
Subject: Re: SVDFIT Problems

Posted by [William Clodius](#) on Thu, 05 Sep 2002 23:43:20 GMT

[View Forum Message](#) <> [Reply to Message](#)

James Kuyper wrote:

> Bill wrote:

>> <snip>

>>

>> Two comments:

>>

>> 1. Unless the double keyword is being ignored, TOL should not be the same for

>> single and double precision. All computers IDL is currently available on use

>> IEEE 754 math. In this standard the mantissa is represented by 23 bits in

>> single precision and 52 bits in double precision. With 754's hidden bit,

>> single precision has a relative precision of $1/2^{24} \sim 6e-8$ and double has a

>> relative precision of $1/2^{53} \sim 1e-16$. SVD should identify as singular any

>> value that is largely determined by the precision of the arithmetic. Such

>

> True, but it should also identify as singular any value that is largely

> determined by the precision of the input data. With double precision

> floating point, the precision of the result is likely to be dominated by

> the precision of the input data, not by the precision of the arithmetic.

>

> That's why TOL shouldn't just slavishly depend upon the precision of the

> data type. It's also why TOL should be adjustable by the user.

Not quite true. TOL in the original Numerical Recipes routine is intended to represent exclusively the effects of the machine numerics on the solution of the equations. The effects of the precision of the input data are in effect handled by the keyword MEASURE_ERRORS in what is in the current version of IDL's SVDFIT. They need to be separated because in linear regression the effects of numerical precision tends to increase with an increasing number of measurements included in the regression, while effect of measurement precision (provided the measurements are independent) tends to decrease with an increasing number of measurements.

This discussion then prompted me to look more closely at IDL's code. In that code the determination of singularity uses the condition

```
small = where(VARIANCE lt max(VARIANCE)*THRESH, count)
```

where apparently THRESH is intended to replace Numerical Recipe's TOL. However in my old Fortran version of Numerical Recipes, (I don't have a C version readily available), the condition is actually equivalent to

```
small = where(SINGULAR_VALUE lt max(SINGULAR_VALUE)*TOL, count)
```

While the variance is related to the singular value through, the V matrix they are not linearly related (the variance is linearly related to the INVERSE SQUARES of the singular values) and there is no reason to expect that the two conditions would be equivalent. The call to NR__SVDFIT that does the actual fitting in SVDFIT does not return the singular value matrix or the V matrix so I cannot

currently do a detailed analysis, but I suspect the comparison is incorrect.
