Subject: Re: SVDFIT Problems
Posted by William Clodius on Thu, 05 Sep 2002 17:26:26 GMT
View Forum Message <> Reply to Message

Chris Torrence wrote:

> Hi Bill,
>
> Actually, the SVDFIT code *does* use THRESH = TOL*wmax. If you look
> carefully at the IDL code, around line 249, there is the following line:
>
>  small=WHERE(variance LE max(variance)*thresh, cc)
>
> (The variable name "thresh" was an unfortunate choice, and should really
> have been "tol".)
>
> Internally, the C code is identical to the Numerical Recipes code, except
> for the TOL value, which is 1e-9 for both single and double precision. We
> could consider adding a TOL keyword to the SVDFIT function, which would
> allow the user to change this default.
>
> Cheers,
>
> Chris
> Research Systems, Inc.
> <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 precison of 1/2^24 ~ 6e-8 and double has a relative precision of 1/2^53 ~ 1e-16. SVD should identify as singular any value that is largely determined by the precision of the arithmetic. Such values will be less than a small multiple of the relative precision with the largest eigenvalue. For single precision the Numerical Recipes code uses a value of 1e-5, or about 150 times the precision. The current value of 1.e-9 will miss many values that are effectively singular for single precision calculations. If double precision is used on more complicated problems, then perhaps a reasonable estimate of TOL is 150^2 * 1e-16 ~ 2e-12. The current value of 1.e-9 will treat many values that are not singular as singular for double precision calculations.

2. Even given the above I strongly believe that either Numerical Recipe's implementation of SVDFIT is wrong, or IDL has made a mistake in implementing it. For infinite precision arithmetic, scaling for each sample "measurement"

the dependent variable, the vector of values returned by the function of the measurement, and the dependent variable's sigma by the same sample dependent finite value has no effect on the solution of the equations. While that does not strictly hold for finite precision arithmetic, I would expect it to almost always hold for relative changes less than the square root of the precision. I would also expe that using such scalings to ensure that the typical values of the functions are on the order of one would improve the regression. Unfortunately doing this on my fits to proprietary data increases the number of sinularities identified by SVDFIT.

---