Subject: Re: IDL inverse matrix problem??
Posted by Craig Markwardt on Fri, 23 Jan 2015 19:50:46 GMT
View Forum Message <> Reply to Message

On Friday, January 23, 2015 at 1:32:44 PM UTC-5, Amin Farhang wrote:
> Dear Chris,
>
> I wonder that why IDL could not handle this, therefore check the inversion with matlab and Python but both of them return correct values. I think it is necessary that IDL developers consider this kind of issues, since I'm seeing that the Astronomer's community are slowly shift to Python or other languages.

A few more notes. The condition number of your example matrix (first element -6.8599934d+16) is about kappa = 3e9.

Let's check Wikipedia for Condition Number, "As a general rule of thumb, if the condition number kappa = 10^k, then you may lose up to k digits of accuracy on top of what would be lost to the numerical method due to loss of precision from arithmetic methods." (more caveats to be read in that article, it's a rule of thumb, not a firm equation)

In our case k = 9. In your matrix you were seeing ~1e-8 errors and double precision is capable of precision 2e-16, which implies a loss of precision of (1e-8/2e-16) ~ 8 digits. That's pretty close to the k=9 predicted by the rule of thumb. The rule of thumb seems plausible. I'm not sure how you will do much better with Python or Matlab, unless they have a quad-precision mode.

Also, I stand by my previous statement... If numerical precision is important to you, you probably should not be directly inverting the matrix. Keep on poking yourself in the eye if you want though. If you are trying to solve normal equations in a least squares problem, best to factor the matrix with a stable factorization algorithm. Or even better, skip the normal equations altogether and factor the Jacobian directly.

CM