```
C
C        CHECK FOR CONVERGENCE
C
      EPS = ZERO
      DO 9 L = 1, IP
      DO 9 J = 1, IP
      DIFF = ABS(B(L, J) - BOLD(L, J))
      IF (DIFF .GT. EPS) EPS = DIFF
    9 CONTINUE
      IF (EPS .GE. EPSF .AND. NF .LT. MAXF) GOTO 4
      IF (EPS .GE. EPSF) IFAULT = 4
      RETURN
      END
```

Remark AS R72

A Remark on Algorithm AS 128. Approximating the Covariance Matrix of Normal Order Statistics

By B. L. Shea† and A. J. Scallon

Manchester Polytechnic, UK *Polytechnic of North London, UK*

[Received October 1986. Revised July 1987]

Purpose

Davis and Stephens (1978) have provided an algorithm for approximating the variance–covariance matrix of an ordered sample of independent observations from a standard normal distribution. Their algorithm requires the user to supply values for $V11$ (the exact value for the variance of the largest order statistic), $EX1$ (the expected value of the largest order statistic), $EX2$ (the expected value of the second largest order statistic) and $SUMM2$ (the sum of the squares of the expected values of the order statistics). Since AS 128 was published Royston (1982) has supplied a routine to compute exact and approximate values for the latter three quantities. To date, there appears to be no algorithm to compute $V11$. Davis and Stephens (1978) have pointed out that values of $V11$ are tabulated in Ruben (1954) and Borenius (1966). Unfortunately this is only for sample sizes up to 120. Even if these tables went beyond a sample size of 120, it would still be useful to have a routine to compute the exact value of $V11$ for an arbitrary sample size. The algorithm presented here computes an excellent approximation to $V11$ using a polynomial expansion.

Technique

For a sample of size n, the variance of the largest order statistic is given by:

$$V(n) = n \int_{-\infty}^{\infty} x^2 \{\Phi(x)\}^{n-1} \phi(x) \mathrm{d}x - \{E(n)\}^2 \qquad (1)$$

† *Address for correspondence*: Department of Mathematics and Physics, Manchester Polytechnic, All Saints, Manchester, M15 6BH, UK.

where $E(n)$ is the expected value of the largest order statistic, $\phi(x)$ is the standard normal density function and $\Phi(x)$ is the cumulative distribution function of a standard normal random variable. The integral on the right-hand side of (1) could be evaluated using a numerical integration procedure similar to that used in Royston's (1982) algorithm *NSCOR*1. However, a good approximation for $V(n)$ can be obtained by using a polynomial expansion of the form:

$$V(n) = \exp\left\{\sum_{i=0}^{r} c_i[x(n, \lambda)]^i\right\}.$$

Table 1 gives the values chosen for r and the functional form of $x(n, \lambda)$ for different values of n. The coefficients $\{c_i\}$ for $i = 0, 1, \ldots, r$ and λ were recalculated (using least squares) for different ranges of values of n.

Structure

REAL FUNCTION V11(N, IFAULT)

Formal parameters

N	Integer	input:	sample size, n ($\geqslant 1$)
IFAULT	Integer	output:	a fault indicator
			$= 1$ if $N < 1$;
			$= 0$ otherwise (on a successful exit)

Precision

This algorithm was developed on a 32-bit machine (PRIME 9750). It may be that the user's calling (sub)program requires *V*11 to be implemented in double precision. To construct a double precision version we simply replace *REAL* by *DOUBLE PRECISION* in lines 1 and 9.

Timing and Accuracy

The number of multiplications and divisions required does not depend on n. Comparison with the tabulated values in Borenius (1966) and with the exact values obtained using numerical integration reveals that for values of n up to 500 *V*11 is accurate to at least 6 decimal places on a 32-bit machine. For $n > 500$, *V*11 is accurate to approximately 5 decimal places.

The upper entries displayed in Table 2 are the values of the input arguments for subroutine *COVMAT* (AS 128) obtained using Royston's (1982) *NSCOR*1 algorithm

TABLE 1

Range of values of n	r	$x(n, \lambda)$
$2 \leqslant n \leqslant 100$	9	$(n^\lambda - 1)/\lambda$
$101 \leqslant n \leqslant 200$	5	$\log(\lambda + n)$
$201 \leqslant n \leqslant 370$	5	$\log(\lambda + n)$
$n > 370$	4	$(n^\lambda - 1)/\lambda$

TABLE 2
Sample size (n)

	10	*50*	*100*	*500*	*1000*	*2000*
$EX1$	1.538753	2.249074	2.507594	3.036699	3.241436	3.435337
	1.538776	2.249029	2.507579	3.036737	3.241436	3.435265
$EX2$	1.001357	1.854872	2.148145	2.732308	2.954133	3.162570
	1.001350	1.854839	2.148145	2.732326	2.954116	3.162496
$SUMM2$	7.914272	47.421696	97.259994	496.958582	996.851672	1996.754915
	7.914321	47.421675	97.259692	496.959319	996.851860	1996.752644
$V11$	0.344344	0.215712	0.184404	0.137201	0.123454	0.112191
Time (s)	0.66	2.65	4.64	17.68	32.54	61.29
	0.01	0.03	0.04	0.21	0.42	0.82

for different values of n. The time spent in both $NSCOR1$ and $V11$ is also given. These results were obtained on a PRIME 9750 machine using the FTN77 compiler. The lower entries in Table 2 were obtained using $NSCOR2$ instead of $NSCOR1$. $NSCOR1$ only has high accuracy on a machine of small word-length if it is run in double precision (see p. 163, Royston (1982)). Hence the results obtained in Table 2 used double precision versions of $NSCOR1$, $NSCOR2$ and $V11$.

Table 3 below gives the maximum absolute difference between corresponding elements of the matrix V on exit from $COVMAT$ when the input arguments are calculated using $NSCOR1$ and $NSCOR2$. The time taken to compute V is also displayed.

TABLE 3

n	*Maximum absolute error*	*Time (s)*
10	0.000058	0.10
50	0.000041	2.21
100	0.000153	8.07
200	0.000050	32.23
300	0.000230	72.07
350	0.000289	98.37

From Table 3 we see that when $NSCOR2$ is used to compute the expected values of the order statistics instead of $NSCOR1$, the elements of V on exit from $COVMAT$ are usually accurate to about 4 decimal places.

Further Comments

In AS 128 the array V should be declared as $V(MDIM, N)$ rather than $V(MDIM, MDIM)$, for as it stands a large non-square array capable of holding V cannot be passed in.

In order to prevent overflow occurring when $N = 2$ in the last line of the following statements:

```
      SUM = ZERO
      DO 80 J = 3, N
80    SUM = SUM + V(1, J)
      CNST = (ONE - V(1, 1) - V(1, 2))/SUM
```

the statement

```
      IF(N .EQ. 2) RETURN
```

must be inserted before SUM = ZERO.

Acknowledgement

We are grateful to the referee for some helpful suggestions.

References

Borenius, G. (1966) On the limit distribution of an extreme value in a sample from a normal approximation. *Skand. Aktuartidskr.*, 1965, 1–15.

Davis, C. S. and Stephens, M. A. (1978) Algorithm AS128. Approximating the covariance matrix of normal order statistics. *Appl. Statist.*, **27**, 206–212.

Harter, H. L. (1961) Expected values of normal order statistics. *Biometrika*, **48**, 151–165.

Royston, J. P. (1982) Algorithm AS177. Expected normal order statistics (exact and approximate). *Appl. Statist.*, **31**, 161–165.

Ruben, H. (1954) On the moments of order statistics in samples from normal populations. *Biometrika*, **41**, 200–227.

```
      REAL FUNCTION V11(N, IFAULT)
C
C        ASR 72 (REMARK ON AS 128) APPL. STATIST. (1988) VOL. 37, NO. 1
C
C        CALCULATES AN APPROXIMATION TO THE VARIANCE OF THE
C        LARGEST NORMAL ORDER STATISTIC.
C
      INTEGER N, IFAULT
      REAL ZERO, ONE, A0, A1, A2, A3, A4, A5, A6, D0, X, D1, D2, D3, D4,
     *  D5, D6, PT09, C0, C1, C2, C3, C4, C5, C6, C7, C8, C9,
     *  MPT15, B0, B1, B2, B3, B4
      PARAMETER(MPT15 = -0.15, B0 = -0.934E-4, ZERO = 0.0, ONE = 1.0,
     *  B1 = -0.5950321, B2 = 0.0165504, B3 = 0.0056975,
     *  PT09 = 0.091105452691946, C0 = 0.7956E-11,
     *  C1 = -0.595628869836878, C2 = 0.08967827948053,
     *  C3 = -0.007850066416039, C4 = -0.296537314353E-3,
     *  C5 = 0.215480033104E-3, C6 = -0.33811291323E-4,
     *  C7 = 0.2738431187E-5, C8 = -0.106432868E-6, C9 = 0.1100251E-8,
     *  A0 = 0.046198318476960, A1 = -0.147930264017706,
     *  A2 = -0.451288155800301, A3 = 0.010055707621709,
     *  A4 = 0.007412441980877, A5 = -0.001143407259055,
     *  A6 = 0.54428754576E-4, D0 = 0.093256818332708,
     *  D1 = 1.336952989217635, D2 = -1.783195691545387,
     *  D3 = 0.488682076188729, D4 = -0.078737246197474,
     *  D5 = 0.006625619878060, D6 = -0.226486218258E-3,
     *  B4 = -0.8531E-3)
      INTRINSIC EXP, LOG
C
      V11 = ZERO
      IFAULT = 1
      IF (N .LT. 1) RETURN
      IFAULT = 0
      IF (N .EQ. 1) THEN
         V11 = ONE
         RETURN
      ENDIF
```

```
C
      X = N
      IF (N .GT. 370) THEN
         X = ((X ** MPT15) - ONE) / MPT15
         V11 = EXP(B0 + X * (B1 + X * (B2 + X * (B3 + X * B4))))
      ELSEIF (N .LE. 100) THEN
         X = ((X ** PT09) - ONE) / PT09
         V11 = EXP(C0 + X * (C1 + X * (C2 + X * (C3 + X * (C4 + X * (C5
     *      + X * (C6 + X * (C7 + X * (C8 + X * C9)))))))))
      ELSEIF (N .LE. 200) THEN
         X = LOG(A0 + X)
         V11 = EXP(A1 + X * (A2 + X * (A3 + X * (A4 + X * (A5 + X * A6)
     *      ))))
      ELSE
         X = LOG(D0 + X)
         V11 = EXP(D1 + X * (D2 + X * (D3 + X * (D4 + X * (D5 + X * D6)
     *      ))))
      ENDIF
C
      END
```