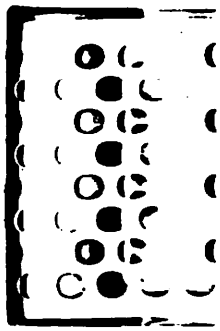
Underflow Can Hurt

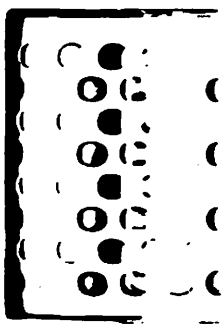
By Eric Grosse and Cleve Moler



The package of Fortran programs known by the acronym EISPACK is a systematized collection of subroutines that compute the eigenvalues and/or eigenvectors of six classes of matrices—complex general, complex Hermitian, real general, real symmetric, real symmetric tridiagonal, and special real tridiagonal. The subroutines are based mainly on Algol procedures published in the Handbook series of Springer-Verlag by J.H. Wilkinson and C. Reinsch. Initially, they were adapted to and tested on several different machines—IBM 360-370, CDC 6000-7000, Univac 1110, Honeywell 6070, DEC PDP-10, and Burroughs 6700. Two manuals for users of EISPACK were published by Springer-Verlag as part of the Lecture Notes in Computer Science series (B.T. Smith, J.M. Boyle, J.J. Dongarra, et al., Matrix Eigensystem Routines—EISPACK Guide, *ed. 2, 1976; B.S. Garbow, J.M. Boyle, J.J. Dongarra, C.B. Moler,* Matrix Eigensystem Routines—EISPACK Guide Extension, *1977).*

In an interesting case that surfaced recently, EISPACK failed to give the correct eigenvalues for what appeared to be an easy matrix. The difficulties were traced to floating point underflow. They are most insidious in double precision arithmetic on the Digital Equipment Corporation's VAX, where the "D" floating point format has an unfortunately small exponent range. A scaled version of the example, however, can fail on any machine, including machines that fully conform to the IEEE floating point standard. A simple change in the EISPACK top-level routine RS (real symmetric) should protect most users from the problem.

The example was provided by Guenter Ziegler of the University of Augsburg, West Germany, and Andrew Odlyzko of AT&T Bell Laboratories. They were investigating a question raised by Amir Dembo of Brown University regarding the distribution of rank in real symmetric Hankel matrices whose elements are +1 and −1. (A Hankel matrix is constant along each antidiagonal, although this feature is irrelevant for the present discussion.) One of their matrices is 9-by-9:



$$\begin{matrix}
-1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\
1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\
1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 \\
-1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 \\
-1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\
1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\
1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & -1 \\
-1 & -1 & 1 & 1 & -1 & -1 & 1 & -1 & 1 \\
-1 & 1 & 1 & -1 & -1 & 1 & -1 & 1 & 1
\end{matrix}$$

It is not obvious, but this matrix has four eigenvalues equal to zero and, hence, a rank of five. From the many possible methods for computing the rank of such matrices, Ziegler and Odlyzko, for convenience, chose to use the EISPACK routine RS and count the number of negligible computed eigenvalues. For this example, running on a VAX in D format double precision, EISPACK incorrectly claimed there were five eigenvalues on the order of roundoff error. The same program, running on almost any other computer, would produce the correct answer of four negligible eigenvalues.

The cause of the problem was found to be a catastrophic underflow in the EISPACK routine TQLRAT. This is a square-root-free variant of the QR algorithm for finding eigenvalues of a symmetric tridiagonal matrix. It operates on the squares of off-diagonal elements. On the VAX, the square of double precision roundoff error is roughly 10^{-34}, and the underflow limit is only 10^{-38}. These two numbers are not far enough apart for TQLRAT to operate properly. On other computers, similar difficulties will occur if the example is scaled by a factor on the order of the square root of the underflow limit. For IEEE machines the scale factor would have to be about 10^{-150}), and such examples are thus much less likely in practice, but TQLRAT might not properly handle any that do arise.

The easiest solution is to replace

```
CALL TQLRAT(N,W,FV2,IERR)
```

in EISPACK routine RS by

```
CALL TQL1(N,W,FV1,IERR).
```

Since TQL1 does not work with the squares of the tridiagonal elements, it is much less prone to underflow trouble. No change is needed in the computation of eigenvectors, for which RS calls TQL2 rather than TQLRAT.

An alternative solution, an improved version of TQLRAT, is available from the authors. It is, however, still limited to a smaller floating point exponent range than TQL1 and TQL2.

Ironically, advances in floating point hardware make the need for square-root-free algorithms less pressing. On one recent chip, the built-in square root is even slightly faster than division!

Eric Grosse is at AT&T Bell Laboratories, Murray Hill, New Jersey, and Cleve Moler is with Dana Computer, Sunnyvale, California.