02-24-2002, 09:28 PM #1

Bill Press
Numerical Recipes Author

Join Date: Dec 2001
Posts: 160

Bugs in version 2.08 fixed in version 2.10

Here is the complete list of verified bugs in Numerical Recipes version 2.08. All of these were fixed in the latest version 2.10.

1. In Fortran 77 caldat and julday change all the real constants to double by appending "d0". In the Fortran 90 version, change all _sp's to _dp's. Fixes a roundoff problem present on some machines. No dates plus or minus 300 years of the present are affected. No change in C version is required.

2. In Fortran 77 and 90 julday change the obvious line to read

Code:

```
julday=365*jy+int(0.25*jy+2000.)+int(30.6001*jm)+id+1718995
```

No change in C version is required. Bug affects only some B.C. dates.

3. For icrc() in C, documentation should state clearly that the string to be checksummed is in locations bufptr[1..len], not bufptr[0..len-1].

4. In the main books, equations 18.3.14 and 18.3.16 are missing a minus sign before the ln. The programs and figure are correct.

5. In the C version of dftcor the obvious line should be

Code:

```
if (a >= b || th < 0.0e0 || th > 3.1416e0) ...
```

etc.

6. medfit encounters a divide by zero if the input vectors are a perfect straight line. The fix is to put a line

Code:

```
    if (sigb>0.0) {
```

just after the line f1=rofunc(b1), after sigb is computed, and then to put the corresponding line

Code:

```
    }
```

just before the line

Code:

```
*a=aa;
```

7. In mpinv.f90 (Fortran 90 version only), some Windows distribution media contain the incorrect line

Code:

```
    allocate(rr(max(n+m)+n+1),s(n))
```

instead of the intended and correct

Code:

```
    allocate(rr(max(n,m)+n+1),s(n))
```

8. In the vector version of bessjy.f90 (Fortran 90 version only) the line

Code:

```
where (h < FPMIN) h=FPMIN
```

in bessjy_xltxmin should be deleted. (It's already appeared, correctly, in the calling routine bessjy_v.) The bug causes many, but not all, values to be incorrect for x < 2.

9. shoot and shootf in the Fortran 90 version (ONLY) can give a runtime error because of the statement

Code:

```
if (associated(xp)) deallocate(xp,yp)
```

The associated function is called even if xp is not defined, which is an error. The fix is to move the statement

Code:

```
nullify(xp,yp)
```

in odeint.f90 to just before the first occurrence of the statement

Code:

```
if (save_steps) then
```

10. In rj.c, the lines

Code:

```
if (FMIN(FMIN(x,y),z) < 0.0 || FMIN(FMIN(x+y,x+z),FMIN(y+z,f
   || FMAX(FMAX(x,y),FMAX(z,fabs(p))) > BIG)
   nrerror("invalid arguments in rj");
```

should be

Code:

```
if (FMIN(FMIN(x,y),z) < 0.0 || FMIN(FMIN(FMIN(x+y,x+z),y+z),
   || FMAX(FMAX(FMAX(x,y),z),fabs(p)) > BIG)
   nrerror("invalid arguments in rj");
```

and similarly the lines

Code:

```
} while (FMAX(FMAX(fabs(delx),fabs(dely)),
   FMAX(fabs(delz),fabs(delp))) > ERRTOL);
```

should be

Code:

```
} while (FMAX(FMAX(FMAX(fabs(delx),fabs(dely)),
   fabs(delz)),fabs(delp)) > ERRTOL);
```

11. In indexx_i4b (Fortran 90 only) change do i=j-1,1,-1 to do i=j-1,l,-1 (i.e., change the limit from "one" to "ell"). There is no corresponding change needed in indexx.for or indexx.c. However, the routines iindexx.for and iindex.c (integer versions of indexx) need several changes to bring them into conformity with indexx.for and indexx.c. These are obvious from the respective codes.

12. Professor Christian Reinsch (reinsch@mathematik.tu-muenchen.de) has distributed the following error notice, which affects the hqr routine in Numerical Recipes (all versions, all languages):

The original Handbook routine HQR for the computation of all eigenvalues of a real, non-symmetric matrix by Francis' method of double-QR-steps contains an error. Since this routine has also been included in the Eispack-software and in the "Numerical Recipes", you are kindly asked to help inform potential users. The error occurs only in rare situations: If a subdiagonal element is negligible in the sense of routine
HQR, viz.

Code:

```
      |H[i,i-1]| <= epsmach * (|H[i-1,i-1]| + |H[i,i]|)
```

then the lower, righthand block [i:n]*[i:n] alone undergoes the next transformation. This saves computational labor. However, the new H[i,i] could be smaller, so that at the beginning of a subsequent iteration, the criterion (*) need no longer be satisfied. (The other two entries, H[i-1,i-1] and H[i,i-1], are not changed.) In such a case, a larger block (or even the entire matrix) will be treated, and matrix entries come back into play which are invalid since they missed previous QR steps.

As a correction, it is therefore suggested to explicitly reset to zero a subdiagonal entry which satisfies (*). This makes the breakpoint permanent and the splitting feasible. Another remedy would be to do the column modifications on the righthand block [i:n] * [1:n], (as it is done in the Handbook routine HQR2).

The error was detected when a C-version of routine HQR was applied to a large series of projectors A. Projectors have A*A = A, so that their eigenvalues must be from {0, 1} and in linear elementary divisors (1*1 Jordan blocks). For example, the 5*5 lower Hessenberg matrix

Code:

```
          7    0    0    0    0              1     7     49
         -5    8    4    0    0              1     7   -135
A =      10   -2   -1    0    0        X =   1     7     95        Y
        -10   -2    3    8    8              1   -33     -1
          5    3    1   -1   -1              1    12     -8
```

has A*X = 7*X and A*Y = 0. The routine HQR in IEEE double precision would deliver +-O(3e-8) instead of two times zero or O(1e-15). dist(span(X),span(Y)) = 0.368... shows that this matrix is well-conditioned.

C. Reinsch, Munich, Feb 16, 2000

« Previous Thread | Next Thread »

Posting Rules
HTML code is **Off**

All times are GMT -5. The time now is 05:31 AM.