Journal of Statistical Software

June 2008, Volume 26, Issue 2. *http://www.jstatsoft.org/*

Rational Arithmetic **Mathematica** Functions to Evaluate the Two-Sided One Sample K-S Cumulative Sampling Distribution

J. Randall Brown
Kent State University

Milton E. Harvey
Kent State University

Abstract

One of the most widely used goodness-of-fit tests is the two-sided one sample Kolmogorov-Smirnov (K-S) test which has been implemented by many computer statistical software packages. To calculate a two-sided p value (evaluate the cumulative sampling distribution), these packages use various methods including recursion formulae, limiting distributions, and approximations of unknown accuracy developed over thirty years ago. Based on an extensive literature search for the two-sided one sample K-S test, this paper identifies an exact formula for sample sizes up to 31, six recursion formulae, and one matrix formula that can be used to calculate a p value. To ensure accurate calculation by avoiding catastrophic cancelation and eliminating rounding error, each of these formulae is implemented in rational arithmetic. For the six recursion formulae and the matrix formula, computational experience for sample sizes up to 500 shows that computational times are increasing functions of both the sample size and the number of digits in the numerator and denominator integers of the rational number test statistic. The computational times of the seven formulae vary immensely but the Durbin recursion formula is almost always the fastest. Linear search is used to calculate the inverse of the cumulative sampling distribution (find the confidence interval half-width) and tables of calculated half-widths are presented for sample sizes up to 500. Using calculated half-widths as input, computational times for the fastest formula, the Durbin recursion formula, are given for sample sizes up to two thousand.

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

1. Introduction

The Kolmogorov-Smirnov (K-S) family of tests is one of the most widely used goodness-of-

fit tests and is included in some form in many nonparametric statistics texts (Sprent and Smeeton 2001; Gibbons and Chakraborti 2003; Conover 1999; Daniel 1990). These include the one-sided one sample, two-sided one sample, one-sided two sample, and the two-sided two sample tests. The K-S family of tests also include restricted range tests (comparing distributions over a portion of their range) and ratio tests (the ratio of one distribution to another).

For sample size n, the most common K-S test is the two-sided one sample test which uses the maximum absolute distance D_n as the random variable where D_n is the distance between the hypothesized continuous cumulative distribution $F(x)$ and the empirical cumulative distribution $F_n(x)$, $D_n = \sup_{-\infty<x<\infty} |F_n(x) - F(x)|$. The distribution of D_n is independent of the hypothesized continuous cumulative distribution so the sampling distribution of D_n can be derived by using the uniform distribution on the interval $[0, 1]$ as $F(x)$. In a hypothesis testing application, computing the test statistic d is relatively easier than evaluating the cumulative sampling distribution to determine the p value, $P[D_n \geq d]$.

The cumulative sampling distribution for the two-sided one sample K-S test is a piecewise polynomial that is different for each sample size n and the complexity rapidly grows with increasing n so it has not been generated let alone used for $n > 31$ (Ruben and Gambino 1982; Drew, Glen, and Leemis 2000). Consequently, the limiting distribution, various recursion formulae, and various approximations have been used to evaluate the cumulative sampling distribution. Many computer statistical software packages such as SPSS, STATISTICA, R, Numerical Recipes, and IMSL include one or more K-S tests. Although a recursion formula will theoretically determine the p value $P[D_n \geq d]$ for a particular value d of the test statistic, the complexity of the formula is such that roundoff error and catastrophic cancelation can greatly reduce the accuracy of the calculations. Since most procedures used today were developed on pre-1978 computers where only machine precision was available, the accuracy of their results is not known exactly. Consequently, recursion formulae have only been used to generate tables for sample sizes of $n \leq 40$ and various approximations of unknown accuracy have been used for $n > 40$.

In addition to the two-sided one sample case (absolute difference between hypothesized and empirical), the one-sided one sample (difference between hypothesized and empirical) cumulative sampling distribution is also a complex series that can also be evaluated by recursion formulae. Indeed, many computer statistical software packages that implement both the two-sided and one-sided one sample K-S test use different methods to calculate the p values. Brown and Harvey (2007) have summarized how various computer statistical software packages calculate the p values for the one sample K-S tests. Since they have investigated the one-sided one sample case, this paper examines the two-sided one sample case. Following their recommendation, rational arithmetic is used to eliminate all roundoff and catastrophic cancelation error. The following sections review the two-sided one sample K-S cumulative sampling distribution formulae, devise rational arithmetic implementations of each formula using Mathematica 5 (Wolfram 2003), verify the validity of each implementation by determining if each implementation gets exactly the same p value over a broad range of examples, compare the computational times needed for each implementation to determine the fastest formula, develop an efficient method to calculate the half-width (the inverse of the cumulative sampling distribution), and give computation times for the fastest formula.

2. The exact formula

The cumulative sampling distribution for D_n is a piecewise polynomial whose complexity grows with increasing sample size n. For example, the cumulative sampling distribution for $n = 5$ with nine regions is shown below.

$$P(D_5 < d) = \begin{cases} 0 & \text{for } d < \frac{1}{10} \\ \frac{24}{625}(10d-1)^5 & \text{for } \frac{1}{10} \le d < \frac{2}{10} \\ -288d^4 + 240d^3 - \frac{1464}{25}d^2 + \frac{672}{125}d - \frac{96}{625} & \text{for } \frac{2}{10} \le d < \frac{3}{10} \\ 160d^5 - 240d^4 + \frac{424}{5}d^3 + 12d^2 - \frac{168}{25}d + \frac{336}{625} & \text{for } \frac{3}{10} \le d < \frac{4}{10} \\ -20d^5 + 74d^4 - \frac{456}{5}d^3 + \frac{224}{5}d^2 - \frac{728}{125}d & \text{for } \frac{4}{10} \le d < \frac{5}{10} \\ 12d^5 - 6d^4 - \frac{56}{5}d^3 + \frac{24}{5}d^2 + \frac{522}{125}d - 1 & \text{for } \frac{5}{10} \le d < \frac{6}{10} \\ -8d^5 + 22d^4 - \frac{92}{5}d^3 + \frac{12}{25}d^2 + \frac{738}{125}d - 1 & \text{for } \frac{6}{10} \le d < \frac{8}{10} \\ 2d^5 - 10d^4 + 20d^3 - 20d^2 + 10d - 1 & \text{for } \frac{8}{10} \le d < 1 \\ 1 & \text{for } d \ge 1 \end{cases}$$

In general, $\lceil 3n/2 \rceil - 1$ is the number of piecewise polynomials for $1/2n \le d < 1$ where $\lceil 3n/2 \rceil$ is the smallest integer greater than or equal to $3n/2$.

Ruben and Gambino (1982) computed the coefficients of the piecewise polynomials for $n \le 10$ and Drew, Glen, and Leemis (2000) using rational arithmetic in Maple (Maplesoft 2008) computed them for $n \le 31$. The computer storage requirements are immense. For example, for $n = 31$ there are 46 piecewise polynomials requiring more than 20 pages to print and the file to store all the polynomials for sample sizes $n \le 31$ requires 678,000 bytes of storage while the storage for $n = 31$ alone is 91,000 bytes.

Since all calculations in this paper are performed in Mathematica, the Maple program of Drew, Glen, and Leemis (2000) was translated to Mathematica and is contained the file `ExactDistribution.nb`. Although the Mathematica code is large (over 1.5 million bytes), it is simply a list of rational arithmetic equations where $d = t/n$ is used to find the correct equation and that equation is then evaluated to give the p value. The intermediate output for `intermediateOutput` ≥ 1 gives the input parameters t and n as well as the left tail probability $P(D_n < t/n)$ and the right tail probability $P(D_n \ge t/n)$.

The Mathematica code in file `ExactKS2SidedOneSample.nb` assigns the internal variable `tRational` the rational number value of the input variable `tIn` and also assigns the internal variable `sampleSize` the truncated value of the input variable `sampleSizeIn`. The internal variable `x` is then assigned the rational number value of `tRational/sampleSize`. Note that internal variable `x` is the same as $d = t/n$ used in the formulae above. To check the accuracy of the Mathematica code in file `ExactKS2SidedOneSample.nb`, various p values for $n \le 31$ generated by the Mathematica code and the Maple code were compared and found to be identical.

Since the Mathematica code for the exact formula in file `ExactKS2SidedOneSample.nb` requires a large amount of storage and can only be used for sample sizes less than or equal to thirty-one, $n \leq 31$, alternate formulae contained in the literature for calculating the p values of sample sizes greater than $n = 31$ need to be investigated. An extensive literature search identified six recursion formulae (by Kolmogorov, Tingey, Epanechnikov, Noe, Durbin, Pomeranz) and a matrix formula (by Durbin) which can be used for sample sizes exceeding $n = 31$. In Sections 3 through 9, these seven formulae are presented, transformed for implementation, and coded in Mathematica. Section 10 contains a summary of the Mathematica code implementing the seven formulae and describes a program for $n \leq 31$ that checks whether the seven formulae generate the same p values as those produced by the exact formula. Due to the large size of file `ExactKS2SidedOneSample.nb`, the Mathematica code for these seven formulae are contained in another file, `KS2SidedOneSampleRational.nb`.

For four regions and all sample sizes n, Ruben and Gambino (1982) were able to derive the simple formulae shown below. In the implementation of the six recursion formulae and the matrix formula, the formulae below will be used in those regions because they are faster than the recursion and matrix formulae. Thus, the recursion and matrix formulae will only be used for $1 < t < n - 1$.

$$P\left(D_n \geq \frac{t}{n}\right) = \begin{cases} 1 & \text{for } t < \dfrac{1}{2} \\ 1 - \dfrac{n!}{n^n}(2t-1)^n & \text{for } \dfrac{1}{2} \leq t \leq 1 \\ 2\left(1 - \dfrac{t}{n}\right)^n & \text{for } n-1 \leq t \leq n \\ 0 & \text{for } t \geq n \end{cases}$$

For sample size $n = 3$, the exact formula for $1/2 < t < 3$ is shown below. To demonstrate the equivalence of the recursion and matrix formulae to the exact formula, the exact formula below will be derived for the Tingey, Epanechnikov, Durbin, and Durbin matrix formulae.

$$P\left(D_3 \geq \frac{t}{3}\right) = \begin{cases} 1 - \dfrac{2}{9}(2t-1)^3 & \text{for } \dfrac{1}{2} < t < 1 \\ \dfrac{4}{9}t^3 - \dfrac{14}{9}t^2 + \dfrac{8}{9}t + 1 & \text{for } 1 < t \leq \dfrac{3}{2} \\ \dfrac{4}{27}t^3 - \dfrac{2}{9}t^2 - \dfrac{10}{9}t + 2 & \text{for } \dfrac{3}{2} \leq t < 2 \\ \dfrac{2}{27}(3-t)^3 & \text{for } 2 \leq t < 3 \end{cases}$$

3. The Kolmogorov recursion formula

A recursion formula was derived by Kolmogorov (1933) to compute the cumulative probability $P(D_n \geq t/n)$ for integer $t = 1, 2, \ldots, n-1$. The Kolmogorov recursion formula can be transformed into the recursion formula of Massey (1950) which is shown below using Tingey's notation (Tingey 1951).

$$
\begin{aligned}
R_{t,0}(t) &= 1 \\
R_{j,0}(t) &= 0 && \text{for } j \neq t \\
R_{j,k}(t) &= 0 && \text{for } j \leq 0 \text{ and } j \geq 2t \\
R_{j,k+1}(t) &= \sum_{\mu=0}^{j} \frac{R_{j+1-\mu,k}(t)}{\mu!} && \text{for } 1 \leq j \leq 2t-1 \\
P\left(D_n \geq \frac{t}{n}\right) &= 1 - \frac{n!}{n^n} R_{t,n}(t)
\end{aligned}
$$

3.1. Transforming the Kolmogorov recursion formula

For the first iteration with $k = 0$, the recursion formula can be simplified by noting that the only non-zero term in the formula for $R_{j,k+1}(t) = R_{j,1}(t)$ is when $j = t$ because $R_{t,0}(t) = 1$ and $R_{j,0}(t) = 0$ for $j \neq t$. In other words, $R_{j,1}(t) = R_{j+1-\mu,0}(t)/\mu!$ where $j + 1 - \mu = t$ or $\mu = j+1-t$. For the index $\mu = 0, 1, \ldots, j$, the subscript $j+1-\mu = t$ only when $\mu = j+1-t$ which can occur only if $j \geq t-1$. If $j < t-1$, the subscript $j+1-\mu = t$ means $\mu = j+1-t$ but since $j < t-1$, $\mu = j+1-t < 0$ which is impossible because $\mu = 0, 1, \ldots, j$. So for $j < t-1$ or $j \leq t-2$, $R_{j,1}(t) = 0$. Thus, the recursion formula for $R_{j,k+1}(t)$ reduces to the following formulae for $k = 0$.

$$
\begin{aligned}
R_{j,1}(t) &= 0 && \text{for } j \leq t-2 \text{ and } j \geq 2t \\
R_{j,1}(t) &= \frac{1}{(j+1-t)!} && \text{for } j = t-1, t, \ldots, 2t-1
\end{aligned}
$$

For $k = 1$, the formula for $R_{j,k+1}(t) = R_{j,2}(t)$ must contain $R_{m,1}(t)$ for $m \geq t-1$ or it is zero. Since $m = j+1-\mu$ is largest when $\mu = 0$, then $m = j+1 \geq t-1$ for $R_{j,2}(t) \neq 0$ which yields $j \geq t-2$. Thus, for $k = 1$, $R_{j,k+1}(t) = R_{j,2}(t) = 0$ for $j \leq t-3$ and $j \geq 2t$ while $R_{j,k+1}(t) = R_{j,2}(t) \neq 0$ for $t-2 \leq j \leq 2t-1$. Although this restricts the $R_{j,k+1}(t) = R_{j,2}(t)$ terms that need to be calculated (restricts the range of index j), care must be taken so that only non-zero $R_{j+1-\mu,k}(t) = R_{j+1-\mu,1}(t)$ terms are used in the calculations. Using $m = j+1-\mu$, then $R_{m,1}(t) = 0$ for $m \leq t-2$ and $m \geq 2t$ while $R_{m,1}(t) \neq 0$ for $t-1 \leq m \leq 2t-1$. This means the index μ must satisfy the inequality $t-1 \leq m \leq 2t-1$ which by substituting $m = j+1-\mu$ becomes $t-1 \leq j+1-\mu \leq 2t-1$. In addition, index μ must also satisfy the summation index boundaries which are $0 \leq \mu \leq j$. Thus, μ must satisfy the following two inequalities.

$$
\begin{aligned}
t-1 \leq \quad & j+1-\mu \quad && \leq 2t-1 \\
0 \leq \quad & \mu \quad && \leq j
\end{aligned}
$$

Rearranging the first inequality yields

$$
\begin{aligned}
j+2-2t \leq \quad & \mu \quad && \leq j+2-t \\
0 \leq \quad & \mu \quad && \leq j
\end{aligned}
$$

Combining the two inequalities yields

$$\max\{j+2-2t,0\} \le \mu \le \min\{j+2-t,j\}.$$

In general, for $k \ge 1$, the range for zero and non-zero $R_{j,k}(t)$ and $R_{j,k+1}(t)$ terms are shown below.

$$\begin{aligned}
R_{j,k}(t) &= 0 && \text{for } j \le \max\{t-k-1,0\} \text{ and } j \ge 2t \\
R_{j,k}(t) &\ne 0 && \text{for } \max\{t-k-1,0\}+1 \le j \le 2t-1 \\
R_{j,k+1}(t) &= 0 && \text{for } j \le \max\{t-k-2,0\} \text{ and } j \ge 2t \\
R_{j,k+1}(t) &\ne 0 && \text{for } \max\{t-k-2,0\}+1 \le j \le 2t-1
\end{aligned}$$

These results show that in the formula to calculate $R_{j,k+1}(t)$, the index j must be restricted to $\max\{t-k-2,0\}+1 \le j \le 2t-1$. To derive the restrictions for the index μ, note that $R_{m,k}(t)$ where $m = j+1-\mu$ must have the index in the range $\max\{t-k-1,0\}+1 \le j \le 2t-1$. Thus, μ must satisfy the following two inequalities.

$$\begin{aligned}
\max\{t-k-1,0\}+1 \le &\quad j+1-\mu &&\le 2t-1 \\
0 \le &\quad \mu &&\le j
\end{aligned}$$

Rearranging the first inequality yields

$$\begin{aligned}
j+2-2t \le &\quad \mu &&\le j-\max\{t-k-1,0\} \\
0 \le &\quad \mu &&\le j
\end{aligned}$$

Combining the two inequalities yields

$$\max\{j+2-2t,0\} \le \quad \mu \quad \le \min\{j-\max\{t-k-1,0\},j\}.$$

These inequalities produce the following Kolmogorov recursion formula.

$$\begin{aligned}
R_{t,0}(t) &= 1 && \\
R_{j,0}(t) &= 0 && \text{for } j \ne t \\
R_{j,1}(t) &= 0 && \text{for } j \le \max\{t-2,0\} \text{ and } j \ge 2t \\
R_{j,1}(t) &= \frac{1}{(j+1-t)!} && \text{for } \max\{t-2,0\}+1 \le j \le 2t-1 \\
R_{j,k}(t) &= 0 && \text{for } j \le \max\{t-k-2,0\} \text{ and } j \ge 2t \\
R_{j,k+1}(t) &= \sum_{\mu=\max\{j+2-2t,0\}}^{\min\{j-\max\{t-k-1,0\},j\}} \frac{R_{j+1-\mu,k}(t)}{\mu!} && \text{for } \max\{t-k-2,0\}+1 \le j \le 2t-1
\end{aligned}$$

$$P\left(D_n \ge \frac{t}{n}\right) = 1 - \frac{n!}{n^n} R_{t,n}(t)$$

For the Kolmogorov recursion formula, the user specifies an integer t and a sample size n. Corresponding to $k=1$, the values of $R_{j,1}(t)$ for all j are initialized. Then corresponding to $k=2$, the values of $R_{j,2}(t)$ for all j are computed using the values $R_{j,1}(t)$ corresponding to $k=1$. Each succeeding iteration increases k by one and computes the $R_{j,k+1}(t)$ for all

Recursion function variable	Mathematica function variable	Recursion function variable	Mathematica function variable
t	`tRational`	μ	`muIndex`
n	`maxSampleSize`	$R_{j,k}(t)$	`oldRvector[[jIndex ]]`
j	`jIndex`	$R_{j,k+1}(t)$	`newRvector[[jIndex ]]`
k	`k`	$P\left(D_n \geq \frac{t}{n}\right)$	`rightTailProbability`
$k+1$	`sampleSize`		

Table 1: Kolmogorov recursion formula and Mathematica function variables.

j using the $R_{j,k}(t)$ values computed in the previous iteration. The iterations stop when the $R_{j,n}(t)$ values have been computed for $k = n$. At this point, the cumulative probability $P(D_n \geq t/n) = 1 - \frac{n!}{n^n} R_{t,n}(t)$ can be computed. At the end of each iteration which produces the $R_{j,k}(t)$ values, the cumulative probability for a sample size of k when $t < k$ can be computed by $P(D_k \geq t/k) = 1 - \frac{k!}{k^k} R_{t,k}(t)$.

3.2. Kolmogorov recursion formula and Mathematica function variables

Table 1 contains the relationship between the variables in the Kolmogorov recursion formula above and the variables in the Mathematica function `KolmogorovKS2SidedRTProbsRational` contained in Section 1 of the `KS2SidedOneSampleRational.nb` file. This function has four arguments `[tIn_, internalPrecision_, maxSampleSizeIn_, intermediateOutput_]`. `tRational` is set equal to the truncated value of `tIn` and `maxSampleSize` is set equal to the truncated value of `maxSampleSizeIn` so that the input values `tIn` and `maxSampleSizeIn` are not changed by the function.

4. The Tingey recursion formula

A recursion formula for both integer and noninteger t, $1/2 < t < n$, was derived by Tingey (1951). Since it is not convenient in Mathematica to use negative indices, the Tingey recursion formula is transformed to the form shown below that has no negative indices. Note that $s = \lfloor t \rfloor$ is the largest integer less than or equal to t and $j = 1, 2, \ldots, 2s$.

$$
\begin{aligned}
R_{s+1,0}(t) &= 1 \\
R_{j,0}(t) &= 0 \quad \text{for } j \neq s+1 \\
R_{j,k}(t) &= 0 \quad \text{for } \begin{cases} j \leq 0 \text{ and} \\ j \geq 2s+2 \end{cases}
\end{aligned}
$$

$$
\begin{aligned}
R_{j,k+1}(t) &= \frac{\left[1-(s-t+1)^j\right] R_{1,k}(t)}{j!} + \sum_{\mu=0}^{j-1} \frac{R_{j+1-\mu,k}(t)}{\mu!} && \text{for } 1 \le j \le 2s \\
R_{2s+1,k+1}(t) &= \frac{\left[1-2(s-t+1)^{2s+1}+(2s-2t+1)^{2s+1}\right] R_{1,k}(t)}{(2s+1)!} \\
&\quad + \sum_{\mu=1}^{2s} \frac{\left[1-(s+1-t)^{\mu}\right] R_{2s+2-\mu,k}(t)}{\mu!} && \text{if } s \ge t-\tfrac{1}{2} \\
R_{2s+1,k+1}(t) &= \frac{\left[1-2(s-t+1)^{2s+1}\right] R_{1,k}(t)}{(2s+1)!} \\
&\quad + \sum_{\mu=1}^{2s} \frac{\left[1-(s+1-t)^{\mu}\right] R_{2s+2-\mu,k}(t)}{\mu!} && \text{if } s < t-\tfrac{1}{2} \\
P\left(D_n \ge \frac{t}{n}\right) &= 1 - \frac{n!}{n^n} R_{s+1,n}(t) && \text{for } \tfrac{1}{2} < t < n
\end{aligned}
$$

To simplify the Tingey recursion formula, define $W_{\mu,j,k}(t)$ as shown below.

$$
W_{\mu,j,k}(t) = \begin{cases}
\dfrac{R_{j+1-\mu,k}(t)}{\mu!} & \text{for } \begin{cases} 1 \le j \le 2s \text{ and} \\ 0 \le \mu \le j-1 \end{cases} \\
\dfrac{\left[1-(s-t+1)^j\right] R_{1,k}(t)}{j!} & \text{for } \begin{cases} 1 \le j \le 2s \\ \text{and } \mu = j \end{cases} \\
0 & \text{for } \begin{cases} j = 2s+1 \\ \text{and } \mu = 0 \end{cases} \\
\dfrac{\left[1-(s+1-t)^{\mu}\right] R_{2s+2-\mu,k}(t)}{\mu!} & \text{for } \begin{cases} j = 2s+1 \text{ and} \\ 1 \le \mu \le 2s \end{cases} \\
\dfrac{\left[1-2(s-t+1)^{2s+1}+(2s-2t+1)^{2s+1}\right] R_{1,k}(t)}{(2s+1)!} & \text{for } \begin{cases} j = 2s+1, \\ \mu = 2s+1, \\ \text{and } s \ge t-\dfrac{1}{2} \end{cases} \\
\dfrac{\left[1-2(s-t+1)^{2s+1}\right] R_{1,k}(t)}{(2s+1)!} & \text{for } \begin{cases} j = 2s+1, \\ \mu = 2s+1, \\ \text{and } s < t-\dfrac{1}{2} \end{cases}
\end{cases}
$$

For each $W_{\mu,j,k}(t)$ above, there is one R term and the first index of the R term can be written as $j+1-\mu$. This is even true for $W_{0,2s+1,k}(t)$ where $j = 2s+1$ and $\mu = 0$ because in this case $j+1-\mu = 2s+1+1-0 = 2s+2$ so $R_{j+1-\mu,k}(t) = R_{2s+2,k}(t) = 0$ by definition. With this observation, the definition of $W_{\mu,j,k}(t)$ can use $j+1-\mu$ as the first index of each R term as shown below. In addition, the condition $s \ge t - 1/2$ in the next to last term can be written as $t - s \le 1/2$ while the condition $s < t - 1/2$ in the last term can be written as $t - s > 1/2$.

$$W_{\mu,j,k}(t) = \begin{cases} \dfrac{R_{j+1-\mu,k}(t)}{\mu!} & \text{for } 1 \le j \le 2s \text{ and } 0 \le \mu \le j-1 \\ \dfrac{\left[1-(s-t+1)^j\right] R_{j+1-\mu,k}(t)}{j!} & \text{for } 1 \le j \le 2s \text{ and } \mu = j \\ R_{j+1-\mu,k}(t) = R_{2s+2,k}(t) = 0 & \text{for } j = 2s+1 \text{ and } \mu = 0 \\ \dfrac{\left[1-(s+1-t)^\mu\right] R_{j+1-\mu,k}(t)}{\mu!} & \text{for } j = 2s+1 \text{ and } 1 \le \mu \le 2s \\ \dfrac{\left[1-2(s-t+1)^{2s+1}+(2s-2t+1)^{2s+1}\right] R_{j+1-\mu,k}(t)}{(2s+1)!} & \text{for } j = 2s+1, \mu = 2s+1, \text{ and } t-s \le \dfrac{1}{2} \\ \dfrac{\left[1-2(s-t+1)^{2s+1}\right] R_{j+1-\mu,k}(t)}{(2s+1)!} & \text{for } j = 2s+1, \mu = 2s+1, \text{ and } t-s > \dfrac{1}{2} \end{cases}$$

Note that all the coefficients of $R_{j+1-\mu,k}(t)$ must be positive except for the last two terms which can be negative. Thus, unlike the Kolmogorov recursion formula, some of the $W_{\mu,j,k}(t)$'s as well as the $R_{j,k}(t)$'s can be negative.

Using the definition of $W_{\mu,j,k}(t)$ above, the Tingey recursion formula can now be written in a much simpler form.

$$\begin{aligned} R_{s+1,0}(t) &= 1 \\ R_{j,0}(t) &= 0 && \text{for } j \ne s+1 \\ R_{j,k}(t) &= 0 && \text{for } j \le 0, j \ge 2s+2, \text{ and } 0 \le k \le n \\ R_{j,k+1}(t) &= \sum_{\mu=0}^{j} W_{\mu,j,k}(t) && \text{for } 1 \le j \le 2s+1 \text{ and } 0 \le k \le n-1 \\ P\left(D_n \ge \frac{t}{n}\right) &= 1 - \frac{n!}{n^n} R_{s+1,n}(t) && \text{for } \tfrac{1}{2} < t < n \end{aligned}$$

To derive the non-zero terms in computing $R_{j,1}(t)$ for $k = 0$, note that only $R_{s+1,0}(t) \ne 0$ so the first index of the R term in $W_{\mu,j,k}(t) = W_{\mu,j,0}(t)$ is $j+1-\mu$ and must be $j+1-\mu = s+1$ for the corresponding R term to be non-zero. Solving $j+1-\mu = s+1$ for μ yields $\mu = j-s$. The two inequalities that μ and j must satisfy are shown below.

$$\begin{aligned} 0 \le &\quad \mu &&\le j \\ 1 \le &\quad j &&\le 2s+1 \end{aligned}$$

Substituting $\mu = j - s$ into the two inequalities yields

$$\begin{aligned} 0 \le &\quad j-s &&\le j \\ 1 \le &\quad\quad j &&\le 2s+1 \end{aligned}$$

which can only be satisfied if $j \geq s$. Thus, $W_{\mu,j,0}(t) = 0$ for $j \leq \max\{s-1, 0\}$ and $W_{\mu,j,0}(t) \neq 0$ for $\max\{s, 1\} \leq j \leq 2s+1$. The exact formulae for all $R_{j,1}(t)$ can be derived and are shown below. When $j = 2s+1$, $\mu = j - s = s+1$ which produces the term $W_{s+1,2s+1,0}(t) = W_{j-s,j,0}(t)$.

$$\begin{aligned} R_{j,1}(t) &= 0 && \text{for } j \leq \max\{s-1, 0\} \text{ and } j \geq 2s+2 \\ R_{j,1}(t) &= W_{j-s,j,0}(t) && \text{for } \max\{s, 1\} \leq j \leq 2s \end{aligned}$$

Using formulae for $W_{\mu,j,0}(t)$ where $R_{s+1,0}(t) = 1$ yields the following formulae.

$$\begin{aligned} R_{j,1}(t) &= 0 && \text{for } j \leq \max\{s-1, 0\} \text{ and } j \geq 2s+2 \\ R_{j,1}(t) &= \frac{1}{(j-s)!} && \text{for } \max\{s, 1\} \leq j \leq 2s \\ R_{j,1}(t) &= \frac{1-(s+1-t)^{2s+1}}{(2s+1)!} && \text{for } j = 2s+1 \end{aligned}$$

In general, as k increases one more non-zero term is added so $R_{j,k}(t) = 0$ for $j \leq \max\{s-k, 0\}$ and $j \geq 2s+2$ while $R_{j,k}(t) \neq 0$ for $\max\{s+1-k, 1\} \leq j \leq 2s+1$. To calculate $R_{j,k+1}(t)$ for $\max\{s-k, 1\} \leq j \leq 2s+1$, zero terms of $W_{\mu,j,k}(t)$ can be avoided if the range of μ satisfies the two inequalities shown below.

$$\begin{aligned} \max\{s+1-k, 1\} \leq \quad j+1-\mu \quad &\leq 2s+1 \\ 0 \leq \quad \mu \quad &\leq j \end{aligned}$$

Rearranging the first inequality yields

$$\begin{aligned} j-2s \leq \quad \mu \quad &\leq j+1-\max\{s+1-k, 1\} \\ 0 \leq \quad \mu \quad &\leq j \end{aligned}$$

Combining the two inequalities yields

$$\max\{j-2s, 0\} \leq \mu \leq \min\{j+1-\max\{s+1-k, 1\}, j\}.$$

These inequalities produce the following Tingey recursion formula.

$$\begin{aligned} R_{j,1}(t) &= 0 && \text{for } j \leq \max\{s-1, 0\} \text{ and } j \geq 2s+2 \\ R_{j,1}(t) &= \frac{1}{(j-s)!} && \text{for } \max\{s, 1\} \leq j \leq 2s \\ R_{j,1}(t) &= \frac{1-(s+1-t)^{2s+1}}{(2s+1)!} && \text{for } j = 2s+1 \\ R_{j,k}(t) &= 0 && \text{for } j \leq 0, j \geq 2s+2, \text{ and } 0 \leq k \leq n \\ R_{j,k+1}(t) &= \sum_{\mu=\max\{j-2s,0\}}^{\min\{j+1-\max\{s+1-k,1\},j\}} W_{\mu,j,k}(t) && \text{for } \begin{cases} \max\{s-k, 1\} \leq j \leq 2s+1 \\ \text{and } 1 \leq k \leq n-1 \end{cases} \\ P\left(D_n \geq \frac{t}{n}\right) &= 1 - \frac{n!}{n^n} R_{s+1,n}(t) && \text{for } \tfrac{1}{2} < t < n \end{aligned}$$

4.1. The Tingey recursion formula for small test statistic values

The Tingey recursion formula can be considerably simplified when $1/2 < t < 1$. In this case, $s = 0$ and $s < t - 1/2$ so the Tingey recursion formula given in the beginning of Section 4 reduces to the following form.

$$\begin{aligned}
R_{0,0}(t) &= 1 \\
R_{i,0}(t) &= 0 && \text{for } i \neq 0 \\
R_{i,k}(t) &= 0 && \text{for } i \neq 0 \\
R_{0,k+1}(t) &= [1 - 2(-t+1)]\, R_{0,k}(t) = (2t-1)\, R_{0,k}(t) \\
P\left(D_n \geq \frac{t}{n}\right) &= 1 - \frac{n!}{n^n} R_{0,n}(t) && \text{for } \tfrac{1}{2} < t < 1
\end{aligned}$$

Since $R_{0,0}(t) = 1$, then for $k = 0$

$$R_{0,1}(t) = (2t-1)\, R_{0,0}(t) = 2t - 1.$$

Then for $k = 1$, the formula becomes

$$R_{0,2}(t) = (2t-1)\, R_{0,1}(t) = (2t-1)^2 .$$

In general, the formula for $R_{0,n}(t)$ becomes

$$R_{0,n}(t) = (2t-1)\, R_{0,n-1}(t) = (2t-1)^n .$$

Substituting into the right tail probability formula yields the following formula that is equivalent to the formula in Section 2.

$$P\left(D_n \geq \frac{t}{n}\right) = 1 - \frac{n!}{n^n} R_{0,n}(t) = 1 - \frac{n!\,(2t-1)^n}{n^n} \qquad \text{for } \frac{1}{2} < t < 1 \tag{1}$$

4.2. The Tingey recursion formula for a sample size of three

For sample size $n = 3$ and region $1 < t \leq 3/2$, the following derivation shows that the Tingey recursion formula reduces to the exact formula in Section 2.

$$\begin{aligned}
&s = \lfloor t \rfloor = 1 \qquad s + 1 = 2 \\
&R_{1,0}(t) = 0 \qquad R_{2,0}(t) = 1 \qquad R_{3,0}(t) = 0 \\
&R_{1,1}(t) = 1 \qquad R_{2,1}(t) = 1 \qquad R_{3,1}(t) = \frac{1}{2} - \frac{1}{2}(2-t)^2 \\
&R_{1,2}(t) = t \qquad R_{2,2}(t) = 2 - (2-t)^2 \qquad R_{3,2}(t) = -\frac{3}{2}t^3 + 6t^2 - \frac{13}{2}t + 2 \\
&R_{2,3}(t) = -2t^3 + 7t^2 - 4t \qquad P\left(D_3 \geq \frac{t}{3}\right) = 1 - \frac{2}{9} R_{2,3}(t) = \frac{4}{9}t^3 - \frac{14}{9}t^2 + \frac{8}{9}t + 1
\end{aligned}$$

For sample size $n = 3$ and region $3/2 \le t < 2$, the following derivation shows that the Tingey recursion formula reduces to the exact formula in Section 2.

$$s = \lfloor t \rfloor = 1 \qquad s + 1 = 2$$

$$R_{1,0}(t) = 0 \qquad R_{2,0}(t) = 1 \qquad R_{3,0}(t) = 0$$

$$R_{1,1}(t) = 1 \qquad R_{2,1}(t) = 1 \qquad R_{3,1}(t) = \frac{1}{2} - \frac{1}{2}(2-t)^2$$

$$R_{1,2}(t) = t \qquad R_{2,2}(t) = 2 - (2-t)^2 \qquad R_{3,2}(t) = -\frac{1}{6}t^3 + \frac{5}{2}t - \frac{5}{2}$$

$$R_{2,3}(t) = -\frac{2}{3}t^3 + t^2 + 5t - \frac{9}{2} \qquad P\left(D_3 \ge \frac{t}{3}\right) = 1 - \frac{2}{9}R_{2,3}(t) = \frac{4}{27}t^3 - \frac{2}{9}t^2 - \frac{10}{9}t + 2$$

For sample size $n = 3$ and region $2 \le t < 3$, the following derivation shows that the Tingey recursion formula reduces to the exact formula in Section 2.

$$s = \lfloor t \rfloor = 2 \qquad s + 1 = 3$$

$$R_{1,0}(t) = 0 \qquad R_{2,0}(t) = 0 \qquad R_{3,0}(t) = 1 \qquad R_{4,0}(t) = 0 \qquad R_{5,0}(t) = 0$$

$$R_{1,1}(t) = 0 \qquad R_{2,1}(t) = 1 \qquad R_{3,1}(t) = 1 \qquad R_{4,1}(t) = \frac{1}{2} \qquad R_{5,1}(t) = \frac{1}{6} - \frac{1}{6}(3-t)^3$$

$$R_{1,2}(t) = 1 \qquad R_{2,2}(t) = 2 \qquad R_{3,2}(t) = 2 \qquad R_{4,2}(t) = \frac{4}{3}t^3 - \frac{1}{6}(3-t)^3$$

$$R_{5,2}(t) = \frac{1}{24}\left(3t^4 - 28t^3 + 84t^2 - 68t - 24\right) \qquad R_{3,3}(t) = \frac{9}{2} - \frac{1}{3}(3-t)^3$$

$$P\left(D_3 \ge \frac{t}{3}\right) = 1 - \frac{2}{9}R_{3,3}(t) = \frac{2}{27}(3-t)^3 = 2\left(1 - \frac{t}{3}\right)^3$$

4.3. Tingey recursion formula and Mathematica function variables

Table 2 contains the relationship between the variables in the recursion formula above and the variables in the Mathematica function `TingeyKS2SidedRTProbsRational` contained in Section 2 of the `KS2SidedOneSampleRational.nb` file. The function `TingeyKS2SidedRTProbsRational` has four arguments `[tIn_, internalPrecision_, maxSampleSizeIn_, intermediateOutput_]`. `tRational` is set equal to the truncated value of `tIn` and `maxSampleSize` is set equal to the truncated value of `maxSampleSizeIn` so that the input values `tIn` and `maxSampleSizeIn` are not changed by the function.

5. The Epanechnikov recursion formula

The following function is needed for the definition of the recursion formula.

$$\lambda(x) = \begin{cases} 0 & \text{if } x \le 0 \\ x & \text{if } 0 < x < 1 \\ 1 & \text{if } x \ge 1 \end{cases}$$

The recursion formula below was derived by Epanechnikov (1969) and also by Steck (1971).

Recursion function variable	Mathematica function variable	Recursion function variable	Mathematica function variable
t	`tRational`	$t - s$	`tRationalMINUSs`
n	`maxSampleSize`	$\frac{1}{\mu!}$	`muFactorialInverse[[muIndex]]`
$s = \lfloor t \rfloor$	`s`	$\frac{1-(s-t+1)^j}{j!}$	`coeffFORmuEQUALjIndex[[jIndex]]`
j	`jIndex`	$\frac{1-2(s-t+1)^{2s+1}}{(2s+1)!}$	`coefftMINUSsOverHalf`
k	`k`	$\frac{(2s-2t+1)^{2s+1}}{(2s+1)!}$	`coefftMINUSsLessHalf`
$k+1$	`sampleSize`	$R_{j,k}(t)$	`oldRvector[[jIndex ]]`
$2s+1$	`maxDimension`	$R_{j,k+1}(t)$	`newRvector[[jIndex ]]`
μ	`muIndex`	$W_{\mu,j,k}(t)$	`term`
$2s-2t+1$	`termSecond`	$P\left(D_n \geq \frac{t}{n}\right)$	`rightTailProbability`

Table 2: Tingey recursion formula and Mathematica function variables.

$$
\begin{aligned}
Q_n(t) &= 1 \\
b_i &= \lambda\left(\frac{t+i-1}{n}\right) && \text{for } i = 1, 2, \ldots, n \\
a_i &= \lambda\left(\frac{i-t}{n}\right) && \text{for } i = 1, 2, \ldots, n \\
Q_i(t) &= \sum_{k=i+1}^{n} \frac{[\lambda(b_{i+1} - a_k)]^{k-i}}{(k-i)!}(-1)^{k-i-1} Q_k(t) && \text{for } i = n-1, n-2, \ldots, 0
\end{aligned}
$$

$$
P\left(D_n \geq \frac{t}{n}\right) = 1 - n! Q_0(t)
$$

5.1. The Epanechnikov recursion formula for a sample size of three

For sample size $n = 3$ and region $1/2 < t < 1$, the following derivation shows that the Epanechnikov recursion formula reduces to the exact formula in Section 2.

$$
b_1 = \frac{t}{3} \qquad b_2 = \frac{t+1}{3} \qquad b_3 = \frac{t+2}{3} \qquad a_1 = \frac{1-t}{3} \qquad a_2 = \frac{2-t}{3} \qquad a_3 = \frac{3-t}{3}
$$

$$
Q_3(t) = 1 \qquad Q_2(t) = \frac{2t-1}{3} \qquad Q_1(t) = \left(\frac{2t-1}{3}\right)^2 \qquad Q_0(t) = \left(\frac{2t-1}{3}\right)^3
$$

$$
P\left(D_3 \geq \frac{t}{3}\right) = 1 - 6Q_0(t) = 1 - 6\left(\frac{2t-1}{3}\right)^3 = 1 - \frac{2}{9}(2t-1)^3
$$

For sample size $n = 3$ and region $1 < t \leq 3/2$, the following derivation shows that the Epanechnikov recursion formula reduces to the exact formula in Section 2.

$$b_1 = \frac{t}{3} \qquad b_2 = \frac{t+1}{3} \qquad b_3 = 1 \qquad a_1 = 0 \qquad a_2 = \frac{2-t}{3} \qquad a_3 = \frac{3-t}{3}$$
$$Q_3(t) = 1 \qquad Q_2(t) = \frac{t}{3} \qquad Q_1(t) = \frac{1}{3}t - \frac{2}{9} \qquad Q_0(t) = -\frac{2}{27}t^3 + \frac{7}{27}t^2 + \frac{4}{27}t$$
$$P\left(D_3 \geq \frac{t}{3}\right) = 1 - 6Q_0(t) = \frac{4}{9}t^3 - \frac{14}{9}t^2 + \frac{8}{9}t + 1$$

For sample size $n = 3$ and region $3/2 \leq t < 2$, the following derivation shows that the Epanechnikov recursion formula reduces to the exact formula in Section 2.

$$b_1 = \frac{t}{3} \qquad b_2 = \frac{t+1}{3} \qquad b_3 = 1 \qquad a_1 = 0 \qquad a_2 = \frac{2-t}{3} \qquad a_3 = \frac{3-t}{3}$$
$$Q_3(t) = 1 \qquad Q_2(t) = \frac{t}{3} \qquad Q_1(t) = \frac{1}{3}t - \frac{2}{9} \qquad Q_0(t) = \frac{1}{162}\left(-4t^3 + 6t^2 + 30t - 27\right)$$
$$P\left(D_3 \geq \frac{t}{3}\right) = 1 - 6Q_0(t) = \frac{4}{27}t^3 - \frac{2}{9}t^2 - \frac{10}{9}t + 2$$

For sample size $n = 3$ and region $2 \leq t < 3$, the following derivation shows that the Epanechnikov recursion formula reduces to the exact formula in Section 2.

$$b_1 = \frac{t}{3} \qquad b_2 = 1 \qquad b_3 = 1 \qquad a_1 = 0 \qquad a_2 = 0 \qquad a_3 = \frac{3-t}{3}$$
$$Q_3(t) = 1 \qquad Q_2(t) = \frac{t}{3} \qquad Q_1(t) = \frac{t}{3} + \frac{1}{2}\left(\frac{t}{3}\right)^2 \qquad Q_0(t) = \left(\frac{t}{3}\right)^2 - \left(\frac{t}{3}\right)^3 + \frac{1}{6}\left(\frac{2t-3}{3}\right)^3$$
$$P\left(D_3 \geq \frac{t}{3}\right) = 1 - 6Q_0(t) = 1 - 6\left(\left(\frac{t}{3}\right)^2 - \left(\frac{t}{3}\right)^3 + \frac{1}{6}\left(\frac{2t-3}{3}\right)^3\right) = \frac{2}{27}(3-t)^3$$

5.2. Epanechnikov recursion formula and Mathematica function variables

Table 3 contains the relationship between the variables in the recursion formula above and the variables in the Mathematica function `EpanechnikovKS2SidedRTProbRational` contained in Section 3 of the `KS2SidedOneSampleRational.nb` file. The function `EpanechnikovKS2SidedRTProbRational` has four arguments `[tIn_, internalPrecision_, sampleSizeIn_, intermediateOutput_]`. `t` is set equal to the rational number value of `tIn` and `sampleSize` is set equal to the truncated value of `sampleSizeIn` so that the input values `tIn` and `sampleSizeIn` are not changed by the function.

6. The Noe recursion formula

The general recursion formula by Noe (1972) contains both the two-sided and the one-sided K-S statistics as special cases. Let Z be any random variable with distribution function $P(Z \leq z) = F(z)$, not necessarily continuous. Let $X_1^n \leq X_2^n \leq \cdots \leq X_n^n$ be the order

Recursion function variable	Mathematica function variable	Recursion function variable	Mathematica function variable
t	`t`	a_i	`a[[i]]`
n	`sampleSize`	b_i	`b[[i]]`
$n+1$	`sampleSizePlusOne`	$Q_i(t)$	`Q[[i+1]]`
i	`i`	$(-1)^{k-i-1}$	`SIGN`
k	`k`	$[\lambda(b_{i+1}-a_k)]^{k-i}$	`termNumerator`
$i!$	`FACT[[i]]`	$\frac{[\lambda(b_{i+1}-a_k)]^{k-i}}{(k-i)!}(-1)^{k-i-1}Q_k(t)$	`term`
		$P\left(D_n \geq \frac{t}{n}\right)$	`rightTailProbability`

Table 3: Epanechnikov recursion formula and Mathematica function variables.

statistics of a size n sample from $F(z)$. Let $\{\alpha_j, \beta_j; j = 1, 2, \cdots, n\}$ be any numbers which are called α-boundaries and the β-boundaries respectively. We are interested in the probability

$$P_n = P(\alpha_j < X_j^n \leq \beta_j | j = 1, 2, \cdots, n). \tag{2}$$

Such probabilities are related to the K-S statistics by defining

$$L = \sup_z \sqrt{n}\,[F^n(z) - F(z)]\,\psi_L[F(z)], \tag{3}$$

$$M = \sup_z \sqrt{n}\,[F(z) - F^n(z)]\,\psi_M[F(z)], \tag{4}$$

where $F^n(z)$ is the empirical distribution function of the sample and $\psi_L(x)$, $\psi_M(x)$ are some nonnegative weight functions. A two-sided statistic of the K-S type has the distribution function $P(L \leq \lambda, M \leq \lambda)$ which is of the form (2) for well chosen boundaries.

6.1. The two-sided K-S statistic

To derive the special case for the two-sided K-S statistic, the two weight functions are set equal to one, $\psi_L[F(z)] = \psi_M[F(z)] = 1$. In addition, the α-boundaries and the β-boundaries must be carefully defined. Since the distribution function $F(z)$ is an increasing function, Equation (2) can be written as

$$P_n = P\left(F(\alpha_j) < F(X_j^n) \leq F(\beta_j) | j = 1, 2, \cdots, n\right) \tag{5}$$

or as

$$P_n = P\left(F(\alpha_j) - F^n(X_j^n) < F(X_j^n) - F^n(X_j^n) \leq F(\beta_j) - F^n(X_j^n) | j = 1, 2, \cdots, n\right). \quad (6)$$

The empirical distribution function $F^n(z)$ at X_j^n is a step function that goes from $(j-1)/n$ to j/n. To make the lower limit $F(\alpha_j) - F^n(X_j^n)$ in Equation (6) as small as possible, set $F^n(X_j^n) = j/n$ and to make the upper limit $F(\beta_j) - F^n(X_j^n)$ as large as possible, set $F^n(X_j^n) = (j-1)/n$ yielding

$$P_n = P\left(F(\alpha_j) - \frac{j}{n} < F(X_j^n) - F^n(X_j^n) \leq F(\beta_j) - \frac{(j-1)}{n} | j = 1, 2, \cdots, n\right). \quad (7)$$

Since the K-S two-sided distribution function is also $P(L \leq \lambda, M \leq \lambda)$, Equation (7) can be used to redefine L and M from Equations (3) and (4). First, $M \leq \lambda$ corresponds to the right side of the inequality with the weight function $\psi_M[F(z)] = 1$ which yields

$$M = \sup_z \sqrt{n}\left[F(z) - F^n(z)\right] \leq \sup_z \sqrt{n}\left[F(\beta_j) - \frac{(j-1)}{n}\right] = \lambda. \quad (8)$$

Similarly for L,

$$L = \sup_z \sqrt{n}\left[F(z) - F^n(z)\right] \leq \sup_z \sqrt{n}\left[\frac{j}{n} - F(\alpha_j)\right] = \lambda. \quad (9)$$

Since the inequality in both Equations (8) and (9) needs to be satisfied for all z (all $j = 1, 2, \cdots, n$), then each $F(\alpha_j)$ and $F(\beta_j)$ can be expressed in terms of λ yielding for L

$$F(\alpha_j) = \frac{j}{n} - \frac{\lambda}{\sqrt{n}} \quad (10)$$

and for M

$$F(\beta_j) = \frac{(j-1)}{n} + \frac{\lambda}{\sqrt{n}}. \quad (11)$$

One of the requirements for the $F(\alpha_j)$ and $F(\beta_j)$ is that $F(\alpha_j) < F(\beta_j)$. Using Equations (10) and (11), this requirement becomes

$$\lambda > \frac{1}{2\sqrt{n}}. \quad (12)$$

To transform to the test statistic notation $d = t/n$ from the previous sections, note that $d = \frac{t}{n} = \frac{\lambda}{\sqrt{n}}$ or $\lambda = \frac{t}{\sqrt{n}}$ which when substituted into Inequality (12) above yields $t > 1/2$ as expected. Using t, Equation (10) becomes

$$F(\alpha_j) = \frac{j}{n} - \frac{t}{n} = \frac{j-t}{n} \quad (13)$$

and Equation (11) becomes

$$F(\beta_j) = \frac{(j-1)}{n} + \frac{t}{n} = \frac{j-1+t}{n}. \tag{14}$$

However, using the formulae above can yield $F(\alpha_j) = (j-t)/n < 0$ when $t > j$. Since $F(\alpha_j)$ is a distribution function, it cannot be negative and therefore when $j - t < 0$, it is set to zeero. One way to do this is to set $F(\alpha_j)$ to the maximum between $(j-t)/n$ and zero. This also means that $F(\alpha_0) = 0$.

$$F(\alpha_j) = \max\left[\frac{j-t}{n}, 0\right] \tag{15}$$

Similarly, $F(\beta_j) = (j-1+t)/n > 1$ when $j-1+t > n$ or $t > n-j+1$. Since $F(\beta_j)$ is a distribution function, it cannot exceed one, so it is set equal to one when $t > n-j+1$. One way to do this is to set $F(\beta_j)$ to the minimum between $(j-1+t)/n$ and one.

$$F(\beta_j) = \min\left[\frac{j-1+t}{n}, 1\right] \tag{16}$$

All the $F(\alpha_j)$'s and the $F(\beta_j)$'s are combined into one list and sorted in non-decreasing order. Let $\{F(\gamma_1), F(\gamma_2), \cdots, F(\gamma_{2n-1}), F(\gamma_{2n})\}$ be the sorted list where the origin of each $F(\gamma_j)$ (α-boundary or β-boundary) is kept. Let $F(\alpha_0) = F(\beta_0) = F(\gamma_0) = 0$ and let $F(\alpha_{n+1}) = F(\beta_{n+1} = F(\gamma_{2n+1}) = 1$. Note that Noe (1972) actually sorts the combined list of α_j's and the β_j's but since the distribution function $F(z)$ is a non-decreasing function, this is equivalent to sorting the distribution function where ties are broken by choosing the α-boundary.

Let $g(m)$ for $m = 0, 1, \cdots, 2n$ be the number of α-boundaries in the sorted list $\{F(\gamma_1), F(\gamma_2), \cdots, F(\gamma_m)\}$ where $g(0) = 0$ and $g(2n) = n$. Similarly, let $h(m) - 1$ be the number of β-boundaries in the sorted list $\{F(\gamma_1), F(\gamma_2), \cdots, F(\gamma_{m-1})\}$ where $h(1) = 1$ and $h(2n+1) = n+1$. Finally, let $p_m = F(\gamma_m) - F(\gamma_{m-1})$ where $p_1 = F(\gamma_1)$ since $F(\gamma_0) = 0$.

Noe's recursion relation for the K-S two-sided one sample statistic is shown below where $p_m^0 = 1$ even if $p_m = 0$ since by definition $0^0 = 1$.

$$\begin{aligned}
Q_0(0) &= 1 \\
Q_i(m) &= \sum_{k=h(m)-1}^{i} \binom{i}{k} Q_k(m-1) p_m^{i-k} \quad \text{for} \begin{cases} h(m+1) - 1 \le i \le g(m-1) \\ \text{and } 1 \le m \le 2n \end{cases} \\
Q_{g(m-1)+1}(m) &= 0 \\
P\left(D_n \ge \frac{t}{n}\right) &= 1 - Q_n(2n)
\end{aligned}$$

For each m, the recursion formula needs $Q_k(m-1)$, $h(m+1)-1$, $g(m-1)$, and p_m. The last three quantities can be calculated for all $1 \le m \le 2n$ before the recursion begins to calculate any Q values. To illustrate, Table 4 contains the necessary calculations for $n = 10$ and $t = 3.4$. Before the recursion starts, arrays of $F(\gamma_m)$, $h(m+1)-1$, $g(m-1)$, and p_m are created to be used during the actual recursion.

Since $h(m+1) - 1 \le g(m-1)$, the summation for $Q_i(m)$ is never void. In particular for $m = 1$, $g(m-1) = g(0) = 0$ and $h(m+1) - 1 = h(2) - 1 \le g(m-1) = g(0) = 0$ or

m	$F(\gamma_m)$	Origin	$g(m)$	$h(m)-1$	$h(m+1)-1$	$g(m-1)$	p_m
1	0	$F(\alpha_1)$	1	0	0	0	0
2	0	$F(\alpha_2)$	2	0	0	1	0
3	0	$F(\alpha_3)$	3	0	0	2	0
4	0.06	$F(\alpha_4)$	4	0	0	3	0.06
5	0.16	$F(\alpha_5)$	5	0	0	4	0.1
6	0.26	$F(\alpha_6)$	6	0	0	5	0.1
7	0.34	$F(\beta_1)$	6	0	1	6	0.08
8	0.36	$F(\alpha_7)$	7	1	1	6	0.02
9	0.44	$F(\beta_2)$	7	1	2	7	0.08
10	0.46	$F(\alpha_8)$	8	2	2	7	0.02
11	0.54	$F(\beta_3)$	8	2	3	8	0.08
12	0.56	$F(\alpha_9)$	9	3	3	8	0.02
13	0.64	$F(\beta_4)$	9	3	4	9	0.08
14	0.66	$F(\alpha_{10})$	10	4	4	9	0.02
15	0.74	$F(\beta_5)$	10	4	5	10	0.08
16	0.84	$F(\beta_6)$	10	5	6	10	0.1
17	0.94	$F(\beta_7)$	10	6	7	10	0.1
18	1	$F(\beta_8)$	10	7	8	10	0.06
19	1	$F(\beta_9)$	10	8	9	10	0
20	1	$F(\beta_{10})$	10	9	10	10	0

Table 4: Noe's recursion formula beginning values for $n = 10$ and $t = 3.4$.

$h(2) - 1 = g(0) = 0$. Thus, there is only one term in the summation for $Q_0(1) = Q_0(0) = 1$. To accommodate $Q_0(m,)$ in Mathematica, the index for the Q terms is simply increased by one.

6.2. Noe recursion formula and Mathematica function variables

Table 5 contains the relationship between the variables in the recursion formula above and the variables in the Mathematica function `NoeKS2SidedRTProbRational` contained in Section 4 of the `KS2SidedOneSampleRational.nb` file. The function `NoeKS2SidedRTProbRational` has four arguments `[tIn_, internalPrecision_, sampleSizeIn_, intermediateOutput_]`. Internal variable `tRational` is set equal to the rational number value of `tIn` and `sampleSize` is set equal to the truncated value of `sampleSizeIn` so that the input values `tIn` and `sampleSizeIn` are not changed by the function.

Recursion function variable	Mathematica function variable	Recursion function variable	Mathematica function variable
t	`tRational`	$F(\gamma_m)$	`FgammaSUBmVector[[mIndex]]`
n	`sampleSize`	$h(m+1)-1$	`hSUBmPlusOneMinusOneVector[[mIndex]]`
i	`iIndex`	$g(m-1)$	`gSUBmMinusOneVector[[mIndex]]`
m	`mIndex`	p_m	`pSUBmVector[[mIndex]]`
k	`kIndex`	$Q_i(m-1)$	`oldQvector[[iIndex]]`
$2n$	`twiceSampleSize`	$Q_i(m)$	`newQvector[[iIndex]]`
		$P\left(D_n \geq \frac{t}{n}\right)$	`rightTailProbability`

Table 5: Noe recursion formula and Mathematica function variables.

7. The Durbin recursion formula

Durbin (1968, formulae 23 and 24) developed a difference equation and initial conditions from which the recursion formula below can be derived. Durbin's result is a generalization of the difference equations from Massey (1950) and the generating function approach of Kemperman (1961).

$$
\begin{aligned}
Q_0(t) &= 1 \\
Q_i(t) &= \frac{i^i}{i!} && \text{for } 1 \leq i \leq \lfloor t \rfloor \\
Q_i(t) &= \frac{i^i}{i!} - 2t \sum_{j=0}^{\lfloor i-t \rfloor} \frac{(t+j)^{j-1}}{j!} \frac{(i-t-j)^{i-j}}{(i-j)!} && \text{for } \lfloor t+1 \rfloor \leq i \leq \lfloor 2t \rfloor \\
Q_i(t) &= -\sum_{j=1}^{\lfloor 2t \rfloor} (-1)^j \left[\frac{(2t-j)^j}{j!} \right] Q_{i-j}(t) && \text{for } \lfloor 2t \rfloor + 1 \leq i \leq n \\
P\left(D_n \geq \frac{t}{n}\right) &= 1 - \frac{n!}{n^n} Q_n(t)
\end{aligned}
$$

Similar to Tingey, Durbin's recursion formula above reduces to Equation (1) for $1/2 < t < 1$. The Durbin recursion formula can also produce all the probabilities, $P\left(D_m \geq t/m\right) = 1 - m!Q_m(t)/m^m$, for every $m = 1, 2, \cdots, n$ where $m \geq t$.

7.1. The Durbin recursion formula for small test statistic values

For all sample sizes n and region $1/2 < t < 1$, the following derivation shows that the Durbin recursion formula reduces to the exact formula in Section 2.

$$\lfloor t \rfloor = 0 \qquad \lfloor t+1 \rfloor = 1 \qquad \lfloor 2t \rfloor = 1$$
$$Q_0(t) = 1 \qquad Q_1(t) = 2t-1 \qquad Q_2(t) = (2t-1)^2 \qquad Q_3(t) = (2t-1)^3$$
$$Q_n(t) = (2t-1)^n \qquad P\left(D_n \geq \frac{t}{n}\right) = 1 - \frac{n!}{n^n} Q_n(t) = 1 - \frac{n!}{n^n}(2t-1)^n$$

7.2. The Durbin recursion formula for a sample size of three

For sample size $n = 3$ and region $1 < t \leq 3/2$, the following derivation shows that the Durbin recursion formula reduces to the exact formula in Section 2.

$$\lfloor t \rfloor = 1 \qquad \lfloor t+1 \rfloor = 2 \qquad \lfloor 2t \rfloor = 2 \qquad \lfloor 2-t \rfloor = 0$$
$$Q_0(t) = 1 \qquad Q_1(t) = 1 \qquad Q_2(t) = -t^2 + 4t - 2 \qquad Q_3(t) = -2t^3 + 7t^2 - 4t$$
$$P\left(D_3 \geq \frac{t}{3}\right) = 1 - \frac{2}{9} Q_3(t) = \frac{4}{9}t^3 - \frac{14}{9}t^2 + \frac{8}{9}t + 1$$

For sample size $n = 3$ and region $3/2 \leq t < 2$, the following derivation shows that the Durbin recursion formula reduces to the exact formula in Section 2.

$$\lfloor t \rfloor = 1 \qquad \lfloor t+1 \rfloor = 2 \qquad \lfloor 2t \rfloor = 3 \qquad \lfloor 2-t \rfloor = 0$$
$$Q_0(t) = 1 \qquad Q_1(t) = 1 \qquad Q_2(t) = -t^2 + 4t - 2 \qquad Q_3(t) = -\frac{2}{3}t^3 + t^2 + 5t - \frac{9}{2}$$
$$P\left(D_3 \geq \frac{t}{3}\right) = 1 - \frac{2}{9} Q_3(t) = \frac{4}{27}t^3 - \frac{2}{9}t^2 - \frac{10}{9}t + 2$$

For sample size $n = 3$ and region $2 \leq t < 3$, the following derivation shows that the Durbin recursion formula reduces to the exact formula in Section 2.

$$\lfloor t \rfloor = 2 \qquad \lfloor t+1 \rfloor = 3 \qquad 4 \leq 2t < 6 \qquad \lfloor 3-t \rfloor = 0 \text{ for } 2 < t < 3$$
$$\lfloor 3-t \rfloor = 1 \text{ for } t = 2 \qquad Q_0(t) = 1 \qquad Q_1(t) = 1 \qquad Q_2(t) = 2$$
$$Q_3(t) = \frac{9}{2} - \frac{1}{3}(3-t)^3 \text{ for } 2 < t < 3 \qquad Q_3(2) = \frac{25}{6}$$
$$\text{Since } \frac{9}{2} - \frac{1}{3}(3-t)^3 = \frac{25}{6} \text{ for } t = 2, \text{ then } Q_3(t) = \frac{9}{2} - \frac{1}{3}(3-t)^3 \text{ for } 2 \leq t < 3$$
$$P\left(D_3 \geq \frac{t}{3}\right) = 1 - \frac{2}{9} Q_3(t) = \frac{2}{27}(3-t)^3$$

7.3. Durbin recursion formula and Mathematica function variables

Table 6 contains the relationship between the variables in the recursion formula above and the variables in the Mathematica function `DurbinKS2SidedRTProbRational` contained in Section 5 of the `KS2SidedOneSampleRational.nb` file. The function

Recursion function variable	Mathematica function variable	Recursion function variable	Mathematica function variable
t	`t`	$\lfloor 2t \rfloor$	`Trunc2t`
n	`sampleSize`	$\lfloor 2t \rfloor + 1$	`Trunc2tPlusOne`
i	`i`	$i!$	`FACT[[i]]`
j	`j`	$\lfloor i - t \rfloor$	`iMINUStTrunc`
$2t$	`tTwice`	$(-1)^j$	`SIGN`
$\lfloor t \rfloor$	`tTrunc`	$Q_i(t)$	`Q[[i]]`
$\lfloor t + 1 \rfloor$	`tTruncPlusOne`	$P\left(D_n \geq \frac{t}{n}\right)$	`rightTailProbability`

Table 6: Durbin recursion formula and Mathematica function variables.

`DurbinKS2SidedRTProbRational` has four arguments `[tIn_, internalPrecision_, sampleSizeIn_, intermediateOutput_]`. `t` is set equal to the rational number value of `tIn` and `sampleSize` is set equal to the truncated value of `sampleSizeIn` so that the input values `tIn` and `sampleSizeIn` are not changed by the function.

8. The Pomeranz recursion formula

The recursion formula below is derived from the recursion formula by Pomeranz (1973). In addition to some small changes, the indices are changed by adding one to eliminate the zero index that cannot be programmed in Mathematica. Note that by definition $0^0 = 1$.

$$
\begin{aligned}
a_1 &= 0 \\
a_2 &= \left(\frac{1}{n}\right) \min\{t - \lfloor t \rfloor, \lceil t \rceil - t\} \\
a_3 &= \left(\frac{1}{n}\right) - a_2 \\
a_i &= a_{i-2} + \left(\frac{1}{n}\right) && \text{for } i = 4, 5, \ldots, 2n+1 \\
a_{2n+2} &= 1 \\
v(1,1) &= 1 \\
v(1,j) &= 0 && \text{for } j = 2, 3, \ldots, n+1 \\
v(i,j) &= 0 && \text{for } j = 1, 2, \ldots, \lfloor n a_i - t \rfloor + 1; i = 2, 3, \ldots, 2n+2
\end{aligned}
$$

$$v(i,j) = \sum_{k=1}^{j} \frac{v(i-1,k)[a_i - a_{i-1}]^{j-k}}{(j-k)!} \quad \text{for} \begin{cases} j = \lfloor na_i - t \rfloor + 2, \lfloor na_i - t \rfloor + 3, \\ \qquad \ldots, \lceil na_i + t \rceil \\ i = 2, 3, \ldots, 2n+2 \end{cases}$$

$$v(i,j) = 0 \quad \text{for} \begin{cases} j = \lceil na_i + t \rceil + 1, \lceil na_i + t \rceil + 2, \\ \qquad \ldots, n+1 \\ i = 2, 3, \ldots, 2n+2 \end{cases}$$

$$P\left(D_n \geq \frac{t}{n}\right) = 1 - n! v(2n+2, n+1)$$

Considering all the $v(i,j)$ for a particular i, there are at most $\lceil na_i + t \rceil - (\lfloor na_i - t \rfloor + 2) + 1 \leq \lceil 2t - 1 \rceil$ non-zero $v(i,j)$ values. Just storing the non-zero value of $v(i,j)$ yields the following recursion formula.

$$\begin{aligned} a_1 &= 0 \\ a_2 &= \left(\frac{1}{n}\right) \min\{t - \lfloor t \rfloor, \lceil t \rceil - t\} \\ a_3 &= \left(\frac{1}{n}\right) - a_2 \\ a_i &= a_{i-2} + \left(\frac{1}{n}\right) && \text{for } i = 4, 5, \ldots, 2n+1 \\ a_{2n+2} &= 1 \\ v(1,1) &= 1 \\ v(1,j) &= 0 && \text{for } j = 2, 3, \ldots, n+1 \end{aligned}$$

$$v(i,j) = \sum_{k=\max\{1, \lfloor na_{i-1} - t \rfloor + 2\}}^{\min\{j, \lceil na_{i-1} + t \rceil\}} \frac{v(i-1,k)[a_i - a_{i-1}]^{j-k}}{(j-k)!} \quad \text{for} \begin{cases} j = \lfloor na_i - t \rfloor + 2, \\ \qquad \ldots, \lceil na_i + t \rceil \\ i = 2, 3, \ldots, 2n+2 \end{cases}$$

$$P\left(D_n \geq \frac{t}{n}\right) = 1 - n! v(2n+2, n+1)$$

Pomeranz (1973) also observed that the differences $a_i - a_{i-1}$ take on at most four distinct values over all i: 0, a_2, $2a_2$, or $1/n - 2a_2$. This means that the term $[a_i - a_{i-1}]^{j-k}/(j-k)!$ is repetitious and can be calculated separately to save time.

Like the Noe recursion formula, the Pomeranz recursion formula has no negative terms.

8.1. Pomeranz recursion formula and Mathematica function variables

Table 7 contains the relationship between the variables in the recursion formula above and the variables in the Mathematica function `PomeranzKS2SidedRTProbRational` contained in Section 6 of the `KS2SidedOneSampleRational.nb` file. This function has four arguments `[tIn_, internalPrecision_, sampleSizeIn_, intermediateOutput_]`. `t` is set equal to the rational number value of `tIn` and `sampleSize` is set equal to the truncated

value of `sampleSizeIn` so that the input values `tIn` and `sampleSizeIn` are not changed by the function. `numberValuesaiMinusaiMinusOne` represents the number of unique values of $a_i - a_{i-1}$ which cannot exceed 4. The unique values are contained in the vector `differentValuesaiMinusaiMinusOneVector` indexed by `differentValuesIndex`. For a particular $a_i - a_{i-1}$, its position in `differentValuesaiMinusaiMinusOneVector` is set equal to `valueIndex` so $a_i - a_{i-1}$ = `differentValuesaiMinusaiMinusOneVector[[valueIndex]]`. To calculate $v(i, j)$, each term $v(i-1, k)[a_i - a_{i-1}]^{j-k}/(j-k)!$ inside the summation must be evaluated. To facilitate the computation of each term, $[a_i - a_{i-1}]^j/(j)!$ is calculated for all possible values and put in `jMinuskTermMatrix[[valueIndex,j+1]]`.

For i from 2 through $2n+2$, the algorithm proceeds by calculating the vector $v(i, \cdot)$ for each i using the vector $v(i-1, \cdot)$. Thus, only two vectors need to be stored, $v(i-1, \cdot)$ and $v(i, \cdot)$. Since there are at most $\lceil 2t-1 \rceil$ = `twoTminusOneCeiling` non-zero values in both $v(i-1, \cdot)$ and $v(i, \cdot)$, the two vectors `vOldVector[[]]` and `vNewVector[[]]` each with $\lceil 2t-1 \rceil$ elements are used to store the non-zero parts of $v(i-1, \cdot)$ and $v(i, \cdot)$ respectively. Specifically, the first value `vOldVector[[1]]` stores the value of $v(i-1, j)$ for j = `vOldBeginIndex`. In general, the value of $v(i-1, j)$ is stored in `vOldVector[[j-vOldBeginIndexMinusOne]]`. Similarly, the value of $v(i, j)$ is stored in `vNewVector[[j-vNewBeginIndexMinusOne]]`.

9. The Durbin matrix formula

Durbin (1973, formulae 2.4.3 and 2.4.4 on pages 10–11) develops a formula (DurbinMatrix) that constructs a matrix H and then raises it to the power n, H^n. Let $k = \lceil t \rceil$, $\delta = k - t$, $p = 2k - 1$, and

$$f(\delta, p) = \begin{cases} 0 & \text{if } \delta < 1/2 \\ (2\delta - 1)^p & \text{if } \delta \geq 1/2 \end{cases}$$

Compute H^n using the matrix H with p rows and p columns shown below.

$$H = \begin{bmatrix} 1-\delta & 1 & 0 & 0 & 0 & \cdots & 0 \\ (1-\delta^2)/2! & 1/1! & 1 & 0 & 0 & \cdots & 0 \\ (1-\delta^3)/3! & 1/2! & 1/1! & 1 & 0 & \cdots & 0 \\ (1-\delta^4)/4! & 1/3! & 1/2! & 1/1! & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \dfrac{1-\delta^{p-1}}{(p-1)!} & \dfrac{1}{(p-1)!} & \dfrac{1}{(p-2)!} & \dfrac{1}{(p-3)!} & \dfrac{1}{(p-4)!} & \cdots & 1 \\ \dfrac{1-\delta^p + f(\delta,p)}{p!} & \dfrac{1-\delta^{p-1}}{(p-1)!} & \dfrac{1-\delta^{p-2}}{(p-2)!} & \dfrac{1-\delta^{p-3}}{(p-3)!} & \dfrac{1-\delta^{p-4}}{(p-4)!} & \cdots & 1-\delta \end{bmatrix}.$$

If $H^n(k, k)$ represents the (k, k)th element in H^n, then

$$P\left(D_n \geq \frac{t}{n}\right) = 1 - \frac{n!}{n^n} H^n(k, k).$$

The Durbin matrix formula can also produce all the probabilities, $P(D_m \geq t/m) = 1 - m! H^m(k, k)/m^m$, for every $m = 1, 2, \cdots, n$ where $m \geq t$. Using the formula above,

Recursion function variable	Mathematica function variable	Recursion function variable	Mathematica function variable
t	`t`	a_i	`aVector[[i]]`
n	`sampleSize`	$v(i-1,j)$	`vOldVector[[j]]`
i	`i`	$v(i,j)$	`vNewVector[[j]]`
j	`j`	$n+1$	`sampleSizePlusOne`
k	`k`	$2n+1$	`twoSampleSizePlusOne`
$\lfloor t \rfloor$	`tFloor`	$2n+2$	`twoSampleSizePlusTwo`
$\lceil t \rceil$	`tCeiling`	$\frac{1}{n}$	`oneOverSampleSize`
$i!$	`FACT[[i+1]]`	$\lceil 2t-1 \rceil$	`twoTminusOneCeiling`
$\max\{1, \lfloor na_{i-1} - t \rfloor + 2\}$	`vOldBeginIndex`		
$\min\{n+1, \lceil na_{i-1} + t \rceil\}$	`vOldEndIndex`		
$\max\{1, \lfloor na_i - t \rfloor + 2\}$	`vNewBeginIndex`		
$\min\{n+1, \lceil na_i + t \rceil\}$	`vNewEndIndex`		
$\max\{1, \lfloor na_{i-1} - t \rfloor + 2\} - 1$	`vOldBeginIndexMinusOne`		
$\max\{1, \lfloor na_i - t \rfloor + 2\} - 1$	`vNewBeginIndexMinusOne`		
$\max\{1, \lfloor na_{i-1} - t \rfloor + 2\}$	`kLowBound`		
$\min\{j, \lceil na_{i-1} + t \rceil\}$	`kHighBound`		
$a_i - a_{i-1}$	`aiMinusaiMinusOneVector[[i]]`		
$a_i - a_{i-1}$	`differentValuesaiMinusaiMinusOneVector[[valueIndex]]`		
0	`differentValuesaiMinusaiMinusOneVector[[1]]`		
a_2	`differentValuesaiMinusaiMinusOneVector[[2]]`		
$2a_2$	`differentValuesaiMinusaiMinusOneVector[[3]]`		
$\frac{1}{n} - 2a_2$	`differentValuesaiMinusaiMinusOneVector[[4]]`		
$\frac{[a_i - a_{i-1}]^{j-k}}{(j-k)!}$	`jMinuskTermMatrix[[valueIndex,j-k+1]]`		
$P\left(D_n \geq \frac{t}{n}\right)$	`rightTailProbability`		

Table 7: Pomeranz recursion formula and Mathematica function variables.

Marsaglia, Tsang, and Wang (2003) developed a double precision C program that for sample size $n = 16,000$ and test statistic $t = 171.9008$ ($d = t/n = 0.0107438$) computes $P(D_{16,000} \leq 0.0107438) = 0.95060023909487545$. Since the actual value is $P(D_{16,000} \leq 0.0107438) =$

$0.95060023909504600 63\cdots$, the C program has only 11 digits of accuracy not the 13 to 15 digits claimed by the authors.

The result of the Durbin matrix formula for sample size $n = 3$ and region $2 \le t < 3$ will not be derived in the following subsections because of the complexity of the formula. For example, matrix H has five rows and five columns while $f(\delta, p)$ has two values: $f(\delta, p) = 0$ for $\frac{9}{4} < t < 3$ and $f(\delta, p) = (5 - 2t)^5$ for $2 \le t \le \frac{9}{4}$.

9.1. The Durbin matrix recursion formula for small test statistic values

For all sample sizes n and region $1/2 < t < 1$, the following derivation shows that the Durbin matrix recursion formula reduces to the exact formula in Section 2.

$$k = \lceil t \rceil = 1 \qquad \delta = k - t = 1 - t \qquad p = 2k - 1 = 1 \qquad f(\delta, p) = 0 \qquad 1 - 2\delta = 2t - 1$$

$$H = [1 - 2\delta] = [2t - 1] \qquad H^2 = \left[(2t-1)^2\right] \qquad H^3 = \left[(1-2\delta)^3\right] = \left[(2t-1)^3\right]$$

$$H^n = [(1-2\delta)^n] = [(2t-1)^n] \qquad P\left(D_n \ge \frac{t}{n}\right) = 1 - \frac{n!}{n^n} H^n(1,1) = 1 - \frac{n!}{n^n}(2t-1)^n$$

9.2. Diagonal elements in matrix H for a sample size of three

For sample size $n = 3$ and $p = 3$, define A, B, and C below.

$$A = 1 - \delta, \qquad B = \frac{1}{2}\left(1 - \delta^2\right), \qquad C = \frac{1 - 2\delta^p + f(\delta, p)}{p!}.$$

Then the matrix H for $n = 3$ and $p = 3$ is

$$H = \begin{bmatrix} A & 1 & 0 \\ B & 1 & 1 \\ C & B & A \end{bmatrix}$$

and the diagonal elements for H^3 are

$$H^3(1,1) = A^3 + 2AB + B + C$$
$$H^3(2,2) = 2AB + 4B + C + 1$$
$$H^3(3,3) = A^3 + 2AB + B + C$$

9.3. The Durbin matrix formula for a sample size of three

For sample size $n = 3$ and region $1 < t \le 3/2$, the following derivation using the formulae in Section 9.2 shows that the Durbin matrix formula reduces to the exact formula in Section 2.

$$k = \lceil t \rceil = 2 \qquad \delta = 2 - t \qquad p = 2k - 1 = 3 \qquad f(\delta, p) = (2\delta - 1)^p = (3 - 2t)^3$$
$$A = t - 1 \qquad B = \frac{1}{2} - \frac{1}{2}(2 - t)^2 = -\frac{1}{2}t^3 + 2t - \frac{3}{2}$$
$$C = \frac{1}{6} - \frac{1}{3}(2 - t)^3 + \frac{1}{6}(3 - 2t)^3 = -(t - 2)(t - 1)^2$$
$$H^3(2, 2) = 2AB + 4B + C + 1 = -2t^3 + 7t^2 - 4t$$
$$P\left(D_3 \geq \frac{t}{3}\right) = 1 - \frac{2}{9}H^3(2, 2) = \frac{4}{9}t^3 - \frac{14}{9}t^2 + \frac{8}{9}t + 1$$

For sample size $n = 3$ and region $3/2 \leq t < 2$, the following derivation using the formulae in Section 9.2 shows that the Durbin matrix formula reduces to the exact formula in Section 2.

$$k = \lceil t \rceil = 2 \qquad \delta = 2 - t \qquad p = 2k - 1 = 3 \qquad f(\delta, p) = 0$$
$$A = t - 1 \qquad B = \frac{1}{2} - \frac{1}{2}(2 - t)^2 = -\frac{1}{2}t^3 + 2t - \frac{3}{2}$$
$$C = \frac{1}{6} - \frac{1}{3}(2 - t)^3 + \frac{1}{6}(3 - 2t)^3 = -(t - 2)(t - 1)^2$$
$$H^3(2, 2) = 2AB + 4B + C + 1 = -\frac{2}{3}t^3 + t^2 + 5t - \frac{9}{2}$$
$$P\left(D_3 \geq \frac{t}{3}\right) = 1 - \frac{2}{9}H^3(2, 2) = \frac{4}{27}t^3 - \frac{2}{9}t^2 - \frac{10}{9}t + 2$$

9.4. Durbin matrix recursion formula and Mathematica function variables

Table 8 contains the relationship between the variables in the recursion formula above and the variables in the Mathematica function `DurbinMatrixKS2SidedRTProbRational` contained in Section 7 of the `KS2SidedOneSampleRational.nb` file. The function `DurbinMatrixKS2SidedRTProbRational` has four arguments `[tIn_, internalPrecision_, sampleSizeIn_, intermediateOutput_]`. `t` is set equal to the rational number value of `tIn` and `sampleSize` is set equal to the truncated value of `sampleSizeIn` so that the input values `tIn` and `sampleSizeIn` are not changed by the function.

10. Implementation considerations

Sections 2 through 9 developed eight different methods to calculate the two-sided one sample K-S p value $P(D_n \geq d = t/n)$. As noted earlier, the six recursion formulae (Kolmogorov, Tingey, Epanechnikov, Noe, Durbin, Pomeranz) and the matrix formula (DurbinMatrix) unlike the Exact formula can be used for sample sizes exceeding $n = 31$. Table 9 contains a summary of the Mathematica function name implementing each formula, the name of the file containing the function, and its location in the file.

Each Mathematica function listed in Table 9 has a built-in intermediate output option that is controlled by the value of the input variable `intermediateOutput` which can be used to show how the function works and if necessary to facilitate debugging. No intermediate output is

Recursion function variable	Mathematica function variable	Recursion function variable	Mathematica function variable
t	`t`	$\delta = k - t$	`delta`
n	`sampleSize`	$\frac{1}{i!}$	`FACTinverse[[i]]`
$k = \lceil t \rceil$	`tCeil`	$H(i, j)$	`H[[i,j]]`
$p = 2k - 1$	`TwoCeilingTminusOne`	$H^n(i, j)$	`HpowerSampleSize[[i,j]]`
		$P\left(D_n \geq \frac{t}{n}\right)$	`rightTailProbability`

Table 8: Durbin matrix recursion formula and Mathematica function variables.

produced for `intermediateOutput` ≤ 0 while increasing amounts of intermediate output are produced for larger values of `intermediateOutput`. The maximum amount of intermediate output is produced for `intermediateOutput` ≥ 4.

Since all numbers in the Mathematica function for each formula are rational numbers, the intermediate output gives two forms of most numbers: the internal rational number and a decimal form of the number with `internalPrecision` significant decimal digits. Giving the decimal equivalent of a rational number makes the intermediate output easier to interpret.

As an accuracy check for the eight different Mathematica functions (Exact, Kolmogorov, Tingey, Epanechnikov, Noe, Durbin, Pomeranz, and DurbinMatrix), a Mathematica program (function `KS2SidedOneSampleFormulaeAccuracyCheck` contained in Section 8 of the `KS2SidedOneSampleRational.nb` file) was written to test whether they all get the same result. Because the Kolmogorov recursion formula is only applicable for integer t, it is not run for non-integer t. For every test statistic t and sample size n tried, the right tail probability was calculated for each of the eight Mathematica functions and the results compared. An error message was written if any right tail probability was different from any other right tail probability. The program was run on all test statistics t from 1 to n using an increment of 0.1 and for all sample sizes from $n = 2$ to $n = 31$. Since no errors were found, we are certain that all eight Mathematica functions are correct.

The Noe, Pomeranz, and Durbin matrix formulae are the only formulae that contain all non-negative terms and therefore are not subject to catastrophic cancelation. Using intermediate results, the Kolmogorov, Tingey, Durbin, and Durbin matrix formulae can produce all the probabilities, $P(D_k \geq t/k)$ for every $k = 1, 2, \cdots, n$ when $t < k$.

11. Timing the seven formulae

In order to determine which of the seven formulae (the six recursion formulae and the matrix formula) is the fastest, a Mathematica function `SevenFormulaeTiming` contained in Section 9 of the `KS2SidedOneSampleRational.nb` file was developed to time all seven formula and

Formula name	Mathematica function name Contained in file name	Listed in file section
Exact	`ExactKS2SidedRTProbs` `ExactKS2SidedOneSample.nb`	1
Kolmogorov	`KolmogorovKS2SidedRTProbsRational` `KS2SidedOneSampleRational.nb`	1
Tingey	`TingeyKS2SidedRTProbsRational` `KS2SidedOneSampleRational.nb`	2
Epanechnikov	`EpanechnikovKS2SidedRTProbRational` `KS2SidedOneSampleRational.nb`	3
Noe	`NoeKS2SidedRTProbRational` `KS2SidedOneSampleRational.nb`	4
Durbin	`DurbinKS2SidedRTProbRational` `KS2SidedOneSampleRational.nb`	5
Pomeranz	`PomeranzKS2SidedRTProbRational` `KS2SidedOneSampleRational.nb`	6
DurbinMatrix	`DurbinMatrixKS2SidedRTProbRational` `KS2SidedOneSampleRational.nb`	7

Table 9: Mathematica function name and file name for the eight formulae.

create a spreadsheet file of the results that can be opened in Excel. For every sample size n and test statistic t, the p value is calculated for each of the seven formulae and the computer time needed by each formula is recorded. In addition, the seven p values are compared to make sure they are all equal. If some are not equal, an error message is written.

Tables 10 and 11 contain computational experience for the seven formulae (six recursion formulae and DurbinMatrix) where all timings were done on a Pentium IV running at 2.4 GHz. Because the Kolmogorov recursion formula is only valid for integer t, the entries in these tables for Kolmogorov are blank for non-integer test statistics t. Although Tingey's recursion formula reduces mathematically to the Kolmogorov recursion formula for integer t, the tables show that the Kolmogorov recursion formula is always slightly faster than Tingey's. This difference in speed is due to the extra overhead incurred by the more general Tingey recursion formula. The seven formulae, ranging from the fastest to the slowest computational times for values of t from 30 to 40, are Durbin, Epanechnikov, Kolmogorov, Tingey, Durbin matrix, Pomeranz, and Noe. The computation time for every formula increases as both the test statistic t and sample size n increase. As the test statistic t increases, the computation time for the Durbin matrix formula increases much faster than the six recursion formulae. Although the Durbin matrix formula is slightly faster than the Durbin recursion formula for $n = 500$ and very small values of t, the Durbin recursion formula is much faster in all other situations. Since it is almost always the fastest formula, the rest of the paper uses the Durbin recursion formula exclusively.

Sample size n	Recursion formula	Time in seconds to compute $P(D_n \geq t/n)$ for $t =$ 4.9	5	9.9	10	14.9	15	19.9	20
50	Kolmogorov		.078		.328		.703		1.250
	Tingey	.093	.110	.422	.406	.922	.875	1.562	1.516
	Epanechnikov	.047	.047	.078	.079	.093	.078	.110	.094
	Noe	.218	.172	.813	.625	1.609	1.234	2.250	1.797
	Durbin	.016	.016	.031	.031	.063	.031	.047	.031
	Pomeranz	.188	.125	.687	.407	1.437	.828	2.328	1.281
	DurbinMatrix	.047	.031	.532	.312	1.688	1.062	3.688	2.500
100	Kolmogorov		.188		.781		1.813		3.328
	Tingey	.250	.266	1.062	.985	2.437	2.172	4.391	3.906
	Epanechnikov	.188	.187	.250	.235	.328	.281	.406	.344
	Noe	.578	.437	2.438	1.718	5.453	3.735	9.234	6.375
	Durbin	.047	.032	.109	.078	.156	.110	.203	.140
	Pomeranz	.469	.297	2.000	1.078	4.484	2.344	7.922	4.062
	DurbinMatrix	.093	.063	.969	.578	3.484	2.188	8.250	5.312
200	Kolmogorov		.422		1.969		4.734		8.875
	Tingey	.625	.594	2.860	2.390	6.766	5.531	12.500	10.156
	Epanechnikov	.703	.672	.953	.828	1.219	1.000	1.469	1.218
	Noe	1.719	1.156	7.735	4.672	18.484	10.891	33.203	19.578
	Durbin	.125	.078	.282	.187	.484	.313	.688	.468
	Pomeranz	1.406	.719	6.390	3.094	15.219	7.109	28.391	13.062
	DurbinMatrix	.172	.094	1.922	1.094	7.218	4.313	17.844	10.953
300	Kolmogorov		.719		3.609		8.938		16.937
	Tingey	1.157	.984	5.453	4.266	13.343	10.188	25.219	19.015
	Epanechnikov	1.610	1.500	2.062	1.828	2.641	2.218	3.235	2.640
	Noe	3.829	2.266	17.234	9.469	41.047	22.109	74.563	40.125
	Durbin	.203	.141	.547	.359	.937	.610	1.359	.906
	Pomeranz	2.828	1.328	13.704	6.078	33.609	14.375	62.328	26.469
	DurbinMatrix	.266	.156	3.281	1.875	12.282	7.218	30.782	18.625
400	Kolmogorov		1.094		5.672		14.250		27.281
	Tingey	1.844	1.453	8.953	6.563	22.219	15.953	41.797	30.063
	Epanechnikov	2.875	2.656	3.703	3.234	4.688	3.891	5.734	4.641
	Noe	6.171	3.391	27.969	14.219	66.796	33.625	122.547	61.187
	Durbin	.328	.188	.890	.563	1.578	.984	2.282	1.437
	Pomeranz	4.828	2.094	24.250	10.047	60.250	24.187	112.547	45.203
	DurbinMatrix	.343	.157	4.125	2.281	15.984	9.063	40.578	23.859
500	Kolmogorov		1.484		8.141		20.781		40.344
	Tingey	2.687	1.938	13.453	9.281	33.750	22.938	64.000	43.796
	Epanechnikov	4.516	4.140	5.875	5.094	7.516	6.203	9.203	7.406
	Noe	11.422	5.766	52.218	24.813	123.312	59.094	224.766	106.156
	Durbin	.484	.266	1.344	.812	2.391	1.437	3.485	2.156
	Pomeranz	7.672	3.156	39.391	15.656	98.875	38.203	186.219	72.375
	DurbinMatrix	.454	.203	5.485	2.968	21.672	12.157	56.000	32.328

Table 10: Computational times for all formulae using rational arithmetic, I. (All timings on a Pentium IV running at 2.4 GHz.)

Sample size n	Recursion formula	Time in seconds to compute $P(D_n \geq t/n)$ for $t =$ 24.9	25	29.9	30	34.9	35	39.9	40
50	Kolmogorov		1.906		2.688		3.547		4.469
	Tingey	2.422	2.265	3.360	3.156	4.359	4.094	5.453	5.141
	Epanechnikov	.125	.093	.110	.093	.110	.094	.093	.078
	Noe	2.485	2.172	2.375	2.109	2.187	1.938	1.953	1.734
	Durbin	.031	.032	.031	.031	.016	.015	.016	.000
	Pomeranz	3.172	1.734	3.938	2.156	4.578	2.484	5.000	2.703
	DurbinMatrix	6.812	4.766	11.234	8.156	17.375	12.938	25.672	19.953
100	Kolmogorov		5.312		7.672		10.500		13.844
	Tingey	6.891	6.125	9.984	8.797	13.687	11.969	17.719	15.687
	Epanechnikov	.469	.390	.532	.453	.594	.468	.641	.500
	Noe	13.485	9.390	17.719	12.641	21.578	15.843	24.594	18.781
	Durbin	.250	.172	.266	.187	.266	.188	.265	.203
	Pomeranz	12.094	6.141	16.796	8.360	21.890	10.797	27.032	13.296
	DurbinMatrix	15.875	10.531	26.907	18.000	41.656	28.297	60.687	42.657
200	Kolmogorov		15.000		22.734		30.641		41.484
	Tingey	20.235	16.515	29.938	24.234	41.719	33.625	55.500	45.109
	Epanechnikov	1.781	1.438	2.078	1.672	2.375	1.891	2.719	2.125
	Noe	51.125	30.344	72.609	43.484	96.641	58.328	122.859	74.969
	Durbin	.891	.625	1.125	.750	1.297	.891	1.484	1.031
	Pomeranz	44.844	20.375	65.484	29.188	89.156	39.656	116.204	51.718
	DurbinMatrix	36.234	22.938	63.078	40.031	100.703	64.313	150.453	98.062
300	Kolmogorov		28.469		42.625		60.406		81.875
	Tingey	41.125	31.532	61.328	46.922	86.500	65.390	115.953	88.625
	Epanechnikov	3.907	3.156	4.594	3.640	5.328	4.157	6.078	4.734
	Noe	117.625	64.532	169.203	91.562	229.328	125.157	296.343	163.391
	Durbin	1.813	1.203	2.281	1.516	2.750	1.812	3.219	2.156
	Pomeranz	102.078	43.063	150.172	62.531	208.391	86.406	276.344	115.359
	DurbinMatrix	63.015	39.969	112.188	71.171	181.391	116.047	276.812	178.250
400	Kolmogorov		46.500		70.187		99.485		136.547
	Tingey	69.656	50.531	104.485	75.468	147	106.641	199.375	145.468
	Epanechnikov	6.953	5.547	8.203	6.375	9.484	7.344	10.859	8.344
	Noe	195.094	97.391	284.312	142.657	390.640	196.735	511.406	259.594
	Durbin	3.094	1.969	3.890	2.485	4.703	3.062	5.578	3.641
	Pomeranz	185.703	74.313	274.812	108.828	382.469	151.844	510.766	203.203
	DurbinMatrix	86.516	51.266	154.609	95.125	247.984	152.360	379.797	237.156
500	Kolmogorov		69.391		105.218		149.938		207.156
	Tingey	107.532	74.265	161.703	111.766	228.703	158.735	311.156	217.953
	Epanechnikov	11.188	8.859	13.219	10.312	15.375	11.844	17.703	13.547
	Noe	356.187	168.422	518.047	246.485	708.625	339.296	927.094	446.797
	Durbin	4.734	2.969	5.969	3.766	7.265	4.610	8.671	5.547
	Pomeranz	309.844	120.234	460.906	177.266	645.844	249.093	867.969	336.156
	DurbinMatrix	117.203	70.234	212.016	127.734	346.953	211.516	536.656	331.953

Table 11: Computational times for all formulae using rational arithmetic, II. (All timings on a Pentium IV running at 2.4 GHz.)

12. Calculating the two-sided half-width

In addition to calculating the p value for hypothesis testing, the two-sided one sample K-S cumulative sampling distribution can be used to construct a two-sided confidence band around the empirical distribution $F_n(x)$. The half-width of a two-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 half-width 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 half-width 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 half-width rounded to ρ significant digits for sample size n and confidence coefficient $1-\alpha$. Note that half-width $d(n, \alpha, \rho)$ is also the hypothesis testing critical value for an α level of significance.

Computationally finding the half-width (critical value) $d(n, \alpha, \rho)$ is a three step process: (1) use an approximation to find an initial value close to $d(n, \alpha, \rho)$, (2) use this initial value to find upper and lower bounds on $d(n, \alpha, \rho)$, and (3) use a search procedure to narrow the distance between the lower and upper bounds until both the lower and upper bounds are the same to ρ significant digits. The first step uses the K-S one-sided one sample approximation of Maag and Dicaire (1971) with a probability of $\alpha/2$ (a confidence coefficient of $1-\alpha/2$) to find the initial value. Specifically, the approximation $\alpha/2 \simeq \exp\left(-\left[6nd+1\right]^2/18n\right)$ is solved for d yielding $d \simeq \sqrt{-\ln(\alpha/2)/2n} - 1/6n$. The second step gradually increases the distance away from the initial value until both a lower and an upper bound on the actual value is found. Step three then uses linear interpolation (linear search) to find a new bound. The resulting linear search algorithm is shown below.

K-S two-sided half-width algorithm using linear search

Step 1 (Find initial half-width): Using the one-sided approximation, calculate the initial half-width $dApprox = \sqrt{-\ln(\alpha/2)/2n} - 1/6n$. Truncate the value of $dApprox$ to ρ significant digits and go to Step 2.

Step 2 (Determine if initial half-width is a lower or upper bound): Compute $p = P[D_n \geq dApprox]$. If $p > \alpha$, then $dApprox$ is a lower bound, set $d_L = dApprox$, set $p_L = p$, and go to Step 3. Otherwise, $dApprox$ is an upper bound, set $d_U = dApprox$, set $p_U = p$, and go to Step 6.

Step 3 (Determine an upper bound): Convert d_L to a numerator integer $dNum_L$ and a denominator integer $dDen_L$ where $dDen_L$ is a power of ten so that $d_L = dNum_L/dDen_L$. If the number of significant digits does not exceed six, $\rho \leq 6$, set increment $inc = 1/10$. Otherwise $\rho > 6$ and set the increment $inc = 10^{\rho-7}$. Go to Step 4.

Step 4 (Construct and test a possible upper bound): Set $inc = inc * 10$, set $dTryNum = dNum_L + inc$, and set $dTryDen = dDen_L$. If $dTryNum \geq dTryDen$,

set $p_U = 0$, set $dNum_U = dNum_L$, set $dDen_U = dNum_L$ which yields the upper bound $d_U = 1$, and then go to Step 9. Otherwise, $dTryNum < dTryDen$ and calculate $p = P[D_n \geq dTryNum/dTryDen]$. If $p > \alpha$, then a new lower bound has been found and go to Step 5. Otherwise, the initial upper bound has been found, set $dNum_U = dTryNum$, set $dDen_U = dTryDen$, set $p_U = p$, and go to Step 9.

Step 5 (New lower bound): Set $dNum_L = dTryNum$, $dDen_L = dTryDen$, and $p_L = p$. Go to Step 4.

Step 6 (Determine a lower bound): Convert d_U to a numerator integer $dNum_U$ and a denominator integer $dDen_U$ where $dDen_U$ is a power of ten so that $d_U = dNum_U/dDen_U$. If the number of significant digits does not exceed four, $\rho \leq 6$, set increment $inc = 1/10$. Otherwise $\rho > 6$ and set the increment $inc = 10^{\rho-7}$. Go to Step 7.

Step 7 (Construct and test a possible lower bound): Set $inc = inc * 10$, set $dTryNum = dNum_U - inc$, and set $dTryDen = dDen_U$. If $inc < 10^{\rho-1}$, set $dDen_U = dDen_U * 10$, set $dNum_U = dNum_U * 10$, set $dTryDen = dTryDen * 10$, set $dTryNum = dTryNum * 10$, and set $inc = inc * 10$. If $dTryNum \leq 0$, set $p_L = 1$, set $dNum_L = 0$, set $dDen_L = dTryDen$ which yields the lower bound $d_L = 0$, and then go to Step 9. Otherwise, $dTryNum > 0$ and calculate $p = P[D_n \geq dTryNum/dTryDen]$. If $p < \alpha$, then a new upper bound has been found and go to Step 8. Otherwise, the initial lower bound has been found, set $dNum_L = dTryNum$, set $dDen_L = dTryDen$, set $p_L = p$, and go to Step 9.

Step 8 (New upper bound): Set $dNum_U = dTryNum$, $dDen_U = dTryDen$, and $p_U = p$. Go to Step 7.

Step 9 (Linear search iteration): If $dNum_U - dNum_L \leq 1$, go to Step 12. Using linear interpolation, set $dTryNum = \lfloor dNum_L + (dNum_U - dNum_L) \times (p_L - \alpha)/(p_L - p_U) \rfloor$ and $dTryDen = dDen_U$. If $dTryNum \geq dNum_U$, set $dTryNum = dNum_U - 1$. If $dTryNum \leq dNum_L$, set $dTryNum = dNum_L + 1$. Calculate $p = P[D_n \geq dTryNum/dTryDen]$. If $p > \alpha$, then a new lower bound has been found and go to Step 10. Otherwise, a new upper bound has been found and go to Step 11.

Step 10 (Linear search new lower bound): Set $dNum_L = dTryNum$, $dDen_L = dTryDen$, $p_L = p$, and go to Step 9.

Step 11 (Linear search new upper bound): Set $dNum_U = dTryNum$, $dDen_U = dTryDen$, $p_U = p$, and go to Step 9.

Step 12 (Determine whether to use lower or upper bound): Calculate $p = P[D_n \geq (dNum_L \times 10 + 5)/(dDen_L \times 10)]$. If $p < \alpha$, then use the lower bound by setting $dNum_F = dNum_L$, $dDen_F = dDen_L$, and $d_F = dNum_L/dDen_L$, and then going to Step 13. Otherwise, use the upper bound by setting $dNum_F = dNum_U$, $dDen_F = dDen_U$, $d_F = dNum_U/dDen_U$, and then going to Step 13.

Step 13 (Half-width found): Terminate the algorithm with the half-width $d_F = dNum_F/dDen_F$.

Linear search algorithm variable	Mathematica function variable	Linear Search algorithm variable	Mathematica function variable
α	`alpha, alphaIn`	$d_L = \frac{dNum_L}{dDen_L}$	$\frac{\texttt{dLowLimitNumerator}}{\texttt{dLowLimitDenominator}}$
n	`sampleSize, sampleSizeIn`	$dNum_U$	`dHighLimitNumerator`
ρ	`numberDigitsPrecision`	$dDen_U$	`dHighLimitDenominator`
$dApprox$	`dOneSide`	$d_U = \frac{dNum_U}{dDen_U}$	$\frac{\texttt{dHighLimitNumerator}}{\texttt{dHighLimitDenominator}}$
p	`alphaOutput`	inc	`dNumeratorIncrement`
$\alpha/2$	`oneHalfAlpha`	$dTryNum$	`dTryNumerator`
p_L	`dLowLimitAlpha`	$dTryDen$	`dTryDenominator`
p_U	`dHighLimitAlpha`	$dNum_F$	`dNumeratorFinal`
$dNum_L$	`dLowLimitNumerator`	$dDen_F$	`dDenominatorFinal`
$dDen_L$	`dLowLimitDenominator`	d_F	`dFinal`

Table 12: Linear search algorithm and Mathematica function variables.

Table 12 contains the relationship between the variables in the linear search algorithm listed above and the Mathematica function `KS2SidedHalfWidthByLinearSearch` contained in Section 10 of the `KS2SidedOneSampleRational.nb` file.

The Mathematica function `KS2SidedHalfWidthsToFile` contained in Section 11 of the `KS2SidedOneSampleRational.nb` file finds half-widths using linear search with the Durbin formula and writes these half-widths 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 half-widths where every digit in a half-width is output separately so the half-width can be reconstructed to any desired accuracy. These text files will be used as input files to produce the computational experience in Section 14.

Tables 13 and 14 contain the half-widths 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 $n = 2$ through $n = 500$.

13. Test statistic complexity

The test statistics $d = t/n$ used in Tables 10 and 11 are simple rational numbers with at most six integer digits in both the numerator and denominator. A natural question to ask is how does increasing the number of digits in the test statistic's numerator and denominator effect the computation time.

Let the complexity of a rational number be measured by the number of digits in its numerator and denominator integers. To understand how the complexity of the test statistic d and the

Sample	Half-width $d(n, \alpha, \rho = 6)$					
size n	$\alpha = 0.2$	$\alpha = 0.1$	$\alpha = 0.05$	$\alpha = 0.02$	$\alpha = 0.01$	$\alpha = 0.001$
2	.683772	.776393	.841886	.900000	.929289	.977639
3	.564810	.636045	.707598	.784557	.829002	.920630
4	.492653	.565216	.623939	.688870	.734238	.850465
5	.446973	.509449	.563275	.627180	.668531	.781369
6	.410354	.467993	.519262	.577407	.616607	.724791
7	.381452	.436068	.483424	.538440	.575812	.679305
8	.358286	.409622	.454267	.506543	.541793	.640979
9	.339071	.387463	.430011	.479596	.513317	.608464
10	.322568	.368662	.409246	.456624	.488932	.580417
11	.308257	.352419	.391224	.436703	.467702	.555878
12	.295734	.338149	.375430	.419178	.449045	.534217
13	.284662	.325487	.361432	.403621	.432473	.514899
14	.274770	.314168	.348901	.389695	.417616	.497534
15	.265849	.303970	.337596	.377127	.404199	.481818
16	.257746	.294717	.327333	.365709	.392007	.467505
17	.250350	.286266	.317963	.355275	.380862	.454398
18	.243564	.278508	.309360	.345693	.370622	.442338
19	.237309	.271354	.301425	.336852	.361170	.431192
20	.231519	.264731	.294075	.328661	.352411	.420851
21	.226137	.258574	.287242	.321044	.344263	.411222
22	.221116	.252832	.280869	.313936	.336659	.402227
23	.216419	.247459	.274904	.307283	.329540	.393800
24	.212012	.242417	.269307	.301039	.322857	.385883
25	.207866	.237674	.264041	.295163	.316567	.378427
26	.203957	.233202	.259075	.289621	.310633	.371388
27	.200262	.228974	.254380	.284381	.305022	.364729
28	.196763	.224971	.249934	.279417	.299707	.358418
29	.193443	.221172	.245715	.274706	.294661	.352424
30	.190287	.217560	.241703	.270227	.289864	.346722
31	.187282	.214122	.237884	.265962	.285295	.341290
32	.184416	.210842	.234241	.261893	.280936	.336106
33	.181679	.207710	.230761	.258007	.276772	.331152
34	.179061	.204714	.227434	.254290	.272789	.326411
35	.176555	.201846	.224247	.250730	.268974	.321870
36	.174151	.199095	.221191	.247316	.265315	.317513
37	.171844	.196455	.218257	.244038	.261803	.313329
38	.169627	.193918	.215438	.240889	.258427	.309307
39	.167494	.191477	.212727	.237858	.255179	.305437
40	.165440	.189127	.210115	.234940	.252051	.301709
41	.163461	.186862	.207598	.232128	.249036	.298115
42	.161552	.184677	.205170	.229414	.246127	.294646
43	.159708	.182567	.202826	.226794	.243319	.291296
44	.157927	.180529	.200561	.224262	.240604	.288059
45	.156204	.178557	.198370	.221814	.237979	.284927
46	.154537	.176650	.196250	.219445	.235439	.281896
47	.152923	.174802	.194197	.217150	.232978	.278960
48	.151358	.173012	.192208	.214926	.230594	.276114
49	.149841	.171276	.190278	.212769	.228281	.273353
50	.148369	.169592	.188406	.210677	.226037	.270675

Table 13: Half-width $d(n, \alpha, \rho = 6)$ to six significant digits for $n = 2$ to $n = 50$.

Sample	Half-width $d(n, \alpha, \rho = 6)$					
size n	$\alpha = 0.2$	$\alpha = 0.1$	$\alpha = 0.05$	$\alpha = 0.02$	$\alpha = 0.01$	$\alpha = 0.001$
60	.135708	.155103	.172305	.192675	.206731	.247614
70	.125833	.143804	.159747	.178632	.191668	.229607
80	.117850	.134671	.149596	.167280	.179490	.215041
90	.111223	.127089	.141169	.157855	.169379	.202942
100	.105605	.120663	.134028	.149868	.160809	.192684
110	.100765	.115127	.127874	.142985	.153424	.183843
120	.0965371	.110291	.122500	.136974	.146974	.176119
130	.0928023	.106019	.117753	.131665	.141276	.169296
140	.0894716	.102210	.113520	.126930	.136195	.163210
150	.0864769	.0987857	.109714	.122673	.131627	.157738
160	.0837651	.0956848	.106268	.118818	.127491	.152783
170	.0812944	.0928596	.103129	.115307	.123723	.148268
180	.0790308	.0902715	.100253	.112090	.120271	.144132
190	.0769471	.0878890	.0976054	.109129	.117093	.140325
200	.0750204	.0856863	.0951578	.106391	.114155	.136804
210	.0732321	.0836418	.0928860	.103850	.111429	.133537
220	.0715662	.0817373	.0907699	.101483	.108889	.130493
230	.0700095	.0799577	.0887925	.0992717	.106515	.127649
240	.0685504	.0782898	.0869393	.0971989	.104291	.124983
250	.0671792	.0767223	.0851977	.0952510	.102200	.122478
260	.0658873	.0752456	.0835570	.0934160	.100231	.120118
270	.0646675	.0738513	.0820078	.0916833	.0983719	.117889
280	.0635131	.0725319	.0805419	.0900438	.0966124	.115781
290	.0624187	.0712809	.0791521	.0884894	.0949443	.113781
300	.0613790	.0700927	.0778320	.0870129	.0933599	.111883
310	.0603899	.0689621	.0765759	.0856081	.0918524	.110076
320	.0594471	.0678847	.0753789	.0842694	.0904158	.108354
330	.0585473	.0668563	.0742365	.0829917	.0890446	.106710
340	.0576873	.0658735	.0731445	.0817705	.0877342	.105140
350	.0568642	.0649328	.0720995	.0806018	.0864800	.103636
360	.0560754	.0640314	.0710981	.0794819	.0852782	.102196
370	.0553187	.0631666	.0701374	.0784075	.0841252	.100814
380	.0545918	.0623360	.0692147	.0773755	.0830178	.0994867
390	.0538930	.0615374	.0683276	.0763834	.0819532	.0982106
400	.0532204	.0607688	.0674737	.0754285	.0809285	.0969824
410	.0525724	.0600284	.0666512	.0745087	.0799414	.0957992
420	.0519476	.0593144	.0658581	.0736217	.0789896	.0946584
430	.0513446	.0586254	.0650927	.0727658	.0780711	.0935574
440	.0507622	.0579599	.0643534	.0719390	.0771839	.0924940
450	.0501992	.0573166	.0636388	.0711399	.0763264	.0914662
460	.0496545	.0566943	.0629476	.0703669	.0754968	.0904719
470	.0491273	.0560918	.0622784	.0696185	.0746938	.0895093
480	.0486165	.0555083	.0616302	.0688936	.0739159	.0885769
490	.0481214	.0549426	.0610018	.0681909	.0731618	.0876731
500	.0476412	.0543938	.0603923	.0675093	.0724304	.0867964

Table 14: Half-width $d(n, \alpha, \rho = 6)$ to six significant digits for $n = 60$ to $n = 500$.

Sample size n	Attributes of $P(D_n \geq d)$	Input test statistic d 3/100	31/1000	311/10000	3111/100000
200	No. of numerator digits	441	581	730	930
	No. of denominator digits	441	581	730	930
	p value to 5 decimal digits	0.99144	0.98758	0.98713	0.98709
	Time in seconds to compute	0.11	0.156	0.203	0.265
1,000	No. of numerator digits	2973	2975	3700	4700
	No. of denominator digits	2973	2975	3701	4700
	p value to 5 decimal digits	0.32269	0.28581	0.28229	0.28194
	Time in seconds to compute	14.531	15.609	27.36	43.25
2,000	No. of numerator digits	6566	6568	7966	9408
	No. of denominator digits	6567	6569	7967	9410
	p value to 5 decimal digits	0.053546	0.041923	0.040891	0.040789
	Time in seconds to compute	201.969	209.579	371.578	573.047

Table 15: p values using the Durbin recursion formula. (All timings on a Pentium IV running at 2.4 GHz.)

sample size n affect the complexity of the p value, consider the following three examples. $P(D_4 \geq 3/10) = 1927/2500$ has 4 digits in both the numerator and denominator integers. $P(D_4 \geq 31/100) = 9240701/12500000$ has 7 digits in the numerator integer and 8 digits in the denominator integer. $P(D_6 \geq 3/10) = 2247811/4050000$ has 7 digits in both the numerator and denominator integers. These three examples suggest that the complexity of the p value increases as the complexity of the test statistic increases and as the sample size increases. Table 15 shows that both the complexity of the p value and the time in seconds to compute the p value, increases with the complexity of the test statistic d and the sample size n.

14. Computational times using the half-widths

Using selected half-widths from Tables 13 and 14, Table 16 contains the Durbin computational times so a time for $\rho = 3$ significant digits can be easily compared to the corresponding time for $\rho = 6$. As expected from the results in Section 13, the computational times for half-widths with $\rho = 6$ significant digits are greater than those for $\rho = 3$. The computational times in Table 16 were produced by the Mathematica function `TimingToFileByAlphaRationalDurbin` contained in Section 12 of the `KS2SidedOneSampleRational.nb` file.

15. Summary and conclusions

Eight different formulae (the exact, Kolmogorov, Tingey, Epanechnikov, Noe, Durbin, Pomeranz, and Durbin matrix) for calculating the two-sided one sample K-S p value were identified.

Sample size n	Digits of precision ρ	Time in seconds for Durbin to calculate $P[D_n \geq d(n,\alpha,\rho)]$ $\alpha = 0.02$	$\alpha = 0.1$	$\alpha = 0.05$	$\alpha = 0.02$	$\alpha = 0.01$	$\alpha = 0.001$
50	3	0.031	0.016	0.047	0.047	0.047	0.062
	6	0.047	0.047	0.062	0.063	0.062	0.063
100	3	0.109	0.125	0.141	0.125	0.172	0.203
	6	0.172	0.234	0.250	0.266	0.296	0.344
200	3	0.328	0.734	0.828	0.735	0.797	0.984
	6	1.110	1.343	1.500	1.453	1.313	1.844
300	3	1.609	2.032	2.187	2.000	2.938	2.609
	6	2.719	4.016	3.390	5.235	5.734	5.828
400	3	3.359	3.985	2.187	5.235	5.718	5.297
	6	6.766	7.969	9.562	9.125	9.891	14.125
500	3	4.391	5.219	6.000	4.921	7.547	9.453
	6	11.093	13.594	15.938	18.359	17.594	23.594
1,000	3	30.094	21.203	41.485	47.515	52.969	41.093
	6	71.781	82.125	97.391	98.141	153.234	192.016
2,000	3	256.235	315.328	359.906	416.062	457.422	344.390
	6	574.828	1017.270	1110.080	677.265	1387.490	1668.440

Table 16: Time In seconds for Durbin to calculate $P[D_n \geq d(n,\alpha,\rho)]$. (All timings on a Pentium IV running at 2.4 GHz.)

Since the exact formula has a different piecewise polynomial for each sample size and the number of terms in the piecewise polynomial grows rapidly with the sample size, the piecewise polynomial has only been generated for sample sizes $n \leq 31$. Of the remaining seven formulae, the Kolmogorov, Tingey, Epanechnikov, Noe, Durbin, and Pomeranz formulae are complex recursion formulae while the Durbin matrix formula is a matrix formula. Because of the complexity of these seven formulae which were all derived prior to 1973, it has been impossible to calculate p values exactly. Given the advances in computing power and computational software in the past thirty years, it is time to devise techniques to accurately calculate the two-sided one sample K-S p values and determine which of the seven formulae is most efficient. Since rational arithmetic allows the exact evaluation of each formula, all seven formulae were implemented in rational arithmetic using Mathematica and the correctness of the code verified by comparing various p values generated by all eight formulae.

For the six recursion formulae and the Durbin matrix formula, the analysis of computational times revealed that the Durbin recursion formula is almost always the fastest. Using the

Durbin recursion formula, a linear search technique to determine the two-sided half-width to a specified number of digits of precision ρ was developed. Using the half-widths as input, the computational time needed by the Durbin recursion formula was found for various sample sizes $n \leq 2,000$, p values α, and digits of precision ρ. Computational time increases with increasing sample size n, decreasing p value α, and increasing digits of precision ρ.

One direction for future research is to use the methodology in this paper to systematically evaluate other K-S sampling distributions. A logical candidate for such an in-depth analysis is the commonly used two-sample K-S sampling distributions (difference between two empirical distributions) for equal and unequal sample sizes. Other K-S tests such as restricted-range tests (two distributions compared over a portion of their range) and ratio tests (the ratio of one distribution to another) have not been implemented in commercial software because of the difficulty in calculating p values due to the complexity of the formulae. In order for these tests to be implemented in commercial software and become available to the general user, research using the methodology developed in this paper is a necessary first step.

Acknowledgments

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

References

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 (1999). *Practical Nonparametric Statistics*. 3rd edition. John Wiley & Sons, New York.

Daniel WW (1990). *Applied Nonparametric Statistics*. Second edition. PWS-KENT Publishing Company, Boston, Massachusetts.

Drew JH, Glen AG, Leemis LM (2000). "Computing the Cumulative Distribution Function of the Kolmogorov-Smirnov Statistic." *Computational Statistics & Data Analysis*, **34**(1), 1–15.

Durbin J (1968). "The Probability that the Sample Distribution Function Lies Between Two Parallel Straight Lines." *The Annals of Mathematical Statistics*, **39**(2), 398–411.

Durbin J (1973). *Distribution Theory for Tests Based on the Sample Distribution Function*. Society for Industrial and Applied Mathematics, Philadelphia.

Epanechnikov VA (1969). "The Significance Level and Power of the Two-Sided Kolmogorov Test in the Case of Small Sample Sizes." *Theory of Probability and Its Applications*, **13**, 686–690.

Gibbons JD, Chakraborti S (2003). *Nonparametric Statistical Inference*. Fourth edition. Marcel Dekker, New York.

Kemperman JHB (1961). *The Passage Problem for a Stationary Markov Chain*. University of Chicago Press, Chicago.

Kolmogorov A (1933). "Sulla Determinazione Empirica di una Legge di Distribuzione." *Giornala dell'Istitoto Italiano degli Attuari*, **4**, 83–91.

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

Maplesoft (2008). *Maple 12*. Waterloo Maple Inc., Waterloo. URL http://www.maplesoft.com/products/maple/.

Marsaglia G, Tsang WW, Wang J (2003). "Evaluating Kolmogorov's Distribution." *Journal of Statistical Software*, **8**(18), 1–4. URL http://www.jstatsoft.org/v18/i08/.

Massey FJ (1950). "A Note on the Estimation of a Distribution Function By Confidence Limits." *The Annals of Mathematical Statistics*, **21**(1), 116–119.

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

Pomeranz JE (1973). "Exact Values of the Two-Sided Kolmogorov-Smirnov Cumulative Distribution for Finite Sample Size." *Technical report*, Purdue University.

Ruben H, Gambino J (1982). "The Exact Distribution of Kolmogorov's Statistic D_n for $n \leq 10$." *The Annals of the Institute Statistical Mathematics*, **34**, 167–173.

Sprent P, Smeeton NC (2001). *Applied Nonparametric Statistical Methods*. Chapman & Hall/CRC, New York.

Steck GP (1971). "Rectangle Probabilities for Uniform Order Statistics and the Probability that the Empirical Distribution Function Lies Between Two Distribution Functions." *The Annals of Mathematical Statistics*, **42**, 1–11.

Tingey FH (1951). *Contributions to the Theory of Kolmogorov's Statistic*. Ph.D. thesis, University of Washington.

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 2 *Submitted:* 2007-05-23
June 2008 *Accepted:* 2007-06-25