```
/*
The following program code generates the approximated tail lower bound
distribution of the
QLR tet statistic given in the papger, Testing for Regime Switching by Cho, JS
and White, H.

The following four different cases are those considered in Table 1 and obtained
by Theorem 7
in the paper after letting gamma be 2, so that Pickand's constant is 1/sqrt(pi).

Case 1: When the variance term is known, and the null parameter value is at the
center of
        the parameter space.

        The matrix "ttc" given below provides the desired lower bound. The first
column of
        "ttc" represent the percentile, and the other columns are the qunatiles
corresponding
        to the various paraemeter spaces considered in the paper. For example,
ttc[.,2]
        provides the tail lower bound when the parameter space is [-1, 1].

Case 2: When the variance term is known, and the null parameter value is at the
corner of
        the parameter space.

        The matrix "ttc0" given below provides the desired lower bound. The
first column of
        "ttc0" represents the percentile, and the other columns are the
qunatiles corresponding
        to the various paraemeter spaces consdiered in the paper. For example,
ttc0[.,2]
        provides the tail lower bound when the parameter space is [0, 1].

Case 3: When the variance term is unknown, and the null parameter value is at
the center of
        the parameter space.

        The matrix "ttd" given below provides the desired lower bound. The first
column of
        "ttd" represents the percentile, and the other columns are the qunatiles
corresponding
        to the various paraemeter spaces considered in the paper. For example,
ttd[.,2]
        provides the tail lower bound when the parameter space is [-1, 1].

Case 4: When the variance term is unknown, and the null parameter value is at
the corner of
        the parameter space.

        The matrix "ttd0" given below provides the desired lower bound. The
first column of
        "ttd0" represents the percentile, and the other columns are the
qunatiles corresponding
        to the various paraemeter spaces consdiered in the paper. For example,
ttd0[.,2]
        provides the tail lower bound when the parameter space is [0, 1].
```

```
Comment: Researchers can modify the parameter space.
*/

new;
library pgraph;
_pdate = {0};

uuu = seqa(0, 10/10000, 10001);
ttc = zeros(10001,6);
ttc0= zeros(10001,6);
ttd = zeros(10001,6);
ttd0= zeros(10001,6);

/* parameter space [-1, 1] */
upp1 = 1; low1 = -1;

/* parameter space [-2, 2] */
upp2 = 2; low2 = -2;

/* parameter space [-3, 3] */
upp3 = 3; low3 = -3;

/* parameter space [-4, 4] */
upp4 = 4; low4 = -4;

/* parameter space [-5, 5] */
upp5 = 5; low5 = -5;

/* parameter space [0, 1] */
upp1 = 1; low0 = 0;

/* parameter space [0, 1] */
upp2 = 2; low0 = 0;

/* parameter space [0, 1] */
upp3 = 3; low0 = 0;

/* parameter space [0, 1] */
upp4 = 4; low0 = 0;

/* parameter space [0, 1] */
upp5 = 5; low0 = 0;

lam11 = upp1-low1;
lam22 = upp2-low2;
lam33 = upp3-low3;
lam44 = upp4-low4;
lam55 = upp5-low5;
lam01 = upp1-low0;
lam02 = upp2-low0;
lam03 = upp3-low0;
lam04 = upp4-low0;
lam05 = upp5-low0;

iii = 1;
do until iii > 10001;
```

```
    ttc[iii,1] = sqrt(uuu[iii,1]);
    ttc[iii,2] = 1-lam11/sqrt(pi)*ttc[iii,1]*(1-cdfn(ttc[iii,1]));
    ttc[iii,3] = 1-lam22/sqrt(pi)*ttc[iii,1]*(1-cdfn(ttc[iii,1]));
    ttc[iii,4] = 1-lam33/sqrt(pi)*ttc[iii,1]*(1-cdfn(ttc[iii,1]));
    ttc[iii,5] = 1-lam44/sqrt(pi)*ttc[iii,1]*(1-cdfn(ttc[iii,1]));
    ttc[iii,6] = 1-lam55/sqrt(pi)*ttc[iii,1]*(1-cdfn(ttc[iii,1]));

    ttc0[iii,1] = sqrt(uuu[iii,1]);
    ttc0[iii,2] = 1-lam01/sqrt(pi)*ttc0[iii,1]*(1-cdfn(ttc0[iii,1]));
    ttc0[iii,3] = 1-lam02/sqrt(pi)*ttc0[iii,1]*(1-cdfn(ttc0[iii,1]));
    ttc0[iii,4] = 1-lam03/sqrt(pi)*ttc0[iii,1]*(1-cdfn(ttc0[iii,1]));
    ttc0[iii,5] = 1-lam04/sqrt(pi)*ttc0[iii,1]*(1-cdfn(ttc0[iii,1]));
    ttc0[iii,6] = 1-lam05/sqrt(pi)*ttc0[iii,1]*(1-cdfn(ttc0[iii,1]));

    ttd[iii,1] = sqrt(uuu[iii,1]);
    ttd[iii,2] = 1-lam11/sqrt(pi)*ttd[iii,1]*(1-cdfn(ttd[iii,1])) - (1-
cdfn(ttd[iii,1]));
    ttd[iii,3] = 1-lam22/sqrt(pi)*ttd[iii,1]*(1-cdfn(ttd[iii,1])) - (1-
cdfn(ttd[iii,1]));
    ttd[iii,4] = 1-lam33/sqrt(pi)*ttd[iii,1]*(1-cdfn(ttd[iii,1])) - (1-
cdfn(ttd[iii,1]));
    ttd[iii,5] = 1-lam44/sqrt(pi)*ttd[iii,1]*(1-cdfn(ttd[iii,1])) - (1-
cdfn(ttd[iii,1]));
    ttd[iii,6] = 1-lam55/sqrt(pi)*ttd[iii,1]*(1-cdfn(ttd[iii,1])) - (1-
cdfn(ttd[iii,1]));

    ttd0[iii,1] = sqrt(uuu[iii,1]);
    ttd0[iii,2] = 1-lam01/sqrt(pi)*ttd0[iii,1]*(1-cdfn(ttd0[iii,1])) - (1-
cdfn(ttd0[iii,1]));
    ttd0[iii,3] = 1-lam02/sqrt(pi)*ttd0[iii,1]*(1-cdfn(ttd0[iii,1])) - (1-
cdfn(ttd0[iii,1]));
    ttd0[iii,4] = 1-lam03/sqrt(pi)*ttd0[iii,1]*(1-cdfn(ttd0[iii,1])) - (1-
cdfn(ttd0[iii,1]));
    ttd0[iii,5] = 1-lam04/sqrt(pi)*ttd0[iii,1]*(1-cdfn(ttd0[iii,1])) - (1-
cdfn(ttd0[iii,1]));
    ttd0[iii,6] = 1-lam05/sqrt(pi)*ttd0[iii,1]*(1-cdfn(ttd0[iii,1])) - (1-
cdfn(ttd0[iii,1]));

    iii = iii + 1;
endo;

print "Case 1";
print uuu~ttc[.,2:6];

print "Case 2";
print uuu~ttc0[.,2:6];

print "Case 3";
print uuu~ttd[.,2:6];

print "Case 4";
print uuu~ttd0[.,2:6];
```