Remark on Algorithm 705: A Fortran-77 Software Package for Solving the Sylvester Matrix Equation $AXB^T + CXD^T = E$

TIM HOPKINS
University of Kent

We present a number of corrections to Algorithm 705 [Gardiner et al. 1992].

Categories and Subject Descriptors: G.1.3 [**Numerical Analysis**]: Numerical Linear Algebra—*Linear systems (direct and iterative methods)*; G.4 [**Mathematics of Computing**]: Mathematical Software—*Reliability and robustness*

General Terms: Algorithms, Performance, Reliability

The following problems were encountered in Algorithm 705 when the source code was being tested as part of the project to bring all the CALGO algorithms that have appeared in ACM TOMS to a consistent level of quality [Hopkins 2002].

The statements provided when updating the 2×2 `R` submatrix for the case when `I = K` and `JOB = 1` are incorrect in both the routines `BKCON` and `BKDIS`. In each case, where the linear system solved is of order 3, element 4 of the approximation to the null vector returned by the Linpack [Dongarra et al. 1979] routine `DGECO` is accessed when it has not been assigned a value.

In `BKCON` in the `IF` block guarded by

```
      IF (DOSEP) THEN
```

following the statement

```
      NSYS = 3
```

replace

```
              TMP = R(K,K)*E(1) + R(K,K-1)*E(2)
     *            + R(K-1,K)*E(2) + R(K-1,K-1)*E(4)
```

by

```
              TMP = R(K,K)*E(1) + R(K,K-1)*E(2)
     *            + R(K-1,K)*E(2) + R(K-1,K-1)*E(3)
```

Author's address: Computing Laboratory, University of Kent, Canterbury, Kent, CT2 7NF, UK.

In BKDIS in a similar position replace

```
               TMP = R(K,K)*E(1) + R(I,K-1)*E(2)
     *             + R(K-1,K)*E(3) + R(K-1,K-1)*E(4)
```

by

```
               TMP = R(K,K)*E(1) + R(K,K-1)*E(2)
     *             + R(K-1,K)*E(2) +R(K-1,K-1)*E(3)
```

and

```
               R(K-1,K) = R(K-1,K) + TMP*E(3)
               R(K-1,K-1) = R(K-1,K-1) + TMP*E(4)
```

by

```
               R(K-1,K) = R(K-1,K) + TMP*E(2)
               R(K-1,K-1) = R(K-1,K-1) + TMP*E(3)
```

There are also problems in the routines HSCO and HSSL where calls to the BLAS [Lawson et al. 1979] routines DDOT and DAXPY cause array accesses outside of the range of the actual vector arguments. To prevent these illegal, out of bounds, assignments taking place:

in the routine HSCO replace the statement in the DO 120 block

```
         IF (K .LT. N) Z(K) = Z(K) + DDOT(SD,AV(I1+1),1,Z(K+1),1)
```

by

```
         IF (K .LT. N)
     +      Z(K) = Z(K) + DDOT(MIN(SD, N-K),AV(I1+1),1,Z(K+1),1)
```

and replace the statement in the DO 140 block

```
         IF (K .LT. N) CALL DAXPY(SD,T,AV(I1+1),1,Z(K+1),1)
```

by

```
         IF (K .LT. N)
     +      CALL DAXPY(MIN(SD, N-K),T,AV(I1+1),1,Z(K+1),1)
```

Similarly in the DO 20 block in the routine HSSL replace

```
            CALL DAXPY(SD,T,AV(I1+1),1,B(K+1),1)
```

by

```
            CALL DAXPY(MIN(SD, N-K),T,AV(I1+1),1,B(K+1),1)
```

In the routine BKHS2, it is necessary to change the test immediately following label 110 from

```
    IF (M.GT.1) THEN
```

to

```
    IF (M.GT.1 .AND. J.NE.M) THEN
```

to avoid accessing row `M+1` of the array `P` which is only required to be `M` by `M`.

The Fortran standard does not prescribe how the components of a compound relational expression should be evaluated. In particular it is not standard conforming to assume either left to right evaluation of subexpressions or short circuit evaluation (for example, only the first component of an 'and' is evaluated if it is found to be false). Thus the four statements of the form

```
IF (I .GT. 1  .AND.  S(I,I-1) .NE. 0.0D0) GO TO label
```

that appear in routines `BKCON` and `BKDIS` (after labels 40 and 150 in both routines) should be replaced by

```
IF (I .GT. 1 ) THEN
  IF (S(I,I-1) .NE. 0.0D0) GO TO label
ENDIF
```

This is guaranteed to avoid the out of bounds array access to `S(1,0)` in the case of compilers that evaluate the right-hand subexpression.

In addition, it is possible to replace the declarations of all dummy arguments of length 1 in the routines `BKCON`, `BKDIS`, `BKHS2`, `HSCO`, `HSSL`, `HSFA`, `QZHESG`, `QZITG`, `SEPG`, `SEPGC`, `SEPGD`, `SYLG`, `SYLGC` and `SYLGD` by expressions involving other dummy arguments (there is one exception, the workspace array `WKV` to `SYLG`, which needs to be assumed size).

Calls to the routine `EPSLON` used by Eispack [Hopkins and Slater 1993] to obtain a value for the machine epsilon were replaced with calls to the Port function `D1MACH` [Fox et al. 1978] both for efficiency and consistency reasons.

In the test driver programs array declarations were parameterized to make the codes easier to alter and to keep the reserved space down to an essential minimum. This helped to ensure that array bound violations were more likely to be detected by the compiler's run-time system since the lengths of many of the workspace arrays were overestimated.

Finally, there are some minor pieces of clutter that may be removed from the code

(1) the intrinsic functions `DABS` and `DMAX1` do not need to be declared as double precision in the function `D1NRM`,

(2) the variable `INDXTO` is declared and initialized in both the routines `SYLGC` and `SYLGD` but is not referenced in either routine. All references to `INDXTO` may thus be removed.

(3) The label 100 in the routine `QZVALG` is never referenced and may be removed.

(4) The replacement of all Hollerith constants by character constants.

(5) The reliance by one of the test driver programs on the Unix random number generator has been removed. The driver program has been altered to use Schrage's portable random number generator [Schrage 1979]. This allows the software to be more easily ported to non-Unix platforms and also simplifies the comparison of results from different compiler/hardware combinations by ensuring a repeatable sequence.

REFERENCES

DONGARRA, J. J., MOLER, C. B., BUNCH, J. R., AND STEWART, G. W. 1979. *LINPACK: Users' Guide.* SIAM, Philadelphia.

FOX, P. A., HALL, A. D., AND SCHRYER, N. L. 1978. Framework for a portable library. *ACM Trans. Math. Softw. 4,* 2 (June), 177–188.

GARDINER, J. D., WETTE, M. R., LAUB, A. J., AMATO, J. J., AND MOLER, C. B. 1992. A Fortran-77 software package for solving the Sylvester matrix equation $AXB^T + CXD^T = E$. *ACM Trans. Math. Softw. 18,* 2 (June), 232–238.

HOPKINS, T. 2002. Renovating the collected algorithms from acm. *ACM Trans. Math. Softw. 28,* 1, ???–???

HOPKINS, T. AND SLATER, J. 1993. A comment on the Eispack machine epsilon routine. *ACM SIGNUM 28,* 4 (Oct.), 2–6.

LAWSON, C. L., HANSON, R. J., KINCAID, D. R., AND KROGH, F. T. 1979. Basic linear algebra subprograms for Fortran usage. *ACM Trans. Math. Softw. 5*, 308–323.

SCHRAGE, L. 1979. A more portable fortran random number generator. *ACM Trans. Math. Softw. 5,* 2 (June), 132–138.