAltD, n, dp, \alpha_{MD})$.

For the first step, the fitted function $mpdp(DwassAltD, n, dp, \alpha_{MD} = 0.001) \simeq 4.83943059 + 1.03244211\sqrt{n}$ was found by using stepwise regression with a variety of independent variables that are all functions of n. Rounding up the coefficients yields the predicting function shown in Equation (2) below.

$$mpdp(DwassAltD, n, dp, \alpha_{MD} = 0.001) \simeq 4.84 + 1.033\sqrt{n} \quad (2)$$

To determine if the predicting function in Equation (2) is a good predictor, additional points for other sample sizes between the ones used to derive the function were calculated and then the function was used to predict the actual $mpdp(DwassAltD, n, dp, \alpha_{MD} = 0.001)$. The results are summarized in Table 10. The differences between the actual and predicting $mpdp(DwassAltD, n, dp, \alpha_{MD} = 0.001)$ show that the predicted function in Equation (2) is a very good fit.

For the second step with $n = 1,000,000$, a function is fitted to predict the proportion $P001mpdp(DwassAltD, n = 1000000, dp, \alpha_{MD})$ by using α_{MD} as the independent variable. Using the data in the first two columns of Table 11, stepwise regression with a variety

Sample size n	$mpdp(DwassAltD, n, dp, \alpha_{MD})$ $\alpha_{MD} =$ 0.001	0.1	0.5	0.9	$P001mpdp(DwassAltD, n, dp, \alpha_{MD})$ $\alpha_{MD} =$ 0.001	0.1	0.5	0.9
100	14	8	5	2	1	0.571429	0.357143	0.142857
300	22	13	7	3	1	0.590909	0.318182	0.136364
600	30	17	10	4	1	0.566667	0.333333	0.133333
1,000	37	21	12	5	1	0.567568	0.324324	0.135135
3,000	61	35	20	8	1	0.573770	0.327869	0.131148
6,000	85	49	28	11	1	0.576471	0.329412	0.129412
10,000	108	63	35	14	1	0.583333	0.324074	0.129620
30,000	184	107	59	24	1	0.581522	0.320652	0.130435
60,000	258	149	83	33	1	0.577519	0.321705	0.127907
100,000	332	192	106	42	1	0.578313	0.319277	0.126506
300,000	571	330	182	72	1	0.577933	0.318739	0.126095
600,000	805	465	256	101	1	0.577630	0.318012	0.125466
1,000,000	1,038	600	330	130	1	0.578035	0.317919	0.125241
2,000,000	1,465	847	466	183	1	0.578157	0.318089	0.124915
3,000,000	1,793	1,036	570	224	1	0.577803	0.317903	0.124930
4,000,000	2,070	1,196	657	258	1	0.577778	0.317391	0.124638
5,000,000	2,314	1,337	735	288	1	0.577787	0.317632	0.124450
6,000,000	2,534	1,464	804	315	1	0.577743	0.317285	0.124309
7,000,000	2,736	1,581	869	340	1	0.577851	0.317617	0.124269
8,000,000	2,925	1,690	928	364	1	0.577778	0.317265	0.124444
9,000,000	3,102	1,792	984	385	1	0.577692	0.317215	0.124113
10,000,000	3,269	1,889	1,037	406	1	0.577853	0.317222	0.124197

Table 9: DwassAltD minimum precision minus desired precision proportion.

of independent variables was used to find the fitted function $P001mpdp(DwassAltD, n = 1000000, dp, \alpha_{MD}) \simeq 0.61058331 - 0.09039563\alpha_{MD} - 0.35661608\alpha_{MD}^{1/3} - 0.09981378\alpha_{MD}^{4} - 0.06160217 \ln \alpha_{MD}$ where $ln(\alpha_{MD})$ is the natural logarithm (base e) of α_{MD}. Rounding the coefficients yields the predicting function shown in Equation (3) below.

$$P001mpdp(DwassAltD, n = 1000000, dp, \alpha_{MD}) \simeq \begin{array}{l} 0.611 - 0.0904\alpha_{MD} - 0.357\alpha_{MD}^{1/3} \\ -.0998\alpha_{MD}^{4} - 0.0616 \ln \alpha_{MD} \end{array} \quad (3)$$

Sample size n	$mpdp(DwassAltD, n, dp, \alpha_{MD} = 0.001)$ Actual	Predicted	Predicted minus actual
50	11	12.14	1.14
200	19	19.45	0.45
800	33	34.06	1.06
2,000	51	51.04	0.04
4,500	74	74.14	0.14
8,000	97	97.23	0.23
20,000	151	150.93	−0.07
45,000	224	223.97	−0.03
80,000	297	297.02	0.02
200,000	467	466.81	−0.19
450,000	698	697.80	−0.20
800,000	929	928.78	−0.22
1,500,000	1,270	1,270.00	0.00
2,500,000	1,638	1,638.16	0.16
3,500,000	1,937	1,937.41	0.41
4,500,000	2,195	2,196.16	1.16
5,500,000	2,426	2,427.44	1.44
6,500,000	2,637	2,638.48	1.48
7,500,000	2,832	2,833.83	1.83
8,500,000	3,015	3,016.53	1.53
9,500,000	3,187	3,188.76	1.76
10,500,000	3,350	3,352.14	2.14

Predicted: $4.84 + 1.033\sqrt{n}$

Table 10: DwassAltD prediction of $mpdp(DwassAltD, n, dp, \alpha_{MD} = 0.001)$.

The differences reported in Table 11 between the actual and predicted proportions show that the predicted function in Equation (3) is a very good fit.

The first and second fitted functions in Equations (2) and (3) are multiplied together to get an initial function that predicts $mpdp(DwassAltD, n, dp, \alpha_{MD})$. The resultant formula denoted by $mpdpInitPred(DwassAltD, n, dp, \alpha_{MD})$ is shown in Equation (4) below.

Probability α_{MD}	*P001mpdp*(*DwassAltD*, $n = 1000000, dp, \alpha_{MD}$) Actual proportion	Predicted proportion	Predicted minus actual
0.001	1.00000	1.00073	0.00073
0.002	0.94798	0.94866	0.00068
0.003	0.91715	0.91708	−0.00006
0.004	0.89403	0.89409	0.00006
0.005	0.87572	0.87588	0.00016
0.006	0.86031	0.86073	0.00042
0.007	0.84778	0.84773	−0.00006
0.008	0.83622	0.83630	0.00008
0.009	0.82563	0.82610	0.00047
0.01	0.81696	0.81686	−0.00009
0.02	0.75241	0.75327	0.00086
0.03	0.71291	0.71336	0.00045
0.04	0.68304	0.68357	0.00053
0.05	0.65896	0.65950	0.00054
0.06	0.63873	0.63912	0.00039
0.07	0.62042	0.62135	0.00093
0.08	0.60501	0.60552	0.00051
0.09	0.59056	0.59120	0.00064
0.1	0.57803	0.57808	0.00005
0.11	0.56551	0.56596	0.00045
0.12	0.55491	0.55465	−0.00026
0.13	0.54432	0.54405	−0.00027
0.14	0.53372	0.53405	0.00033
0.15	0.52505	0.52457	−0.00048
0.175	0.50289	0.50277	−0.00012
0.2	0.48362	0.48313	−0.00050
0.3	0.41811	0.41825	0.00014
0.4	0.36513	0.36569	0.00056
0.5	0.31792	0.31891	0.00099
0.6	0.27360	0.27419	0.00058
0.7	0.22929	0.22875	−0.00054
0.8	0.18112	0.18014	−0.00098
0.9	0.12524	0.12597	0.00073

$$\text{Predicted Proportion} = 0.611 - 0.0904\alpha_{MD} - 0.357\alpha_{MD}^{1/3} - .0998\alpha_{MD}^{4} - 0.0616 \ln \alpha_{MD}$$

Table 11: Prediction of *P001mpdp*(*DwassAltD*, $n = 1000000, dp, \alpha_{MD}$) by α_{MD}.

$$\begin{aligned} \alpha_{MD} &= \exp\left(-2(d^+)^2 n - \frac{2d^+}{3} - \frac{1}{18n}\right) \\ mpdpInitPred(DwassAltD, n, dp, \alpha_{MD}) &= (4.84 + 1.033\sqrt{n}) \\ &\quad \times (0.611 - 0.0904\alpha_{MD} - 0.357\alpha_{MD}^{1/3} \\ &\quad -0.0998\alpha_{MD}^4 - 0.0616 \ln \alpha_{MD}) \end{aligned} \tag{4}$$

The predictive ability of Equation (4) is tested by applying it to the samples sizes n and p values α_{MD} in Table 12 which are different than the sample sizes n and p values used to construct the approximation in Equation (4). Table 12 shows that the initial predicted values $mpdpInitPred(DwassAltD, n, dp, \alpha_{MD})$ are close to the actual values $mpdp(DwassAltD, n, dp, \alpha_{MD})$. Let $mpdpIPError(DwassAltD, n, dp, \alpha_{MD})$ defined in Equation (5) below represent the error in the initial predicted values.

$$\begin{aligned} mpdpIPError(DwassAltD, n, dp, \alpha_{MD}) &= mpdpInitPred(DwassAltD, n, dp, \alpha_{MD}) \\ &\quad - mpdp(DwassAltD, n, dp, \alpha_{MD}) \end{aligned} \tag{5}$$

Table 13 contains the initial prediction error $mpdpIPError(DwassAltD, n, dp, \alpha_{MD})$ for the sample sizes n and p values α_{MD} in Table 12.

Since the largest negative error in Table 13 is -1.30, construct the final prediction by conservatively adding 3 to the initial prediction $mpdpInitPred(DwassAltD, n, dp, \alpha_{MD})$ and then round up. The resultant final prediction $mpdpFinalPred(DwassAltD, n, dp, \alpha_{MD})$ is shown in Equation (6) below where $\lceil x \rceil$ is the smallest integer greater than or equal to x.

$$\begin{aligned} \alpha_{MD} &= \exp\left(-2(d^+)^2 n - \frac{2d^+}{3} - \frac{1}{18n}\right) \\ mpdpFinalPred(DwassAltD, n, dp, \alpha_{MD}) &= \lceil 3 + (4.84 + 1.033\sqrt{n}) \\ &\quad \times (0.611 - 0.0904\alpha_{MD} - 0.357\alpha_{MD}^{1/3} \\ &\quad -0.0998\alpha_{MD}^4 - 0.0616 \ln \alpha_{MD}) \rceil \end{aligned} \tag{6}$$

Let $mpdpFPError(DwassAltD, n, dp, \alpha_{MD})$ defined in Equation (7) below represent the error in the final predicted values.

$$\begin{aligned} mpdpFPError(DwassAltD, n, dp, \alpha_{MD}) &= mpdpFinalPred(DwassAltD, n, dp, \alpha_{MD}) \\ &\quad - mpdp(DwassAltD, n, dp, \alpha_{MD}) \end{aligned} \tag{7}$$

To determine how good a predictor Equation (6) is, the final predicted error $mpdpFPError(DwassAltD, n, dp, \alpha_{MD})$ is calculated for the sample sizes n and p values α_{MD} in Table 14. The results in Table 14 show that $mpdpFinalPred(DwassAltD, n, dp, \alpha_{MD})$ overestimates $mpdp(DwassAltD, n, dp, \alpha_{MD})$ especially for $\alpha_{MD} = 0.0005$ and $\alpha_{MD} = 0.95$ which is outside of the range of the p values α_{MD} used to fit the functions to construct $mpdpFinalPred(DwassAltD, n, dp, \alpha_{MD})$ in Equation (6).

By altering the arbitrary version of the DwassAltD formula so that the formulae in Equation (6) are used to set the internal precision $ip = mpdpFinalPred(DwassAltD, n, dp, \alpha_{MD}) + dp$,

Sample size n	Actual $mp - dp$ $mpdp(DwassAltD, n, dp, \alpha_{MD})$ $\alpha_{MD} =$ 0.01	$\alpha_{MD} =$ 0.3	$\alpha_{MD} =$ 0.7	Initial prediction of $mp - dp$ $mpdpInitPred(DwassAltD, n, dp, \alpha_{MD})$ $\alpha_{MD} =$ 0.01	$\alpha_{MD} =$ 0.3	$\alpha_{MD} =$ 0.7
50	9	4	3	9.63	4.93	2.70
200	15	8	4	15.59	7.98	4.37
800	27	14	8	27.53	14.09	7.71
2,000	41	21	12	41.40	21.20	11.59
4,500	60	31	18	60.26	30.86	16.88
8,000	79	41	23	79.13	40.52	22.16
20,000	123	64	35	122.99	62.97	34.44
45,000	183	94	52	182.66	93.53	51.15
80,000	243	125	69	242.33	124.08	67.86
200,000	381	196	108	381.03	195.09	106.70
450,000	570	292	160	569.71	291.70	159.54
800,000	759	389	213	758.39	388.31	212.37
1,500,000	1,037	531	290	1,037.12	531.03	290.43
2,500,000	1,337	685	374	1,337.85	685.01	374.64
3,500,000	1,581	810	442	1,582.30	810.17	443.09
4,500,000	1,793	918	501	1,793.67	918.39	502.28
5,500,000	1,981	1,014	553	1,982.59	1,015.12	555.19
6,500,000	2,153	1,102	601	2,154.98	1,103.39	603.46
7,500,000	2,313	1,184	646	2,314.55	1,185.09	648.15
8,500,000	2,462	1,260	687	2,463.80	1,261.51	689.94
9,500,000	2,602	1,332	726	2,604.48	1,333.54	729.34
10,500,000	2,736	1,400	763	2,737.94	1,401.88	766.71

$$\alpha_{MD} = \exp\left(-2(d^+)^2 n - \frac{2d^+}{3} - \frac{1}{18n}\right)$$

$$mpdpInitPred(DwassAltD, n, dp, \alpha_{MD}) = (4.84 + 1.033\sqrt{n}) \times (0.611 - 0.0904\alpha_{MD} - 0.357\alpha_{MD}^{1/3} - 0.0998\alpha_{MD}^{4} - 0.0616 \ln \alpha_{MD})$$

Table 12: DwassAltD, initial prediction of $mpdp(DwassAltD, n, dp, \alpha_{MD})$.

Sample size n	Initial prediction error $mpdpIPError(DwassAltD, n, dp, \alpha_{MD})$ $\alpha_{MD} =$ 0.01	$\alpha_{MD} =$ 0.3	$\alpha_{MD} =$ 0.7
50	0.63	0.93	−0.30
200	0.59	−0.02	0.37
800	0.53	0.09	−0.29
2,000	0.40	0.20	−0.41
4,500	0.26	−0.14	−1.12
8,000	0.13	−0.48	−0.84
20,000	−0.01	−1.03	−0.56
45,000	−0.34	−0.47	−0.85
80,000	−0.67	−0.92	−1.14
200,000	0.03	−0.91	−1.30
450,000	−0.29	−0.30	−0.46
800,000	−0.61	−0.69	−0.63
1,500,000	0.12	0.03	0.43
2,500,000	0.85	0.01	0.64
3,500,000	1.30	0.17	1.09
4,500,000	0.67	0.39	1.28
5,500,000	1.59	1.12	2.19
6,500,000	1.98	1.39	2.46
7,500,000	1.55	1.09	2.15
8,500,000	1.79	1.51	2.94
9,500,000	2.48	1.54	3.34
10,500,000	1.94	1.88	3.71

$$mpdpIPError(DwassAltD, n, dp, \alpha_{MD}) = mpdpInitPred(DwassAltD, n, dp, \alpha_{MD}) - mpdp(DwassAltD, n, dp, \alpha_{MD})$$

Table 13: DwassAltD, initial prediction error for $mpdp(DwassAltD, n, dp, \alpha_{MD})$.

the Mathematica function `DwassAltDKS1SidedOneSampleRTdesiredPrecision` contained in Section 6 of the `KS1SidedOneSampleDwassFormulae.nb` file computes the p value, right tail probability $P(D_n^+ \geq d^+)$, to any desired digits of precision. In the unlikely event that the

Sample size n	Final prediction error $mpdpFPError(DwassAltD, n, dp, \alpha_{MD})$ $\alpha_{MD} =$ 0.0005	0.005	0.05	0.15	0.25	0.6	0.8	0.95
50	4	4	4	4	4	4	4	4
200	4	4	4	4	4	4	3	3
800	4	4	4	3	4	3	3	4
2,000	4	4	4	3	3	3	3	3
4,500	3	3	3	3	3	3	3	4
8,000	3	3	3	3	3	3	3	4
20,000	4	3	3	2	3	3	3	4
45,000	3	3	3	3	3	3	2	4
80,000	3	3	3	3	3	3	2	5
200,000	4	3	3	3	3	3	3	6
450,000	4	3	3	3	2	4	2	8
800,000	5	4	4	3	3	4	2	11
1,500,000	6	4	4	4	3	5	2	13
2,500,000	7	4	5	4	3	6	3	18
3,500,000	8	4	5	4	3	7	2	21
4,500,000	8	4	6	4	3	7	2	23
5,500,000	9	4	6	5	4	7	3	26
6,500,000	10	4	6	4	4	8	3	27
7,500,000	10	4	6	5	4	8	3	30
8,500,000	11	5	7	5	4	8	3	31
9,500,000	11	5	7	5	5	9	3	33
10,500,000	12	5	7	5	5	10	3	34

$mpdpFPError(DwassAltD, n, dp, \alpha_{MD}) =$
$mpdpFinalPred(DwassAltD, n, dp, \alpha_{MD})$
$-mpdp(DwassAltD, n, dp, \alpha_{MD})$

Table 14: DwassAltD, final prediction error for $mpdp(DwassAltD, n, dp, \alpha_{MD})$.

resulting precision rp of the p value is less than the desired precision dp, $rp < dp$, the internal precision is increased by $dp - rp$ and the p value is recalculated until $rp \geq dp$.

5. Arbitrary precision implementation methodology

The techniques used in Secton 4 for DwassAltD can be used to develop an arbitrary precision implementation for any formula F. A summary of the general methodology is shown below.

Methodology to implement a formula F in arbitrary precision:

1. Utilizing the rational arithmetic version of formula F programmed by Brown and Harvey (2007) and modified in Section 3, an arbitrary precision version is produced by inputing the internal precision ip to be used in all calculations and then replacing the rational arithmetic calculations with arbitrary precision calculations employing the inputed internal precision ip.

2. For formula F, develop a procedure to determine the minimum precision $mp(F, d^+, n, dp)$ and the minimum precision minus desired precision $mpdp(F, d^+, n, dp)$ by modifying the procedure in Section 4.1. Specifically, use the arbitrary precision version for formula F developed in Step 1 to develop an initial estimate of the minimum precision $mp(F, d^+, n, dp)$.

3. Specify a representative set of sample sizes n for formula F by experimentally determining the time in seconds needed to calculate the right tail p value for various sample sizes. Select the sample size upper limit on n so that the computation time needed for the upper limit is around 100 seconds. Then, select about twenty sample sizes between 10 and the sample size upper limt as the representative set of sample sizes n for formula F.

4. Since the minium precision minus the desired precision $mpdp(F, n, dp, \alpha_{MD})$ does not vary with the desired precision dp, the representative set for dp will be set to $dp = 20$. Specify the representative set of desired precisions as $dp = 20$. Specify the representative set of test statistics d^+ as those generated by using the Maag and Dicaire approximation with the p value representative set $\alpha_{MD} = 0.001, 0.1, 0.5, 0.9$.

5. Find the minimum precision minus desired precisions $mpdp(F, n, dp, \alpha_{MD})$ for the representative set. Fit a function to this data that will predict $mpdp(F, n, dp, \alpha_{MD})$.

6. Using the fitted function found in Step 5, modify the program developed in Section 4.6 for formula F.

6. Arbitrary precision implementation of Dwass-based formulae

Section 4 implemented the DwassAltD formula in arbitrary precision and Section 5 summarized the methodology used in Section 4. This methodology will now be used to implement all the other formulae in arbitrary precision. This section implements the remaining Dwass-based formulae (DwassD, DwassI, DwassAltI) in arbitrary precision, compares computed right tail probabilities of all Dwass-based formulae to make sure there are no implementation errors, and runs computational experience to determine the fastest formula.

The four Mathematica functions needed in the methodology to implemement each Dwass-based formula are listed in Table 15 along with the section number where they are found

Function type Formula name	Mathematica function name	Section number
Arbitrary precision formula (Methodology: Step 1 Result)		
DwassAltD	`DwassAltDKS1SidedOneSampleRTArbPrecision`	1
DwassD	`DwassDKS1SidedOneSampleRTArbPrecision`	7
DwassI	`DwassIKS1SidedOneSampleRTArbPrecision`	12
DwassAltI	`DwassAltIKS1SidedOneSampleRTArbPrecision`	17
Minimum precision minus desired precision (Methodology: Used in Step 2)		
DwassAltD	`MinPrecisionMinusDesiredPrecisionDwassAltD`	2
DwassD	`MinPrecisionMinusDesiredPrecisionDwassD`	8
DwassI	`MinPrecisionMinusDesiredPrecisionDwassI`	13
DwassAltI	`MinPrecisionMinusDesiredPrecisionDwassAltI`	18
Calculation time (Methodology: Used in Step 3)		
DwassAltD	`TimingDwassAltDKS1SidedOneSampleRTArbPrecision`	4
DwassD	`TimingDwassDKS1SidedOneSampleRTArbPrecision`	9
DwassI	`TimingDwassIKS1SidedOneSampleRTArbPrecision`	14
DwassAltI	`TimingDwassAltIKS1SidedOneSampleRTArbPrecision`	19
Minimum precision minus desired precision To file (Methodology: Used in Step 5)		
DwassAltD	`MinPrecisionMinusDesiredPrecisionToFileDwassAltD`	5
DwassD	`MinPrecisionMinusDesiredPrecisionToFileDwassD`	10
DwassI	`MinPrecisionMinusDesiredPrecisionToFileDwassI`	15
DwassAltI	`MinPrecisionMinusDesiredPrecisionToFileDwassAltI`	20
Desired precision function (Methodology: Step 6 Result)		
DwassAltD	`DwassAltDKS1SidedOneSampleRTdesiredPrecision`	6
DwassD	`DwassDKS1SidedOneSampleRTdesiredPrecision`	11
DwassI	`DwassIKS1SidedOneSampleRTdesiredPrecision`	16
DwassAltI	`DwassAltIKS1SidedOneSampleRTdesiredPrecision`	21

Functions listed in file `KS1SidedOneSampleDwassFormulae.nb`

Table 15: Mathematica function names for Dwass-based formulae.

in file `KS1SidedOneSampleDwassFormulae.nb`. The remainder of this section will use these Mathematica functions and the methodology to produce the desired precision function for each Dwass-based formula which is also listed in Table 15.

Steps 1 through 4 of the methodology are basically the same for all Dwass-based formulae (DwassAltD, DwassD, DwassI, and DwassAltI) so we will use the initial estimate of the minimum precision $mp(F, d^+, n, dp) = 4.84 + 1.033\sqrt{n}$ in Step 2 and the representative set in

Step 4 that was developed for DwassAltD in Section 4. Step 5 of the methodology produces the minimum precision minus desired precision $mpdp(F, n, dp, \alpha_{MD})$ data contained in Table 16 for the DwassD, DwassAltD, DwassI, and DwassAltI formulae respectively. Since the minimum precision minus desired precision $mpdp(F, n, dp, \alpha_{MD})$ data for DwassD and DwassAltD in Table 16 are identical, the same fitted function found for DwassAltD can also be used for DwassD in Step 6 of the methodology to produce the desired precision function for DwassD, Mathematica function `DwassDKS1SidedOneSampleRTdesiredPrecision` listed in Section 11 of the `KS1SidedOneSampleDwassFormulae.nb` file.

Similarly, Table 16 shows that $mpdp(F, n, dp, \alpha_{MD})$ is also the same for the DwassI and DwassAltI (Dwass iterative formulae) so the same fitted function to be developed below can be used for both Dwass iterative formulae.

6.1. Predicting internal precision for the Dwass iterative formulae

To investigate how the minimum precision minus desired precision $mpdp(DwassI, n, dp, \alpha_{MD})$ changes with the p value α_{MD}, define the proportion $P001mpdp(DwassI, n, dp, \alpha_{MD})$ shown in Equation (8) below.

$$P001mpdp(DwassI, n, dp, \alpha_{MD}) = \frac{mpdp(DwassI, n, dp, \alpha_{MD})}{mpdp(DwassI, n, dp, \alpha_{MD} = 0.001)}. \tag{8}$$

Table 17 shows that the proportion $P001mpdp(DwassI, n, dp, \alpha_{MD})$ does not change much as the sample size n varies. This suggests a two stage process for fitting a function to predict $mpdp(DwassI, n, dp, \alpha_{MD})$. First, fit a function to predict $mpdp(DwassI, n, dp, \alpha_{MD} = 0.001)$ by using n as the independent variable. Second, fit a function to predict the proportion $P001mpdp(DwassI, n, dp, \alpha_{MD})$ for some n (say $n = 1,000,000$) by using α_{MD} as the independent variable. Then, multiply the two fitted functions to obtain a function to predict $mpdp(DwassI, n, dp, \alpha_{MD})$.

Using the data in the first two columns of Table 17 for the first step, stepwise regression with a variety of independent variables was used to find fitted function $mpdp(DwassI, n, dp, \alpha_{MD} = 0.001) \simeq 2.98591930 + 1.03190981\sqrt{n}$. Rounding up the coefficients yields the predicting function shown in Equation (9) below.

$$mpdp(DwassI, n, dp, \alpha_{MD} = 0.001) \simeq 2.986 + 1.032\sqrt{n} \tag{9}$$

To determine if the predicting function in Equation (9) is a good predictor, additional points for other sample sizes between the ones used to derive the function were calculated and then the function was used to predict the actual $mpdp(DwassI, n, dp, \alpha_{MD} = 0.001)$. The results are summarized in Table 18. The differences between the actual and predicting $mpdp(DwassI, n, dp, \alpha_{MD} = 0.001)$ show that the predicted function in Equation (9) is a very good fit.

For the second step with $n = 1,000,000$, a function is fitted to predict the proportion $P001mpdp(DwassI, n = 1000000, dp, \alpha_{MD})$ by using α_{MD} as the independent variable. Using the data in the first two columns of Table 19, stepwise regression with a variety of variables was used to find the fitted function $P001mpdp(DwassI, n = 1000000, dp, \alpha_{MD}) \simeq 0.68749934 - 0.14334219\alpha_{MD} - 0.399053723\alpha_{MD}^{1/4} + 0.047033054455\alpha_{MD}^{3} - 0.132298062\alpha_{MD}^{4} -$

Sample size n	$mpdp(F, n, dp, \alpha_{MD})$ DwassD and DwassAltD $\alpha_{MD} =$ 0.001	0.1	0.5	0.9	DwassI and DwassAltI $\alpha_{MD} =$ 0.001	0.1	0.5	0.9
100	14	8	5	2	13	7	3	1
300	22	13	7	3	21	11	6	1
600	30	17	10	4	28	16	8	2
1,000	37	21	12	5	36	20	10	3
3,000	61	35	20	8	60	34	18	6
6,000	85	49	28	11	83	47	25	9
10,000	108	63	35	14	106	61	33	12
30,000	184	107	59	24	182	104	57	21
60,000	258	149	83	33	256	147	80	30
100,000	332	192	106	42	329	190	103	39
300,000	571	330	182	72	568	327	179	69
600,000	805	465	256	101	802	463	253	98
1,000,000	1,038	600	330	130	1,035	597	327	127
2,000,000	1,465	847	466	183	1,462	844	462	179
3,000,000	1,793	1,036	570	224	1,790	1,033	566	220
4,000,000	2,070	1,196	657	258	2,067	1,193	654	254
5,000,000	2,314	1,337	735	288	2,310	1,333	731	284
6,000,000	2,534	1,464	804	315	2,531	1,460	801	311
7,000,000	2,736	1,581	869	340	2,733	1,577	865	336
8,000,000	2,925	1,690	928	364	2,922	1,686	925	360
9,000,000	3,102	1,792	984	385	3,099	1,788	981	381
10,000,000	3,269	1,889	1,037	406	3,266	1,885	1,034	402

Table 16: Minimum precision minus desired precision for Dwass formulae.

$0.055612206 \ln \alpha_{MD}$ where $ln(\alpha_{MD})$ is the logarithm base e of α_{MD}. Rounding the coefficients yields the predicting function shown in Equation (10) below.

$$\begin{aligned} P001mpdp(DwassI, n = 1000000, dp, \alpha_{MD}) \simeq \quad & 0.6875 - 0.1433\alpha_{MD} - 0.399\alpha_{MD}^{1/4} \\ & +0.04703\alpha_{MD}^{3} - 0.1323\alpha_{MD}^{4} \\ & -0.0556 \ln \alpha_{MD} \end{aligned} \tag{10}$$

The differences between the actual and predicted $P001mpdp(DwassI, n = 1000000, dp, \alpha_{MD})$

Sample size n	$mpdp(DwassI, n, dp, \alpha_{MD})$ $\alpha_{MD} =$ 0.001	0.1	0.5	0.9	$P001mpdp(DwassI, n, dp, \alpha_{MD})$ $\alpha_{MD} =$ 0.001	0.1	0.5	0.9
100	13	7	3	1	1	0.53846	0.23077	0.07692
300	21	11	6	1	1	0.52381	0.28571	0.04762
600	28	16	8	2	1	0.57143	0.28571	0.07143
1,000	36	20	10	3	1	0.55556	0.27778	0.08333
3,000	60	34	18	6	1	0.56667	0.30000	0.10000
6,000	83	47	25	9	1	0.56627	0.30120	0.10843
10,000	106	61	33	12	1	0.57547	0.31132	0.11321
30,000	182	104	57	21	1	0.57143	0.31319	0.11538
60,000	256	147	80	30	1	0.57422	0.31250	0.11719
100,000	329	190	103	39	1	0.57751	0.31307	0.11854
300,000	568	327	179	69	1	0.57570	0.31514	0.12148
600,000	802	463	253	98	1	0.57731	0.31546	0.12219
1,000,000	1,035	597	327	127	1	0.57681	0.31594	0.12271
2,000,000	1,462	844	462	179	1	0.57729	0.31601	0.12244
3,000,000	1,790	1,033	566	220	1	0.57709	0.31620	0.12291
4,000,000	2,067	1,193	654	254	1	0.57716	0.31640	0.12288
5,000,000	2,310	1,333	731	284	1	0.57706	0.31645	0.12294
6,000,000	2,531	1,460	801	311	1	0.57685	0.31648	0.12288
7,000,000	2,733	1,577	865	336	1	0.57702	0.31650	0.12294
8,000,000	2,922	1,686	925	360	1	0.57700	0.31656	0.12320
9,000,000	3,099	1,788	981	381	1	0.57696	0.31655	0.12294
10,000,000	3,266	1,885	1,034	402	1	0.57716	0.31660	0.12309

Table 17: DwassI and DwassAltI minimum precision minus desired precision proportion.

in Table 19 show that the predicted function is a very good fit.

The first and second fitted functions in Equations (9) and (10) are multiplied together to get an initial function that predicts $mpdp(DwassI, n, dp, \alpha_{MD})$. The resultant formula denoted by $mpdpInitPred(DwassI, n, dp, \alpha_{MD})$ is shown in Equation (11) below.

Sample size n	$mpdp(DwassI, n, dp, \alpha_{MD} = 0.001)$		
	Actual	Predicted	Predicted minus actual
50	10	10.28	0.28
200	18	17.58	−0.42
800	32	32.18	0.18
2,000	49	49.14	0.14
4,500	72	72.21	0.21
8,000	95	95.29	0.29
20,000	149	148.93	−0.07
45,000	222	221.91	−0.09
80,000	295	294.88	−0.12
200,000	465	464.51	−0.50
450,000	695	695.27	0.27
800,000	926	926.03	0.03
1,500,000	1,267	1,266.92	−0.08
2,500,000	1,635	1,634.72	−0.28
3,500,000	1,934	1,933.68	−0.32
4,500,000	2,192	2,192.19	0.19
5,500,000	2,423	2,423.24	0.24
6,500,000	2,634	2,634.08	0.08
7,500,000	2,829	2,829.23	0.23
8,500,000	3,012	3,011.76	−0.24
9,500,000	3,184	3,183.82	−0.18
10,500,000	3,347	3,347.05	0.05

Predicted: $2.986 + 1.032\sqrt{n}$

Table 18: DwassI prediction of $mpdp(DwassI, n, dp, \alpha_{MD} = 0.001)$.

$$\begin{aligned} \alpha_{MD} &= \exp\left(-2(d^+)^2 n - \frac{2d^+}{3} - \frac{1}{18n}\right) \\ mpdpInitPred(DwassI, n, dp, \alpha_{MD}) &= (2.986 + 1.032\sqrt{n}) \\ &\quad \times (0.6875 - 0.1433\alpha_{MD} - 0.399\alpha_{MD}^{1/4} \\ &\quad\quad +0.04703\alpha_{MD}^3 - 0.1323\alpha_{MD}^4 \\ &\quad\quad -0.0556 \ln \alpha_{MD}) \end{aligned} \tag{11}$$

Probability α_{MD}	$P001mpdp(DwassI, n = 1000000, dp, \alpha_{MD})$ Actual proportion	Predicted proportion	Predicted minus actual
0.001	1.00000	1.00047	0.00047
0.002	0.94879	0.94837	−0.00042
0.003	0.91691	0.91668	−0.00023
0.004	0.89372	0.89358	−0.00014
0.005	0.87536	0.87527	−0.00009
0.006	0.85990	0.86004	0.00014
0.007	0.84734	0.84696	−0.00038
0.008	0.83575	0.83548	−0.00027
0.009	0.82512	0.82522	0.00010
0.01	0.81643	0.81594	−0.00049
0.02	0.75169	0.75209	0.00040
0.03	0.71208	0.71211	0.00003
0.04	0.68213	0.68230	0.00018
0.05	0.65797	0.65823	0.00026
0.06	0.63768	0.63786	0.00018
0.07	0.62029	0.62010	−0.00019
0.08	0.60386	0.60429	0.00042
0.09	0.58937	0.58997	0.00060
0.1	0.57681	0.57685	0.00004
0.11	0.56425	0.56472	0.00047
0.12	0.55362	0.55341	−0.00022
0.13	0.54300	0.54279	−0.00021
0.14	0.53333	0.53277	−0.00057
0.15	0.52367	0.52327	−0.00041
0.175	0.50145	0.50139	−0.00006
0.2	0.48213	0.48166	−0.00046
0.3	0.41643	0.41636	−0.00007
0.4	0.36329	0.36344	0.00015
0.5	0.31594	0.31648	0.00054
0.6	0.27150	0.27177	0.00027
0.7	0.22609	0.22643	0.00034
0.8	0.17874	0.17781	−0.00094
0.9	0.12271	0.12324	0.00054

$$\text{Predicted Proportion} = 0.6875 - 0.1433\alpha_{MD} - 0.399\alpha_{MD}^{1/4} + 0.04703\alpha_{MD}^{3} - 0.1323\alpha_{MD}^{4} - 0.0556 \ln \alpha_{MD}$$

Table 19: DwassI, prediction of $P001mpdp(DwassI, n = 1000000, dp, \alpha_{MD})$ by α_{MD}.

The predictive ability of Equation (11) is tested by applying it to the samples sizes n and p values α_{MD} in Table 20 which are different than the sample sizes n and p values used to construct the approximation in Equation (11). Table 20 shows the initial predicted values $mpdpInitPred(DwassI, n, dp, \alpha_{MD})$ are close to the actual values $mpdp(DwassI, n, dp, \alpha_{MD})$. Let $mpdpIPError(DwassI, n, dp, \alpha_{MD})$ defined in Equation (12) below represent the error in the initial predicted values.

$$\begin{aligned} mpdpIPError(DwassI, n, dp, \alpha_{MD}) &= mpdpInitPred(DwassI, n, dp, \alpha_{MD}) \\ &\quad - mpdp(DwassI, n, dp, \alpha_{MD}) \end{aligned} \tag{12}$$

Table 21 contains the initial prediction error $mpdpIPError(DwassI, n, dp, \alpha_{MD})$ for the sample sizes n and p values α_{MD} in Table 20. Since the largest negative error in Table 21 is -2.44, construct the final prediction by conservatively adding 5 to the initial prediction $mpdpInitPred(DwassI, n, dp, \alpha_{MD})$ and then round up. The resultant final prediction $mpdpFinalPred(DwassI, n, dp, \alpha_{MD})$ is shown in Equation (13) below where $\lceil x \rceil$ is the smallest integer greater than or equal to x.

$$\begin{aligned} \alpha_{MD} &= \exp\left(-2(d^+)^2 n - \frac{2d^+}{3} - \frac{1}{18n}\right) \\ mpdpFinalPred(DwassI, n, dp, \alpha_{MD}) &= \lceil 5 + (2.986 + 1.032\sqrt{n}) \\ &\quad \times (0.6875 - 0.1433\alpha_{MD} - 0.399\alpha_{MD}^{1/4} \\ &\quad + 0.04703\alpha_{MD}^3 - 0.1323\alpha_{MD}^4 \\ &\quad - 0.0556 \ln \alpha_{MD}) \rceil \end{aligned} \tag{13}$$

Let $mpdpFPError(DwassI, n, dp, \alpha_{MD})$ defined in Equation (14) below represent the error in the final predicted values.

$$\begin{aligned} mpdpFPError(DwassI, n, dp, \alpha_{MD}) &= mpdpFinalPred(DwassI, n, dp, \alpha_{MD}) \\ &\quad - mpdp(DwassI, n, dp, \alpha_{MD}) \end{aligned} \tag{14}$$

To determine how good a predictor Equation (13) is, the final predicted error $mpdpFPError(DwassI, n, dp, \alpha_{MD})$ is calculated for the sample sizes n and p values α_{MD} in Table 14. The results in Table 22 show that $mpdpFinalPred(DwassI, n, dp, \alpha_{MD})$ overestimates $mpdp(DwassI, n, dp, \alpha_{MD})$ especially for $\alpha_{MD} = 0.0005$ and $\alpha_{MD} = 0.95$ which is outside of the range of the p values α_{MD} used to fit the functions to construct $mpdpFinalPred(DwassI, n, dp, \alpha_{MD})$ in Equation (13).

The fitted function in Equation (13) is used for both DwassI and DwassAltI to produce the desired precision Mathematica functions `DwassIKS1SidedOneSampleRTdesiredPrecision` and `DwassAltIKS1SidedOneSampleRTdesiredPrecision` listed in Sections 16 and 21 respecitvely of the `KS1SidedOneSampleDwassFormulae.nb` file.

6.2. Computational experience for the Dwass-based formulae

Using the Mathematica function `DwassArbPrecisionRationalTimingsToFile` listed in Section 22 of the `KS1SidedOneSampleDwassFormulae.nb` file, the computer times in Tables 23 and 24 are generated that compare the DwassAltD, DwassD, DwassI, DwassAltI, and the

Sample size n	Actual $mp-dp$ $mpdp(DwassI, n, dp, \alpha_{MD})$ $\alpha_{MD}=0.01$	$\alpha_{MD}=0.3$	$\alpha_{MD}=0.7$	Initial prediction of $mp-dp$ $mpdpInitPred(DwassI, n, dp, \alpha_{MD})$ $\alpha_{MD}=0.01$	$\alpha_{MD}=0.3$	$\alpha_{MD}=0.7$
50	8	4	1	8.39	4.28	2.33
200	14	7	3	14.34	7.32	3.98
800	26	13	6	26.25	13.40	7.29
2,000	40	20	10	40.09	20.46	11.13
4,500	59	29	15	58.92	30.07	16.35
8,000	78	39	21	77.75	39.67	21.58
20,000	121	61	33	121.52	62.01	33.72
45,000	181	92	49	181.06	92.39	50.25
80,000	241	122	66	240.6	122.77	66.77
200,000	379	193	105	379.01	193.40	105.18
450,000	567	290	157	567.3	289.48	157.43
800,000	756	386	209	755.59	385.56	209.68
1,500,000	1,034	528	287	1,033.73	527.49	286.86
2,500,000	1,334	682	370	1,333.83	680.63	370.14
3,500,000	1,578	806	438	1,577.77	805.00	437.83
4,500,000	1,789	914	497	1,788.69	912.73	496.37
5,500,000	1,978	1,011	550	1,977.22	1,008.93	548.68
6,500,000	2,150	1,099	598	2,149.25	1,096.72	596.42
7,500,000	2,310	1,180	642	2,308.48	1,177.97	640.61
8,500,000	2,459	1,256	683	2,457.41	1,253.96	681.94
9,500,000	2,599	1,328	722	2,597.81	1,325.60	720.90
10,500,000	2,732	1,396	760	2,730.99	1,393.56	757.86

$$\alpha_{MD} = \exp\left(-2(d^+)^2 n - \frac{2d^+}{3} - \frac{1}{18n}\right)$$

$$mpdpInitPred(DwassI, n, dp, \alpha_{MD}) = (2.986 + 1.032\sqrt{n}) \times (0.6875 - 0.1433\alpha_{MD} - 0.399\alpha_{MD}^{1/4} + 0.04703\alpha_{MD}^3 - 0.1323\alpha_{MD}^4 - 0.0556 \ln \alpha_{MD})$$

Table 20: DwassI, initial prediction of $mpdp(DwassI, n, dp, \alpha_{MD})$.

Sample size n	Initial prediction error $mpdpIPError(DwassI, n, dp, \alpha_{MD})$ $\alpha_{MD} =$ 0.01	$\alpha_{MD} =$ 0.3	$\alpha_{MD} =$ 0.7
50	0.39	0.28	1.33
200	0.34	0.32	0.98
800	0.25	0.40	1.29
2,000	0.09	0.46	1.13
4,500	−0.08	1.07	1.35
8,000	−0.25	0.67	0.58
20,000	0.52	1.01	0.72
45,000	0.06	0.39	1.25
80,000	−0.40	0.77	0.77
200,000	0.01	0.40	0.18
450,000	0.30	−0.52	0.43
800,000	−0.41	−0.44	0.68
1,500,000	−0.27	−0.51	−0.14
2,500,000	−0.17	−1.37	0.14
3,500,000	−0.23	−0.90	−0.17
4,500,000	−0.31	−1.27	−0.63
5,500,000	−0.78	−2.07	−1.32
6,500,000	−0.75	−2.28	−1.58
7,500,000	−1.52	−2.03	−1.39
8,500,000	−1.59	−2.04	−1.06
9,500,000	−1.19	−2.40	−1.10
10,500,000	−1.01	−2.44	−2.14

$$mpdpIPError(DwassI, n, dp, \alpha_{MD}) = mpdpInitPred(DwassI, n, dp, \alpha_{MD}) - mpdp(DwassI, n, dp, \alpha_{MD})$$

Table 21: DwassI, initial prediction error for $mpdp(DwassI, n, dp, \alpha_{MD})$.

rational DwassAltD formulae. Note that the rational DwassAltD formula is the fastest rational arithmetic implementation of all the Dwass based formulae (Brown and Harvey 2007). As expected, every Dwass-based arbitrary precision formula is much more efficient than the

Sample size n	Final prediction error $mpdpFPError(DwassI, n, dp, \alpha_{MD})$ $\alpha_{MD} =$ 0.0005	0.005	0.05	0.15	0.25	0.6	0.8	0.95
50	5	6	6	6	6	6	6	5
200	5	6	6	6	6	6	7	6
800	5	6	6	6	6	6	6	7
2,000	5	6	6	6	6	7	6	7
4,500	5	6	6	6	6	6	6	7
8,000	6	6	6	6	6	6	6	7
20,000	6	6	6	6	6	6	6	8
45,000	6	6	6	6	6	7	6	8
80,000	5	6	6	6	6	7	6	9
200,000	6	5	6	6	6	7	6	10
450,000	6	5	6	5	5	6	5	12
800,000	6	5	6	5	5	6	5	13
1,500,000	7	5	5	5	5	6	4	15
2,500,000	8	5	6	5	4	6	3	18
3,500,000	9	5	5	4	5	6	3	20
4,500,000	8	4	5	5	4	6	2	22
5,500,000	9	4	6	4	3	6	2	24
6,500,000	9	4	5	4	4	6	2	25
7,500,000	9	5	6	4	3	6	2	26
8,500,000	10	5	5	3	3	6	1	27
9,500,000	10	4	5	3	3	6	1	28
10,500,000	10	4	6	4	2	6	1	29

$$mpdpFPError(DwassI, n, dp, \alpha_{MD}) = mpdpFinalPred(DwassI, n, dp, \alpha_{MD}) - mpdp(DwassI, n, dp, \alpha_{MD})$$

Table 22: DwassI, final prediction error for $mpdp(DwassI, n, dp, \alpha_{MD})$.

rational DwassAltD formula and therefore much more efficient than all rational arithmetic implementations of the Dwass-based formulae.

To get an idea of the relative efficiency of the arbitrary precision implementation of the

Dwass-based formulae, Tables 25 and 26 contain computer times for various sample sizes up to ten million. Unlike the rational arithmetic implementations, DwassD is the fastest arbitrary precision formula.

7. Arbitrary precision implementation of Smirnov formulae

Since the rational arithmetic implementations of the Smirnov-based formulae were faster than the recursion formulae, this section uses the methodology in Section 5 to implement the SmirnovAltD, SmirnovD, SmirnovAltI, and SmirnovI formulae. The five Mathematica functions needed in the methodology to implemement each Smirnov based formula are listed in Table 27 along with the section number in file `KS1SidedOneSampleSmirnovFormulae.nb` where they are found. The remainder of this section uses these Mathematica functions to produce the desired precision function for each Smirnov-based formula.

Preliminary work shows that the minimum precision minus the desired precision for the Smirnov-based formulae is always less than ten for sample sizes up to one million, $n \leq 1,000,000$. Thus the desired precision dp can be used as initial estimate for both the lower and upper bounds in Step 2 of the Section 5 methodology producing the procedure below.

Procedure to determine the Smirnov-based formulae F minimum precision $mp(F, d^+, n, dp)$

Step 1 (initial potential bounds): Set the lower limit l to the desired precision dp, $l = dp$. Set the upper limit and internal precision $u = ip = dp$ and then run the arbitrary precision F method to determine $rp(F, d^+, n, ip)$. If $rp(F, d^+, n, ip) \geq dp$, go to Step 4. Otherwise, set $l = u$ and go to Step 2.

Step 2 (new potential upper bound): At this point, $mp(F, d^+, n, dp) > l$. Set the upper limit u and internal precision ip to the lower limit plus an increment, $u = ip = l + dp - rp(F, d^+, n, l)$.

Step 3 (test potential upper bound): Run the arbitrary precision F method to determine $rp(F, n, nt, ip)$. If $rp(F, d^+, n, ip) \geq dp$, go to Step 4. Otherwise, set $l = u$ and go to Step 2.

Step 4 (test bounds): At this point, $l \leq mp(F, d^+, n, dp) \leq u$. If $l \geq u - 1$, go to Step 6.

Step 5 (binary search): Set $ip = \left\lfloor \frac{l+u}{2} \right\rfloor$ and run the arbitrary precision F method to determine $rp(F, d^+, n, ip)$. If $rp(F, d^+, n, ip) \geq dp$, set $u = ip$ and go to Step 4. Otherwise, set $l = ip$ and go to Step 4.

Step 6 (determine minimum precision): Run the arbitrary precision F method to determine $rp(F, n, nt, l)$. If $rp(F, d^+, n, l) \geq dp$, set $mp(F, d^+, n, dp) = l$ and terminate the procedure. Otherwise, set $mp(F, d^+, n, dp) = u$ and terminate the procedure.

Sample size n	Formula	Time in seconds to calculate $P[D_n^+ \geq d^+]$ for $\alpha_{MD} =$ 0.001	0.01	0.1	0.25	0.5	0.9
10,000	DwassAltD	0.015	0.016	0.015	0.015	0.000	0.000
	DwassD	0.016	0.015	0.016	0.000	0.016	0.000
	DwassI	0.047	0.031	0.016	0.016	0.000	0.015
	DwassAltI	0.047	0.032	0.015	0.016	0.015	0.000
Rational	DwassAltD	17.109	7.328	2.563	7.062	5.329	2.156
30,000	DwassAltD	0.062	0.078	0.078	0.047	0.047	0.031
	DwassD	0.047	0.047	0.047	0.016	0.016	0.000
	DwassI	0.125	0.156	0.141	0.063	0.047	0.047
	DwassAltI	0.156	0.203	0.265	0.078	0.094	0.094
Rational	DwassAltD	320.891	275.75	184.422	6.234	58.000	40.688
50,000	DwassAltD	0.156	0.125	0.093	0.063	0.062	0.063
	DwassD	0.109	0.094	0.032	0.016	0.016	0.000
	DwassI	0.375	0.265	0.156	0.109	0.094	0.078
	DwassAltI	0.500	0.344	0.265	0.187	0.140	0.172
Rational	DwassAltD	982.266	445.188	567.985	441.032	113.005	134.500

Note: All timings on a Pentium IV running at 3.4 GHz.

Note: Desired precision $dp = 20$

Table 23: Time in seconds to calculate $P[D_n^+ \geq d^+]$ using arbitrary precision.

Sample size n	Formula	Time in seconds to calculate $P[D_n^+ \geq d^+]$ for $\alpha_{MD} =$ 0.001	0.01	0.1	0.25	0.5	0.9
10,000	DwassAltD	0.016	0.016	0.015	0.016	0.000	0.016
	DwassD	0.015	0.031	0.016	0.000	0.016	0.000
	DwassI	0.063	0.047	0.031	0.031	0.016	0.000
	DwassAltI	0.078	0.047	0.031	0.031	0.015	0.016
Rational	DwassAltD	17.109	7.328	2.563	7.062	5.329	2.156
30,000	DwassAltD	0.140	0.094	0.078	0.047	0.046	0.031
	DwassD	0.094	0.078	0.047	0.031	0.016	0.015
	DwassI	0.312	0.234	0.140	0.079	0.063	0.047
	DwassAltI	0.391	0.282	0.203	0.109	0.125	0.110
Rational	DwassAltD	320.891	275.75	184.422	6.234	58.000	40.688
50,000	DwassAltD	0.297	0.141	0.093	0.078	0.079	0.047
	DwassD	0.281	0.094	0.063	0.031	0.015	0.016
	DwassI	0.640	0.343	0.203	0.156	0.110	0.078
	DwassAltI	0.594	0.407	0.297	0.250	0.187	0.156
Rational	DwassAltD	982.266	445.188	567.985	441.032	113.500	134.500

Note: All timings on a Pentium IV running at 3.4 GHz.

Note: Desired precision $dp = 100$

Table 24: Time in seconds to calculate $P[D_n^+ \geq d^+]$ using arbitrary precision.

Sample size n	Formula	Time in seconds to calculate $P[D_n^+ \geq d^+]$ for $\alpha_{MD} =$ 0.001	0.01	0.1	0.25	0.5	0.9
100,000	DwassAltD	0.187	0.157	0.109	0.078	0.094	0.079
	DwassD	0.140	0.093	0.047	0.016	0.015	0.000
	DwassI	0.532	0.344	0.187	0.156	0.125	0.109
	DwassAltI	0.671	0.438	0.250	0.250	0.250	0.234
500,000	DwassAltD	1.625	1.828	1.453	1.250	1.031	0.953
	DwassD	1.062	1.157	0.813	0.281	0.141	0.031
	DwassI	3.922	4.031	2.422	1.828	1.406	1.391
	DwassAltI	5.109	6.078	4.484	3.844	3.406	3.344
1,000,000	DwassAltD	5.875	3.907	3.250	2.625	2.328	2.125
	DwassD	4.609	2.750	1.188	0.625	0.266	0.047
	DwassI	18.157	9.593	5.422	4.312	3.125	2.984
	DwassAltI	22.671	15.297	9.875	8.828	7.578	6.797
5,000,000	DwassAltD	49.843	34.937	20.391	18.313	14.500	13.172
	DwassD	36.407	21.141	8.859	4.812	2.015	0.172
	DwassI	100.937	68.297	48.141	32.438	24.719	20.250
	DwassAltI	129.594	90.937	73.578	60.203	52.672	44.765
10,000,000	DwassAltD	111.953	78.734	48.687	40.375	32.172	27.796
	DwassD	85.484	53.156	21.688	11.031	4.781	0.454
	DwassI	241.594	163.250	86.390	82.735	53.125	43.171
	DwassAltI	298.563	216.735	138.797	140.078	110.500	96.157

Note: All timings on a Pentium IV running at 3.4 GHz.

Note: Desired precision $dp = 20$

Table 25: Time in seconds to calculate $P[D_n^+ \geq d^+]$ using arbitrary precision.

Sample size n	Formula	Time in seconds to calculate $P[D_n^+ \geq d^+]$ for $\alpha_{MD} =$ 0.001	0.01	0.1	0.25	0.5	0.9
100,000	DwassAltD	0.235	0.172	0.125	0.093	0.094	0.062
	DwassD	0.187	0.125	0.062	0.047	0.016	0.016
	DwassI	0.688	0.469	0.250	0.172	0.156	0.094
	DwassAltI	0.828	0.594	0.344	0.281	0.297	0.219
500,000	DwassAltD	3.078	2.297	1.531	1.282	1.125	0.969
	DwassD	2.156	1.422	0.625	0.359	0.188	0.047
	DwassI	7.766	5.094	2.844	2.141	1.578	1.406
	DwassAltI	9.312	7.078	4.937	4.078	3.547	3.453
1,000,000	DwassAltD	6.984	5.265	3.406	2.828	2.407	2.156
	DwassD	5.031	3.141	1.406	0.797	0.359	0.063
	DwassI	20.000	12.641	5.391	4.516	3.828	3.015
	DwassAltI	24.313	15.578	10.937	9.093	8.031	6.860
5,000,000	DwassAltD	51.329	36.125	21.781	17.859	14.813	13.125
	DwassD	37.343	20.672	9.719	4.609	2.156	0.297
	DwassI	100.594	68.672	49.375	33.032	24.312	19.906
	DwassAltI	128.906	94.328	75.313	58.562	51.594	44.422
10,000,000	DwassAltD	116.437	82.156	50.171	38.828	31.203	27.594
	DwassD	90.985	55.219	22.860	13.094	5.031	0.656
	DwassI	258.156	167.984	90.344	66.156	53.500	43.578
	DwassAltI	308.765	221.313	143.437	127.797	111.422	93.953

Note: All timings on a Pentium IV running at 3.4 GHz.

Note: Desired precision $dp = 100$

Table 26: Time in seconds to calculate $P[D_n^+ \geq d^+]$ using arbitrary precision.

Function type	Formula	Mathematica function name	Section number
Arbitrary precision formula (Methodology: Step 1 Result)			
	SmirnovAltD	`SmirnovAltDKS1SidedOneSampleRTArbPrecision`	1
	SmirnovD	`SmirnovDKS1SidedOneSampleRTArbPrecision`	7
	SmirnovAltI	`SmirnovAltIKS1SidedOneSampleRTArbPrecision`	13
	SmirnovI	`SmirnovIKS1SidedOneSampleRTArbPrecision`	19
Minimum precision minus desired precision (Methodology: Used in Step 2)			
	SmirnovAltD	`MinPrecisionMinusDesiredPrecisionSmirnovAltD`	2
	SmirnovD	`MinPrecisionMinusDesiredPrecisionSmirnovD`	8
	SmirnovAltI	`MinPrecisionMinusDesiredPrecisionSmirnovAltI`	14
	SmirnovI	`MinPrecisionMinusDesiredPrecisionSmirnovI`	20
Calculation time (Methodology: Used in Step 3)			
	SmirnovAltD	`TimingSmirnovAltDKS1SidedOneSampleRTArbPrecision`	3
	SmirnovD	`TimingSmirnovDKS1SidedOneSampleRTArbPrecision`	9
	SmirnovAltI	`TimingSmirnovAltIKS1SidedOneSampleRTArbPrecision`	15
	SmirnovI	`TimingSmirnovIKS1SidedOneSampleRTArbPrecision`	21
Minimum precision minus desired precision To file (Methodology: Used in Step 5)			
	SmirnovAltD	`MinPrecisionMinusDesiredPrecisionToFileSmirnovAltD`	4
	SmirnovD	`MinPrecisionMinusDesiredPrecisionToFileSmirnovD`	10
	SmirnovAltI	`MinPrecisionMinusDesiredPrecisionToFileSmirnovAltI`	16
	SmirnovI	`MinPrecisionMinusDesiredPrecisionToFileSmirnovI`	22
Minimum precision minus desired precision breakpoints (Methodology: Used in Step 5)			
	SmirnovAltD	`MpMinusDpBreakpointsToFileSmirnovAltD`	5
	SmirnovD	`MpMinusDpBreakpointsToFileSmirnovD`	11
	SmirnovAltI	`MpMinusDpBreakpointsToFileSmirnovAltI`	17
	SmirnovI	`MpMinusDpBreakpointsToFileSmirnovI`	23
Desired precision function (Methodology: Step 6 Result)			
	SmirnovAltD	`SmirnovAltDKS1SidedOneSampleRTdesiredPrecision`	6
	SmirnovD	`SmirnovDKS1SidedOneSampleRTdesiredPrecision`	12
	SmirnovAltI	`SmirnovAltIKS1SidedOneSampleRTdesiredPrecision`	18
	SmirnovI	`SmirnovIKS1SidedOneSampleRTdesiredPrecision`	24
Functions listed in file `KS1SidedOneSampleSmirnovFormulae.nb`			

Table 27: Mathematica function names for Smirnov-based formulae.

Sample size n	Formula	$\alpha_{MD} = 0.001$ $dp = 20$ Minimum precision	Time in seconds	$\alpha_{MD} = 0.001$ $dp = 100$ Minimum precision	Time in seconds	$\alpha_{MD} = 0.9$ $dp = 20$ Minimum precision	Time in seconds
100,000	SmirnovAltD	26	5.797	106	8.922	26	6.110
	SmirnovD	26	6.844	106	15.422	26	11.797
	SmirnovAltI	26	14.063	106	29.266	26	16.406
	SmirnovI	26	15.875	106	28.110	26	15.954
500,000	SmirnovAltD	26	36.250	106	66.75	26	49.281
	SmirnovD	26	61.406	106	91.422	26	63.156
	SmirnovAltI	27	83.703	107	146.125	26	83.64
	SmirnovI	27	84.984	107	149.640	26	83.859
1,000,000	SmirnovAltD	27	109.891	107	166.813	27	113.61
	SmirnovD	27	128.109	107	190.640	27	127.953
	SmirnovAltI	27	171.875	107	298.297	27	170.61
	SmirnovI	27	170.500	107	300.360	27	167.937

Minimum Precision is $mp(F, n, dp, \alpha_{MD})$

Maag and Dicaire test statistic generated with $\rho = 6$ digits of precision for α_{MD}

Computational times on a 3.4 GHz Pentium IV

Table 28: Time in seconds for a Smirnov-based formula F to compute a p value.

Using the minimum precisions $mp(F, d^+, n, dp)$ for the Smirnov-based formulae, various computer times are contained in Table 28. Since all the Smirnov-based formulae for a sample size of one million have computer times exceeding 100 seconds, the following development uses a sample size of one million ($n = 1,000,000$) as the upper limit.

7.1. Predicting internal precision for the Smirnov-based formulae

For the Smirnov-based formulae F (SmirnovAltD, SmirnovD, SmirnovAltI, SmirnovI), Table 29 contains the computed the minimum precision minus desired precisions $mpdp(F, n, dp, \alpha_{MD})$. From the data in this table, the minimum precision minus desired precision $mpdp(F, n, dp, \alpha_{MD})$ is almost always the same for the same sample size n independent of the desired precision dp and the p value α_{MD}. Thus, the function predicting the internal precision just depends on the sample size n. Using binary search and $\alpha_{MD} = 0.001$, Table 30 contains the lower and upper bounds on the sample size n for each $mp - dp$. Conservatively adding one to the $mp - dp$ in Table 30, Table 31 contains the predicted internal precisions used in the Smirnov-based desired precision functions listed in Sections 6 (SmirnovAltD), 12 (SmirnovD), 18 (SmirnivAltI),

and 24 (SmirnovI) of the file `KS1SidedOneSampleSmirnovFormulae.nb` (see Table 27).

7.2. Computational experience for the Smirnov-based formulae

Using the Mathematica function `SmirnovArbPrecisionDwassDTimingsToFile` listed in Section 25 of the `KS1SidedOneSampleSmirnovFormulae.nb` file, the computer times in Tables 32 ($dp = 20$) and 33 ($dp = 100$) are generated to compare the SmirnovAltD, SmirnovD, SmirnovI, SmirnovAltI, and DwassD formulae. The DwassD formula was chosen because it is the fastest arbitrary precision implementation of all the Dwass based formulae. As expected, every Smirnov-based arbitrary precision formula is less efficient than the DwassD formula.

8. Arbitrary precision implementation of recursion formulae

From Table 5 in Section 3, the number of terms in both the Dwass-based formulae and the Smirnov-based formulae are much less than the number of terms in the recursion formulae (Daniels, Noe, Steck, Conover, Bolshev). This data on the number of terms suggests that the recursion formulae will be significantly slower than both the Dwass-based formulae and the Smirnov-based formulae. Consequently, this section will not use the methodology in Section 5 to implement the recursion formulae but instead will compare the computer time needed for their arbitrary precision implementations to the time needed for the Dwass and Smirnov-based formulae. In the unlikely case that some recursion formulae are not significantly slower than the Dwass and Smirnov-based formulae, then those recursion formulae will be implemented using the methodology in Section 5. The Mathematica functions needed to implemement and time the recursion formulae are listed in Table 34 along with the section number where they are found in file `KS1SidedOneSampleRecursionFormulae.nb`. For sample size n, preliminary analysis in finding the minimum precision minus the desired precision $mp - dp$ for each recursion formula yielded an initial estimate of $0.384n + 0.004n^2$ for the Daniels, Steck, Conover, and Bolshev recursion formulae. The initial estimate for the Noe recursion formula is 10. To find $mp - dp$ for each recursion formula, these initial estimates are used in the Mathematica functions contained Sections $2, 6, 10, 14, 18$ of file `KS1SidedOneSampleRecursionFormulae.nb`.

The number of terms in four of the recursion formulae (Daniels, Steck, Conover, Bolshev) are very similar so they will be analyzed first. Since the number of terms for the Noe recursion formula is over 50 times the number of terms in the other recursion formulae, the Noe recursion formula will probably be much slower than the other recusion formulae and it will be analyzed separately.

For the Daniels, Steck, Conover, and Bolshev recursion formulae, the arbitrary precision Mathematica functions contained in file `KS1SidedOneSampleRecursionFormulae.nb` (Sections $3, 4, 7, 8, 11, 12, 15, 16$) are used to determine the minimum precision minus desired precisions $mp - dp$ in Table 35 and the timings in Tables 36 and 37. In addition, these tables include the $mp - dp$ and timings for the fastest Smirnov-based formula (SmirnovAltD) and the fastest Dwass-based formula (DwassD). Since Daniels is the only recursion formula with negative terms, we would expect the $mp - dp$ in Table 35 for Daniels to be be larger than the other recursion formulae which is the case. Interestingly, Steck has significantly smaller $mp - dp$ than the other recursion formulae even though the number of terms is almost the same. From Tables 36 and 37, Steck is the fastest recursion formula but as expected is significantly slower than SmirnovAltD which in turn is slower than DwassD.

Sample size n	$mpdp(F, n, dp, \alpha_{MD})$ SmirnovD and SmirnovAltD $\alpha_{MD} =$ 0.001	0.1	0.5	0.9	SmirnovI and SmirnovAltI $\alpha_{MD} =$ 0.001	0.1	0.5	0.9
100	3	3	3	3	3	3	3	3
200	3	3	3	3	3	3	3	3
400	3	3	3	3	4	4	4	3
600	3	3	3	3	4	4	4	4
800	4	4	4	3	4	4	4	4
1,000	4	4	4	4	4	4	4	4
2,000	4	4	4	4	4	4	4	4
4,000	4	4	4	4	5	5	5	4
6,000	4	4	4	4	5	5	5	5
8,000	5	5	5	4	5	5	5	5
10,000	5	5	5	5	5	5	5	5
20,000	5	5	5	5	5	5	5	5
40,000	5	5	5	5	6	6	6	5
60,000	5	5	5	5	6	6	6	6
80,000	6	6	6	5	6	6	6	6
100,000	6	6	6	6	6	6	6	6
200,000	6	6	6	6	6	6	6	6
400,000	6	6	6	6	7	7	7	6
600,000	6	6	6	6	7	7	7	7
800,000	7	7	7	6	7	7	7	7
1,000,000	7	7	7	7	7	7	7	7

Table 29: Minimum precision minus desired precision for Smirnov formulae.

	Sample size n lower and upper limits for $mp - dp$ and $\alpha_{MD} = 0.001$							
	SmirnovAltD		SmirnovD		SmirnovAltI		SmirnovI	
	Low	High	Low	High	Low	High	Low	High
$mp - dp$	n	n	n	n	n	n	n	n
2	20	66	20	66	20	40	20	40
3	67	671	67	671	41	308	41	308
4	672	6,773	672	6,773	309	2,918	309	2,918
5	6,774	67,939	6,774	67,939	2,919	28,389	2,919	28,389
6	67,940	680,078	67,940	680,077	28,390	280,680	28,390	280,680
7	680,079		680,078		280,681		280,681	

$mp - dp$ is minimum precision minus desired precision

Table 30: Smirnov-based formulae, $mp - dp$ breakpoints for sample sizes $n = 20$ to $n = 1,000,000$.

	Sample size n lower and upper limits for $mp - dp$							
Internal	SmirnovAltD		SmirnovD		SmirnovAltI		SmirnovI	
precision	Low	High	Low	High	Low	High	Low	High
ip	n	n	n	n	n	n	n	n
$dp + 3$	20	66	20	66	20	40	20	40
$dp + 4$	67	671	67	671	41	308	41	308
$dp + 5$	672	6,773	672	6,773	309	2,918	309	2,918
$dp + 6$	6,774	67,939	6,774	67,939	2,919	28,389	2,919	28,389
$dp + 7$	67,940	680,078	67,940	680,077	28,390	280,680	28,390	280,680
$dp + 8$	680,079	999,999	680,078	999,999	280,681	999,999	280,681	999,999
$dp + 9$	$n \geq 1,000,000$		$n \geq 1,000,000$		$n \geq 1,000,000$		$n \geq 1,000,000$	

$mp - dp$ is minimum precision minus desired precision

Table 31: Smirnov-based formulae, internal precision ip.

Sample size n	Formula	Time in seconds to calculate $P[D_n^+ \geq d^+]$ for $\alpha_{MD} =$ 0.001	0.01	0.1	0.25	0.5	0.9
10,000	SmirnovAltD	0.515	0.515	0.516	0.515	0.531	0.531
	SmirnovD	0.578	0.594	0.609	0.625	0.609	0.609
	SmirnovI	0.922	0.906	0.906	0.922	0.907	0.922
	SmirnovAltI	0.922	0.922	0.922	0.922	0.906	0.922
	DwassD	0.016	0.016	0.016	0.016	0.000	0.016
50,000	SmirnovAltD	2.922	2.875	2.875	2.890	2.906	2.890
	SmirnovD	3.281	3.266	3.281	3.313	3.297	3.328
	SmirnovI	4.703	4.734	4.750	5.016	4.735	4.688
	SmirnovAltI	4.766	4.750	4.750	4.796	4.75	4.734
	DwassD	0.062	0.047	0.016	0.016	0.000	0.000
1000,000	SmirnovAltD	5.985	8.625	8.734	8.125	6.203	6.015
	SmirnovD	6.796	6.843	8.359	6.860	6.891	10.297
	SmirnovI	9.610	9.672	9.578	10.093	13.937	10.407
	SmirnovAltI	11.172	9.688	9.922	12.985	9.750	9.703
	DwassD	0.250	0.094	0.047	0.031	0.016	0.015
500,000	SmirnovAltD	47.094	45.562	46.625	43.078	47.016	47.266
	SmirnovD	50.375	43.438	45.171	49.266	48.515	49.750
	SmirnovI	58.312	63.781	64.985	60.016	63.078	65.734
	SmirnovAltI	61.781	62.219	68.297	68.250	65.610	71.578
	DwassD	1.891	1.141	0.390	0.281	0.125	0.016
1,000,000	SmirnovAltD	106.219	97.109	100.75	101.547	98.375	109.297
	SmirnovD	109.984	117.266	116.812	117.063	114.360	116.953
	SmirnovI	140.250	148.297	147.094	144.515	148.093	146.406
	SmirnovAltI	153.000	147.281	146.015	147.391	141.844	154.094
	DwassD	4.688	1.594	1.172	0.359	0.281	0.047

Note: All timings on a Pentium IV running at 3.4 GHz.

Note: Desired precision $dp = 20$

Table 32: Time in seconds to calculate $P[D_n^+ \geq d^+]$ using arbitrary precision.

For the Noe recursion formula, the arbitrary precision Mathematica functions contained in file `KS1SidedOneSampleRecursionFormulae.nb` (Sections 19 and 20) are used to determine the minimum precision minus desired precisions $mp - dp$ in Table 38 and the timings in Table 39. As expected, comparing the computer times in Tables 36 and 37 with those in Table 39 shows that the Noe recursion formula is significantly slower than the other recursion formulae.

From the timings in Sections 6.2, 7.2, and 8, DwassD is the fastest formula and will be used

Sample size n	Formula	Time in seconds to calculate $P[D_n^+ \geq d^+]$ for $\alpha_{MD} =$ 0.001	0.01	0.1	0.25	0.5	0.9
10,000	SmirnovAltD	0.734	0.765	0.750	0.766	0.750	0.750
	SmirnovD	0.860	0.828	0.875	0.843	0.844	0.875
	SmirnovI	1.671	1.688	1.672	1.688	1.703	1.672
	SmirnovAltI	1.688	1.687	1.688	1.687	1.703	1.672
	DwassD	0.016	0.016	0.000	0.016	0.000	0.000
50,000	SmirnovAltD	4.235	4.250	4.234	4.282	4.281	4.234
	SmirnovD	4.750	4.750	4.781	4.750	4.766	4.813
	SmirnovI	8.578	8.547	8.563	8.578	8.547	8.547
	SmirnovAltI	8.593	8.578	8.609	8.547	8.562	8.500
	DwassD	0.094	0.063	0.031	0.031	0.016	0.015
100,000	SmirnovAltD	12.750	13.343	14.375	12.343	10.047	12.984
	SmirnovD	11.594	10.735	9.953	10.000	15.906	12.406
	SmirnovI	19.438	21.750	22.297	20.344	17.688	21.594
	SmirnovAltI	19.875	17.281	19.937	21.297	21.594	21.516
	DwassD	0.203	0.125	0.110	0.031	0.031	0.000
500,000	SmirnovAltD	69.313	68.64	78.235	73.687	75.360	76.219
	SmirnovD	80.218	82.500	86.234	86.656	85.172	77.171
	SmirnovI	122.235	125.375	123.219	129.563	127.718	123.922
	SmirnovAltI	125.609	124.453	129.172	118.953	131.719	126.688
	DwassD	2.141	1.406	0.375	0.359	0.266	0.031
1,000,000	SmirnovAltD	159.734	167.953	165.328	167.656	171.937	171.781
	SmirnovD	177.047	180.047	180.938	186.125	185.109	189.187
	SmirnovI	273.906	276.562	274.734	285.860	288.829	290.407
	SmirnovAltI	264.360	272.781	287.047	286.968	281.328	289.359
	DwassD	4.203	1.844	1.078	0.704	0.375	0.063

Note: All timings on a Pentium IV running at 3.4 GHz.

Note: Desired precision $dp = 100$

Table 33: Time in seconds to calculate $P[D_n^+ \geq d^+]$ using arbitrary precision.

in the next section to calculate bandwidths.

Function type	Formula name	Mathematica function name	Section number
Arbitrary precision formula			
	Daniels	`DanielsKS1SidedOneSampleRTArbPrecision`	1
	Steck	`SteckKS1SidedOneSampleRTArbPrecision`	5
	Conover	`ConoverKS1SidedOneSampleRTArbPrecision`	9
	Bolshev	`BolshevKS1SidedOneSampleRTArbPrecision`	13
	Noe	`NoeKS1SidedOneSampleRTArbPrecision`	17
Minimum precision minus desired precision			
	Daniels	`MinPrecisionMinusDesiredPrecisionDaniels`	2
	Steck	`MinPrecisionMinusDesiredPrecisionSteck`	6
	Conover	`MinPrecisionMinusDesiredPrecisionConover`	10
	Bolshev	`MinPrecisionMinusDesiredPrecisionBolshev`	14
	Noe	`MinPrecisionMinusDesiredPrecisionNoe`	18
Calculation time			
	Daniels	`TimingDanielsKS1SidedOneSampleRTArbPrecision`	3
	Steck	`TimingSteckKS1SidedOneSampleRTArbPrecision`	7
	Conover	`TimingConoverKS1SidedOneSampleRTArbPrecision`	11
	Bolshev	`TimingBolshevKS1SidedOneSampleRTArbPrecision`	15
	Noe	`TimingNoeKS1SidedOneSampleRTArbPrecision`	19
Timing and minimum precision minus desired precision			
	Daniels	`TimingMinPrecisionMinusDesiredPrecisionDaniels`	4
	Steck	`TimingMinPrecisionMinusDesiredPrecisionSteck`	8
	Conover	`TimingMinPrecisionMinusDesiredPrecisionConover`	12
	Bolshev	`TimingMinPrecisionMinusDesiredPrecisionBolshev`	16
	Noe	`TimingMinPrecisionMinusDesiredPrecisionNoe`	20

Functions listed in file `KS1SidedOneSampleRecursionFormulae.nb`

Table 34: Mathematica function names for recursion formulae.

Sample size n	Formula	$mp - dp$ to calculate $P\left[D_n^+ \geq d^+\right]$ for $\alpha_{MD} =$ 0.001	0.01	0.1	0.25	0.5	0.9
100	Daniels	155	151	146	143	140	134
	Steck	6	5	4	4	4	4
	Conover	88	96	107	113	119	128
	Bolshev	88	97	107	113	119	128
	SmirnovAltD	3	3	3	3	3	3
	DwassD	14	11	8	6	5	2
200	Daniels	357	352	344	340	335	326
	Steck	6	5	4	4	5	5
	Conover	249	262	280	290	300	314
	Bolshev	249	263	280	290	300	315
	SmirnovAltD	3	3	3	3	3	3
	DwassD	19	15	11	8	6	3
300	Daniels	580	573	563	557	551	540
	Steck	6	5	5	5	6	7
	Conover	437	455	479	492	505	524
	Bolshev	438	455	479	492	505	524
	SmirnovAltD	3	3	3	3	3	3
	DwassD	22	18	13	10	7	3
400	Daniels	816	808	796	789	781	769
	Steck	6	5	6	6	7	8
	Conover	643	665	694	709	725	748
	Bolshev	644	665	694	710	725	748
	SmirnovAltD	3	3	3	3	3	3
	DwassD	25	20	14	11	8	4
500	Daniels	1,062	1,053	1,039	1,031	1,023	1,008
	Steck	6	6	7	8	8	9
	Conover	862	888	921	939	957	984
	Bolshev	862	888	921	939	957	984
	SmirnovAltD	3	3	3	3	3	3
	DwassD	27	22	16	12	9	4

Number of test statistic digits $\rho = 6$

Table 35: Minimum precision minus desired precision, $mp - dp$, to calculate $P\left[D_n^+ \geq d^+\right]$.

Sample size n	Formula	Time in seconds to calculate $P[D_n^+ \geq d^+]$ for $\alpha_{MD} =$ 0.001	0.01	0.1	0.25	0.5	0.9
100	Daniels	0.250	0.250	0.250	0.234	0.375	0.250
	Steck	0.172	0.157	0.156	0.156	0.157	0.172
	Conover	0.125	0.141	0.172	0.188	0.203	0.235
	Bolshev	0.172	0.157	0.172	0.172	0.187	0.203
	SmirnovAltD	0.000	0.015	0.016	0.016	0.015	0.015
	DwassD	0.000	0.000	0.000	0.000	0.000	0.000
200	Daniels	1.750	1.734	1.781	2.969	3.000	2.938
	Steck	0.657	0.657	0.640	0.640	0.640	0.640
	Conover	1.625	1.766	1.984	2.156	2.406	2.453
	Bolshev	1.125	1.172	2.188	2.172	2.266	2.391
	SmirnovAltD	0.016	0.016	0.016	0.016	0.000	0.000
	DwassD	0.000	0.000	0.000	0.000	0.000	0.000
300	Daniels	11.125	11.984	11.797	12.375	11.953	11.250
	Steck	2.578	2.579	2.593	2.594	2.562	2.609
	Conover	6.437	7.015	8.000	8.078	8.781	10.157
	Bolshev	7.766	8.157	8.687	8.984	9.265	9.609
	SmirnovAltD	0.016	0.031	0.016	0.015	0.015	0.015
	DwassD	0.000	0.000	0.000	0.000	0.000	0.000
400	Daniels	33.140	27.953	29.515	29.734	30.109	29.421
	Steck	4.609	4.860	4.672	4.672	4.703	4.641
	Conover	19.875	21.765	25.782	28.875	29.547	32.453
	Bolshev	21.812	21.125	23.766	25.250	26.219	27.094
	SmirnovAltD	0.015	0.016	0.016	0.016	0.015	0.031
	DwassD	0.000	0.000	0.000	0.000	0.000	0.000
500	Daniels	73.859	74.329	72.672	72.953	72.234	71.078
	Steck	7.282	7.297	7.312	7.359	7.344	7.406
	Conover	53.516	60.343	69.500	75.016	80.125	87.438
	Bolshev	53.016	53.906	59.468	61.594	61.656	65.703
	SmirnovAltD	0.031	0.031	0.031	0.016	0.016	0.032
	DwassD	0.000	0.000	0.000	0.000	0.000	0.000

Note: All timings on a Pentium IV running at 3.4 GHz.

Note: Desired precision $dp = 20$

Note: Number of test statistic digits $\rho = 6$

Table 36: Time in seconds to calculate $P[D_n^+ \geq d^+]$ using arbitrary precision.

Sample size n	Formula	Time in seconds to calculate $P[D_n^+ \geq d^+]$ for $\alpha_{MD} =$ 0.001	0.01	0.1	0.25	0.5	0.9
100	Daniels	0.297	0.297	0.296	0.297	0.282	0.281
	Steck	0.312	0.172	0.187	0.188	0.187	0.188
	Conover	0.156	0.172	0.203	0.219	0.234	0.250
	Bolshev	0.204	0.218	0.235	0.218	0.234	0.235
	SmirnovAltD	0.015	0.016	0.016	0.000	0.000	0.015
	DwassD	0.000	0.000	0.000	0.000	0.000	0.000
200	Daniels	2.094	2.109	2.047	2.078	2.047	2.031
	Steck	0.766	0.750	0.735	0.750	0.750	0.765
	Conover	1.156	1.266	1.391	1.500	1.578	1.750
	Bolshev	1.437	1.485	1.578	1.625	1.656	1.703
	SmirnovAltD	0.016	0.015	0.016	0.000	0.015	0.016
	DwassD	0.000	0.000	0.000	0.000	0.000	0.000
300	Daniels	8.000	8.063	7.921	7.813	7.875	8.031
	Steck	1.750	1.750	1.750	1.750	1.782	1.781
	Conover	4.578	7.797	7.828	9.266	10.406	11.547
	Bolshev	5.594	5.734	6.000	6.141	6.609	11.438
	SmirnovAltD	0.016	0.015	0.016	0.015	0.016	0.016
	DwassD	0.000	0.000	0.000	0.000	0.000	0.000
400	Daniels	32.485	37.109	37.125	35.984	36.438	37.359
	Steck	3.219	3.187	3.110	3.218	3.235	3.265
	Conover	22.594	24.172	27.078	28.062	29.219	31.782
	Bolshev	23.969	27.562	29.141	30.657	30.890	32.547
	SmirnovAltD	0.015	0.016	0.031	0.015	0.016	0.031
	DwassD	0.000	0.000	0.000	0.000	0.000	0.000
500	Daniels	83.969	82.875	82.750	81.593	82.766	79.187
	Steck	5.110	5.156	5.172	5.172	5.172	5.187
	Conover	53.375	56.906	68.406	75.438	81.250	90.562
	Bolshev	62.469	64.797	69.687	68.813	70.890	74.594
	SmirnovAltD	0.032	0.031	0.031	0.031	0.016	0.031
	DwassD	0.016	0.000	0.000	0.000	0.000	0.000

Note: All timings on a Pentium IV running at 3.4 GHz.

Note: Desired precision $dp = 100$

Note: Number of test statistic digits $\rho = 6$

Table 37: Time in seconds to calculate $P[D_n^+ \geq d^+]$ using arbitrary precision.

Sample size n	Noe $mp-dp$ to calculate $P[D_n^+ \geq d^+]$ for $\alpha_{MD} =$ 0.001	0.01	0.1	0.25	0.5	0.9
25	6	5	4	4	3	2
50	7	6	5	4	4	3
100	7	6	5	5	4	4
150	8	7	6	5	5	4
200	8	7	6	6	5	4

Number of test statistic digits $\rho = 6$

Table 38: Noe minimum precision minus desired precision, $mp-dp$, to calculate $P[D_n^+ \geq d^+]$.

Desired precision dp	Sample size n	Noe time in seconds to calculate $P[D_n^+ \geq d^+]$ for $\alpha_{MD} =$ 0.001	0.01	0.1	0.25	0.5	0.9
20	25	0.110	0.125	0.110	0.110	0.110	0.125
	50	0.860	0.828	0.859	0.875	0.860	0.875
	100	12.062	12.047	10.641	12.031	11.188	12.079
	150	39.562	41.328	40.843	40.406	41.015	42.000
	200	99.125	100.109	98.937	100.672	99.860	101.094
100	25	0.125	0.125	0.125	0.125	0.125	0.125
	50	0.953	0.968	0.938	0.937	0.938	0.937
	100	7.829	7.718	7.813	7.797	7.781	7.719
	150	40.265	47.719	45.734	45.984	46.313	46.953
	200	113.828	115.532	112.234	113.656	114.625	111.953

Note: All timings on a Pentium IV running at 3.4 GHz.

Note: Number of test statistic digits $\rho = 6$

Table 39: Noe time in seconds to calculate $P[D_n^+ \geq d^+]$ using arbitrary precision.

9. Calculating the one-sided bandwidth

In addition to calculating the p value for hypothesis testing, the one-sided one sample K-S cumulative sampling distribution can be used to construct a one-sided confidence band around the empirical distribution $F_n(x)$. The bandwidth of a one-sided confidence band with confidence coefficient $1-\alpha$ and sample size n is the value of the test statistic d^+ that satisfies $P(D_n \geq d^+) = \alpha$. Determining a bandwidth d^+ for a particular sample size n and confidence coefficient $1-\alpha$ means evaluating the inverse of the cumulative sampling distribution which can only be done by search techniques such as binary search. Unlike the p value, a bandwidth d^+ cannot in practice be determined exactly because the search technique may not converge to the exact value. For example, binary search with starting values of 0 and 1 would never find $d^+ = 1/3$ and would iterate forever. Thus, search techniques are designed to stop when a specified accuracy is reached. Let $d^+(n, \alpha, \rho)$ represent the bandwidth rounded to ρ significant digits for sample size n and confidence coefficient $1-\alpha$. Note that bandwidth $d^+(n, \alpha, \rho)$ is also the hypothesis testing critical value for an α level of significance.

The linear search algorithm in Brown and Harvey (2007) to determine $d^+(n, \alpha, \rho)$ using rational arithmetic can be modified to use arbitrary precision by replacing the rational arithmetic version of DwassAltD by the desired precision DwassD function contained in Section 11 of the `KS1SidedOneSampleDwassFormulae.nb` file. The resulting Mathematica function `KS1SidedOneSampleBandwidthArbPrecision` and sample output is contained in Section 24 of the `KS1SidedOneSampleDwassFormulae.nb` file.

The Mathematica function `KS1SidedOneSampleArbPrecisionBandwidthsToFile` contained in Section 25 of the `KS1SidedOneSampleDwassFormulae.nb` file finds bandwidths using the Section 24 function and writes these bandwidths to a comma delimited file for input into Excel and a text file that can be used as the input into timing programs. The text file contains bandwidths where every digit in a half-width is output separately so the bandwidth can be reconstructed to any desired accuracy. Using the results of this function, Table 40 contains the bandwidths to six digits of precision ($\rho = 6$) for $\alpha = 0.2, 0.1, 0.05, 0.02, 0.01, 0.001$ and representative sample sizes from $n = 3,000$ through $n = 10,000,000$ (Brown and Harvey 2007, contain bandwidths up to $n = 2,000$).

10. Conclusion and areas of future research

This paper has developed an arbitrary precision method that can be used to compute one-sided one sample K-S p values for sample sizes of at least ten million, $n = 10,000,000$. Most importantly, the method lets the user specify the precision of the resulting p value before the computations are made. In addition, the arbitrary precision method is much faster than the fastest rational arithmetic method. Consequently, it can be used to study the errors in the limiting distribution, approximations, and machine precision implementations.

Acknowledgments

We gratefully acknowledge the financial support the Division of Research and Graduate Studies at Kent State University.

Sample size n	Bandwidth $d^+(n, \alpha, \rho = 6)$					
	$\alpha = 0.2$	$\alpha = 0.1$	$\alpha = 0.05$	$\alpha = 0.02$	$\alpha = 0.01$	$\alpha = 0.001$
3,000	.0163227	.0195343	.0222889	.0254780	.0276475	.0338721
4,000	.0141423	.0169236	.0193092	.0220712	.0239501	.0293412
5,000	.0126531	.0151409	.0172747	.0197451	.0214257	.0262479
6,000	.0115533	.0138244	.0157722	.0180274	.0195617	.0239638
7,000	.0106982	.0128007	.0146042	.0166921	.0181126	.0221882
8,000	.0100087	.0119755	.0136624	.0156155	.0169442	.0207567
9,000	.00943738	.0112917	.0128822	.0147236	.0159763	.0195708
10,000	.00895398	.0107131	.0122220	.0139689	.0151574	.0185674
20,000	.00633486	.0075788	.00864574	.00988104	.0107214	.0131328
30,000	.00517364	.00618931	.00706047	.00806909	.00875527	.0107242
40,000	.00448114	.00536074	.00611519	.00698869	.00758294	.00928808
50,000	.00400845	.00479519	.00546999	.00625127	.00678279	.00830791
60,000	.00365946	.00437765	.00499366	.00570687	.00619208	.00758432
70,000	.00338819	.00405311	.00462342	.00528373	.00573294	.00702191
80,000	.00316951	.00379148	.00432496	.00494262	.00536282	.00656855
90,000	.00298835	.00357475	.00407772	.00466006	.00505623	.00619300
100,000	.00283509	.00339140	.00386856	.00442101	.00479685	.00587529
200,000	.00200506	.00239842	.00273583	.00312647	.00339223	.00415481
300,000	.00163724	.00195843	.00223392	.00255288	.00276987	.00339251
400,000	.00141796	.00169611	.00193470	.00221092	.00239884	.00293807
500,000	.00126830	.00151709	.00173048	.00197755	.00214563	.00262792
600,000	.00115782	.00138493	.00157974	.00180527	.00195871	.00239898
700,000	.00107196	.00128222	.00146257	.00167138	.00181343	.00222104
800,000	.00100273	.00119942	.00136812	.00156345	.00169632	.00207761
900,000	.000945400	.00113084	.00128989	.00147404	.00159932	.00195880
1,000,000	.000896895	.00107282	.00122371	.00139840	.00151726	.00185829
2,000,000	.000634235	.000758630	.000865325	.000988858	.00107289	.00131404
3,000,000	.000517863	.000619431	.000706548	.000807412	.000876031	.00107293
4,000,000	.000448488	.000536449	.000611895	.000699245	.000758672	.000929189
5,000,000	.000401145	.000479819	.000547299	.000625428	.000678580	.000831095
6,000,000	.000366195	.000438016	.000499616	.000570938	.000619459	.000758685
7,000,000	.000339033	.000405525	.000462556	.000528587	.000573510	.000702408
8,000,000	.000317138	.000379336	.000432683	.000494450	.000536471	.000657044
9,000,000	.000299002	.000357642	.000407939	.000466173	.000505791	.000619468
10,000,000	.000283659	.000339290	.000387006	.000442251	.000479836	.000587680

Table 40: Bandwidth $d^+(n, \alpha, \rho = 6)$ to six digits of precision for $n = 3,000$ to $n = 10,000,000$.

References

Birnbaum ZW, Tingey FH (1951). "One-sided Confidence Contours for Probability Functions." *Annals of Mathematical Statistics*, **22**(4), 592–596.

Brown JR, Harvey ME (2007). "Rational Arithmetic Mathematica Functions to Evaluate the One-Sided One Sample K–S Cumulative Sampling Distribution." *Journal of Statistical Software*, **19**(6), 1–32. URL http://www.jstatsoft.org/v19/i06/.

Conover WJ (1972). "A Kolmogorov Goodness-of-Fit Test for Discontinuous Distributions." *Journal of the American Statistical Association*, **67**(339), 591–596.

Daniels HE (1945). "The Statistical Theory of the Strength of Bundles of Threads, I." *Proceedings of the Royal Statistical Society A*, **183**, 405–435.

Dwass M (1959). "The Distribution of the Generalized D_N^+." *Annals of Mathematical Statistics*, **29**(4), 1024–1028.

Feller W (1948). "On the Kolmogorov-Smirnov Limit Theorems for Empirical Distributions." *Annals of Mathematical Statistics*, **19**(2), 177–189.

Kotelnikov VF, Chmaladze EV (1983). "On Computing the Probability of an Empirical Process Not Crossing a Curvilinear Boundary." *Theory of Probability and Its Application*, **27**(3), 640–648.

Maag UR, Dicaire G (1971). "On Kolmogorov-Smirnov Type One-sample Statistics." *Biometrika*, **58**(3), 653–656.

Noe M (1972). "The Calculation of Distributions of Two-Sided Kolmogorov-Smirnov Type Statistics." *Annals of Mathematical Statistics*, **43**(1), 58–64.

Noe M, Vandewiele G (1968). "The Calculation of Distributions of Kolmogorov-Smirmov Type Statistics Including a Table of Significance Points for a Particular Case." *Annals of Mathematical Statistics*, **39**(1), 233–241.

Shorack GR, Wellner JA (1986). *Empirical Processes with Applications to Statistics*. John Wiley & Sons, New York.

Smirnov NV (1944). "Approximate Laws of Distribution of Random Variables from Empirical Data." *Uspekhi Matematicheskikh Nauk*, **10**, 179–206.

Steck GP (1969). "The Smirnov Two Sample Tests as Rank Tests." *Annals of Mathematical Statistics*, **40**(4), 1449–1466.

Wolfram S (2003). *The Mathematica Book*. Fifth edition. Wolfram Media.

Affiliation:

J. Randall Brown
Department of Management & Information Systems
Graduate School of Management
Kent State University
Kent, OH 44240, United States of America
E-mail: jbrown3@kent.edu

Milton E. Harvey
Department of Geography
Kent State University
Kent, OH 44240, United States of America
E-mail: mharvey@kent.edu

Journal of Statistical Software http://www.jstatsoft.org/
published by the American Statistical Association http://www.amstat.org/
Volume 26, Issue 3 *Submitted:* 2007-09-03
June 2008 *Accepted:* 2008-04-03