

---

Subject: IDL inverse matrix problem??

Posted by [amin farhang](#) on Thu, 22 Jan 2015 19:24:04 GMT

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

---

Dear All,

I'm trying to inverse a big matrix with heavy elements. According to [http://www.idlcoyote.com/math\\_tips/sky\\_is\\_falling.html](http://www.idlcoyote.com/math_tips/sky_is_falling.html) discussion by David, I try to produce all elements in double precision. But the INVERT and LA\_INVERT return wrong matrices. For example:

First define a double precision matrix:

```
IDL> m1 = 9.5d15 + 0.5d18*randomu(1,1,5,/double)
```

```
IDL> m2 = -1d0 + 2d0*randomu(2,5,1,/double)
```

```
IDL> Am = m1 ## m2
```

```
IDL> print,Am
```

```
-6.8599934e+16 -8.3590379e+16 -1.8742199e+17 1.5697952e+17 1.1607144e+17
-1.2228319e+17 -1.4900449e+17 -3.3409009e+17 2.7982470e+17 2.0690379e+17
-1.4972730e+17 -1.8244569e+17 -4.0907020e+17 3.4262599e+17 2.5333938e+17
-1.0612015e+17 -1.2930950e+17 -2.8993102e+17 2.4283828e+17 1.7955584e+17
-1.1696076e+17 -1.4251900e+17 -3.1954867e+17 2.6764521e+17 1.9789821e+17
```

```
IDL> max_value = double(max(abs(Am)))
```

```
IDL> Am2 = la_invert(Am/max_value)/(max_value)
```

```
% LA_INVERT: Singular matrix encountered, STATUS=4.
```

```
% Execution halted at: $MAIN$
```

As you see la\_invert encounter error. Since the matrix determinant is big.

Now again examine the following definition:

```
IDL> Am = [[-6.8599934d+16,-8.3590379d+16,-1.8742199d+17,1.5697952d+17 ,
1.1607144d+17],$
[-1.2228319d+17,-1.4900449d+17,-3.3409009d+17,2.7982470d+17, 2.0690379d+17],$
[-1.4972730d+17,-1.8244569d+17,-4.0907020d+17,3.4262599d+17, 2.5333938d+17],$
[-1.0612015d+17,-1.2930950d+17,-2.8993102d+17,2.4283828d+17, 1.7955584d+17],$
[-1.1696076d+17,-1.4251900d+17,-3.1954867d+17,2.6764521d+17, 1.9789821d+17]]
```

```
IDL> max_value = double(max(abs(Am)))
```

```
IDL> Am2 = la_invert(Am/max_value)/(max_value)
```

As you see this time IDL return the inverse BUT this inverse is WRONG!!!

Let me test the IDL inversion:

```
IDL> Bm = Am ## Am2
```

```
IDL> print,Bm
```

```
0.99999999 -1.3038516e-08 -5.5879354e-09 -4.4703484e-08 4.4703484e-08
-7.4505806e-09 1.0000000 1.8626451e-09 -4.4703484e-08 5.9604645e-08
-7.4505806e-09 -7.4505806e-09 0.99999999 -8.9406967e-08 1.1920929e-07
```



```
-7.4505806e-09 -1.1175871e-08 -7.4505806e-09 0.99999994 1.7881393e-07
-7.4505806e-09 -1.1175871e-08 -4.6566129e-09 -1.1920929e-07 1.0000001
```

As you see the returned matrix is not a identity matrix, therefore the inversion is wrong.

The correct inversion is:

1.0e-08 \*

```
0.0166 -0.0010 -0.0008 0.0277 -0.0228
0.0292 -0.3699 0.0482 0.0017 -0.0105
-0.0221 0.0376 0.0224 -0.0485 0.0154
-0.0100 -0.0207 0.0110 -0.0025 -0.0005
-0.0103 0.0520 -0.0125 0.0092 0.0035
```

Am I doing something wrong?

Best regards,

---

---

Subject: Re: IDL inverse matrix problem??

Posted by [wlandsman](#) on Thu, 22 Jan 2015 20:36:33 GMT

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

---

On Thursday, January 22, 2015 at 2:24:06 PM UTC-5, Amin Farhang wrote:

```
> As you see this time IDL return the inverse BUT this inverse is WRONG!!!
> Let me test the IDL inversion:
>
> IDL> Bm = Am ## Am2
> IDL> print,Bm
> 0.99999999 -1.3038516e-08 -5.5879354e-09 -4.4703484e-08 4.4703484e-08
> -7.4505806e-09 1.0000000 1.8626451e-09 -4.4703484e-08 5.9604645e-08
> -7.4505806e-09 -7.4505806e-09 0.99999999 -8.9406967e-08 1.1920929e-07
> -7.4505806e-09 -1.1175871e-08 -7.4505806e-09 0.99999994 1.7881393e-07
> -7.4505806e-09 -1.1175871e-08 -4.6566129e-09 -1.1920929e-07 1.0000001
>
> As you see the returned matrix is not a identity matrix, therefore the inversion is wrong.
```

Are you sure this is wrong? It looks like diagonal elements [0.99999999, 1.0000000, 0.99999999 ...] are equal to 1 within roundoff error, and the non-diagonal elements are all less 1.2d-7. So within roundoff error this appears to be the identity matrix. --Wayne

---

---

Subject: Re: IDL inverse matrix problem??

Posted by [amin farhang](#) on Thu, 22 Jan 2015 21:00:07 GMT

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

---



Yes, 1.0d-07 in non-diagonal elements is very big for my work and should be 0.

As in my post, I write the correct inversion of matrix, but IDL return below inverse matrix which has a big deviation from the correct answer and cause divergence in my code.

The IDL inversion:

IDL> print,Am2

1.0e-08 \*

0.0045161401	-0.025341567	0.0071848571	-0.051201383	0.061103996
-0.012854534	-0.041131820	-0.014869647	-0.12145509	0.17977632
-0.0012123546	0.022650635	0.010223329	0.094675107	-0.12195776
0.0075355569	0.0047161504	0.010238965	0.073592958	-0.089229866
-0.018737239	-0.014402907	-0.0038020538	-0.064385167	0.089332881

---

Subject: Re: IDL inverse matrix problem??

Posted by [Craig Markwardt](#) on Thu, 22 Jan 2015 21:40:41 GMT

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

---

On Thursday, January 22, 2015 at 2:24:06 PM UTC-5, Amin Farhang wrote:

> Am I doing something wrong?

>

I think you are expecting too much from the numerical precision available. Your matrix has a huge range of singular values (factor of 10 billion), so that's going to cause some undesirable round-off errors. The matrices are correct, to within the round-off error of the solution.

If this is very important to you, I think you need to read a lot more about linear algebra. Computing the inverse directly is usually the worst thing you can do. The standard practice is to factor the matrix instead, and solve your problem indirectly. Numerical Recipes has a good amount of beginner discussion on this. Then you can graduate to Golub and van Loan, /Matrix Computations/. Singular Value Decomposition might tell you more about the structure of your matrix.

Craig

---

Subject: Re: IDL inverse matrix problem??

Posted by [amin farhang](#) on Fri, 23 Jan 2015 06:58:33 GMT

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

---

Dear Craig,

Since my original matrix, which is a positive-definite 1000x1000 matrix, before sending my post, I



try to solve the linear equations using Cholesky decomposition, but again IDL could not handle the precision. On the other hand since IDL is widely used in Astronomy and you know the values in Astronomy are huge, I think it is very important to be sure that IDL is reliable for this kind of computations. Anyway thanks and I try to re-write my code in Python or matlab.

Best,

---

---

Subject: Re: IDL inverse matrix problem??

Posted by [chris\\_torrence@NOSPAM](#) on Fri, 23 Jan 2015 15:25:49 GMT

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

---

On Thursday, January 22, 2015 at 11:58:37 PM UTC-7, Amin Farhang wrote:

> Dear Craig,

>

> Since my original matrix, which is a positive-definite 1000x1000 matrix, before sending my post, I try to solve the linear equations using Cholesky decomposition, but again IDL could not handle the precision. On the other hand since IDL is widely used in Astronomy and you know the values in Astronomy are huge, I think it is very important to be sure that IDL is reliable for this kind of computations. Anyway thanks and I try to re-write my code in Python or matlab.

>

> Best,

Hi Amin,

I don't think this is a problem with IDL. You will have a problem with *any* scientific software package that uses IEEE arithmetic, including Python and Matlab. The only package which could have a hope of doing this is a symbolic language like Mathematica or Maple.

You really need to either scale your data to a more reasonable range, or use different linear algebra techniques, such as Craig suggests.

Cheers,

Chris

IDL Project Lead

Exelis

---

---

Subject: Re: IDL inverse matrix problem??

Posted by [amin farhang](#) on Fri, 23 Jan 2015 18:32:41 GMT

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

---

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.

Best,

---

---

Subject: Re: IDL inverse matrix problem??

Posted by [Craig Markwardt](#) on Fri, 23 Jan 2015 18:32:53 GMT

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

---

On Friday, January 23, 2015 at 1:58:37 AM UTC-5, Amin Farhang wrote:

> Dear Craig,

>

> Since my original matrix, which is a positive-definite 1000x1000 matrix, before sending my post, I try to solve the linear equations using Cholesky decomposition, but again IDL could not handle the precision. On the other hand since IDL is widely used in Astronomy and you know the values in Astronomy are huge, I think it is very important to be sure that IDL is reliable for this kind of computations. Anyway thanks and I try to re-write my code in Python or matlab.

Yes, please let us know how it turns out with Python or Matlab. By the way, you were using LA\_INVERT, which is from the LAPACK package. This is one of the premier linear algebra package for computers. If LAPACK is having trouble, what does that say about your problem?

Craig

---

---

Subject: Re: IDL inverse matrix problem??

Posted by [chris\\_torrence@NOSPAM](#) on Fri, 23 Jan 2015 18:56:54 GMT

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

---

On Friday, January 23, 2015 at 11:32:44 AM UTC-7, 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.

>

> Best,

Amin,

I just tried your original matrix, and it didn't work in Python. For example, here's IDL:

```
IDL> m1 = 9.5d15 + 0.5d18*randomu(1,1,5,/double)
```

```
IDL> m2 = -1d0 + 2d0*randomu(2,5,1,/double)
```

```
IDL> Am = m1 ## m2
```

```
IDL> result = LA_INVERT(Am)
```



% LA\_INVERT: Singular matrix encountered, STATUS=5.  
% Execution halted at: \$MAIN\$

Now let's try the Python numpy.linalg package:

```
IDL> linalg = Python.Import('numpy.linalg')
```

```
IDL> result = linalg.inv(Am)
```

% PYTHON\_CALLMETHOD: Exception: Singular matrix.

% Execution halted at: \$MAIN\$

Could you please post an example of a small matrix that works in Python but not in IDL?

Thanks!

-Chris

---

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  $\sim 1e-8$  errors and double precision is capable of precision  $2e-16$ , which implies a loss of precision of  $(1e-8/2e-16) \sim 8$  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

---

---



Hi,

Let me try an example in IDL and matlab for example:

In IDL:

```
IDL> Am = [[-6.8599934d+16,-8.3590379d+16,-1.8742199d+17,1.5697952d+17,
1.1607144d+17],$
IDL> [-1.2228319d+17,-1.4900449d+17,-3.3409009d+17,2.7982470d+17, 2.0690379d+17],$
IDL> [-1.4972730d+17,-1.8244569d+17,-4.0907020d+17,3.4262599d+17, 2.5333938d+17],$
IDL> [-1.0612015d+17,-1.2930950d+17,-2.8993102d+17,2.4283828d+17, 1.7955584d+17],$
IDL> [-1.1696076d+17,-1.4251900d+17,-3.1954867d+17,2.6764521d+17, 1.9789821d+17]]
IDL> Bm = la_invert(Am)
IDL> Cm = Am ## Bm
IDL> print,Cm
    0.99999999 -3.7252903e-09 -1.2107193e-08 -7.4505806e-08 8.9406967e-08
-1.4901161e-08 1.00000000 -1.2107193e-08 -1.0430813e-07 1.1920929e-07
-1.4901161e-08 -7.4505806e-09 0.99999997 -1.4901161e-07 8.9406967e-08
-1.4901161e-08 -7.4505806e-09 -1.4901161e-08 0.99999990 8.9406967e-08
-1.4901161e-08 -2.6077032e-08 -2.0489097e-08 -1.4901161e-07 1.00000002
```

As you see the AA-1 is not a complete identity matrix

In matlab:

```
Am = [ [-6.8599934e+16 , -8.3590379e+16 , -1.8742199e+17 , 1.5697952e+17 ,
1.1607144e+17];...
[-1.2228319e+17 , -1.4900449e+17 , -3.3409009e+17 , 2.7982470e+17 , 2.0690379e+17];...
[-1.4972730e+17 , -1.8244569e+17 , -4.0907020e+17 , 3.4262599e+17 , 2.5333938e+17];...
[-1.0612015e+17 , -1.2930950e+17 , -2.8993102e+17 , 2.4283828e+17 , 1.7955584e+17];...
[-1.1696076e+17 , -1.4251900e+17 , -3.1954867e+17 , 2.6764521e+17 , 1.9789821e+17]];
```

```
Bm = inv(Am);
Cm = Am * Bm
```

```
1.0000 -0.0000 -0.0000 -0.0000 0.0000
0 1.0000 -0.0000 0.0000 -0.0000
-0.0000 -0.0000 1.0000 0 0
-0.0000 0.0000 -0.0000 1.0000 0.0000
-0.0000 0 -0.0000 -0.0000 1.0000
```

As you see the result is a identity matrix



Best,

---

---

Subject: Re: IDL inverse matrix problem??

Posted by [amin farhang](#) on Fri, 23 Jan 2015 20:00:33 GMT

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

---

Dear Craig,

Thanks, you are right I should not directly compute the inversion. I'm trying to use numerical algorithm for this work.

Best,

---

---

Subject: Re: IDL inverse matrix problem??

Posted by [chris\\_torrence@NOSPAM](#) on Fri, 23 Jan 2015 20:06:05 GMT

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

---

On Friday, January 23, 2015 at 12:56:35 PM UTC-7, Amin Farhang wrote:

> Hi,

>

> Let me try an example in IDL and matlab for example:

>

>

> In IDL:

>

> IDL> Am =[-6.8599934d+16,-8.3590379d+16,-1.8742199d+17,1.5697952d+17,  
1.1607144d+17],\$

> IDL> [-1.2228319d+17,-1.4900449d+17,-3.3409009d+17,2.7982470d+17, 2.0690379d+17],\$

> IDL> [-1.4972730d+17,-1.8244569d+17,-4.0907020d+17,3.4262599d+17, 2.5333938d+17],\$

> IDL> [-1.0612015d+17,-1.2930950d+17,-2.8993102d+17,2.4283828d+17, 1.7955584d+17],\$

> IDL> [-1.1696076d+17,-1.4251900d+17,-3.1954867d+17,2.6764521d+17, 1.9789821d+17]]

> IDL> Bm = la\_invert(Am)

> IDL> Cm = Am ## Bm

> IDL> print,Cm

> 0.99999999 -3.7252903e-09 -1.2107193e-08 -7.4505806e-08 8.9406967e-08

> -1.4901161e-08 1.0000000 -1.2107193e-08 -1.0430813e-07 1.1920929e-07

> -1.4901161e-08 -7.4505806e-09 0.99999997 -1.4901161e-07 8.9406967e-08

> -1.4901161e-08 -7.4505806e-09 -1.4901161e-08 0.99999990 8.9406967e-08

> -1.4901161e-08 -2.6077032e-08 -2.0489097e-08 -1.4901161e-07 1.0000002

>

> As you see the AA-1 is not a complete identity matrix

>

>



```

> In matlab:
>
> Am = [ [-6.8599934e+16 , -8.3590379e+16 , -1.8742199e+17 , 1.5697952e+17 ,
1.1607144e+17];...
> [-1.2228319e+17 , -1.4900449e+17 , -3.3409009e+17 , 2.7982470e+17 , 2.0690379e+17];...
> [-1.4972730e+17 , -1.8244569e+17 , -4.0907020e+17 , 3.4262599e+17 , 2.5333938e+17];...
> [-1.0612015e+17 , -1.2930950e+17 , -2.8993102e+17 , 2.4283828e+17 , 1.7955584e+17];...
> [-1.1696076e+17 , -1.4251900e+17 , -3.1954867e+17 , 2.6764521e+17 , 1.9789821e+17]];
>
> Bm = inv(Am);
> Cm = Am * Bm
>
> 1.0000 -0.0000 -0.0000 -0.0000 0.0000
> 0 1.0000 -0.0000 0.0000 -0.0000
> -0.0000 -0.0000 1.0000 0 0
> -0.0000 0.0000 -0.0000 1.0000 0.0000
> -0.0000 0 -0.0000 -0.0000 1.0000
>
> As you see the result is a identity matrix
>
>
> Best,

```

Hi Amin,

I don't think that last example is actually the identity matrix. I would bet that Matlab is only printing out a few digits, so it just looks like the identity matrix. What happens if you print out all of the digits using a Matlab command like `fprintf('%13.9f\n', Cm)`?

-Chris

---

Subject: Re: IDL inverse matrix problem??  
 Posted by [amin farhang](#) on Fri, 23 Jan 2015 20:06:17 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Ahhh sorry matlab round the output and the original AA-1 is similar to `la_invert` in IDL. Yes i should use numerical methods for compute the inversion.

Thanks,

---

Subject: Re: IDL inverse matrix problem??  
 Posted by [wlandsman](#) on Fri, 23 Jan 2015 20:08:19 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---



On Friday, January 23, 2015 at 2:56:35 PM UTC-5, Amin Farhang wrote:

The MATLAB result *\*looks\** right because you aren't printing many decimal places.

Try this with the IDL result

```
IDL> print,CM,format='(5f10.4)'
```

```
1.0000 -0.0000 -0.0000 -0.0000 0.0000
-0.0000 1.0000 -0.0000 -0.0000 0.0000
-0.0000 -0.0000 1.0000 -0.0000 0.0000
-0.0000 -0.0000 -0.0000 1.0000 0.0000
-0.0000 -0.0000 -0.0000 -0.0000 1.0000
```

but both MATLAB and IDL likely store the numbers the same way (IEEE arithmetic) internally.

---

---

Subject: Re: IDL inverse matrix problem??

Posted by [chris\\_torrence@NOSPAM](#) on Fri, 23 Jan 2015 20:28:34 GMT

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

---

On Friday, January 23, 2015 at 1:06:19 PM UTC-7, Amin Farhang wrote:

> Ahhh sorry matlab round the output and the original AA-1 is similar to la\_invert in IDL. Yes i should use numerical methods for compute the inversion.

>

>

> Thanks,

Whew! Okay, balance is restored to the universe. Thanks Amin for checking. Good luck with your research.

Cheers,  
Chris

---

---

Subject: Re: IDL inverse matrix problem??

Posted by [David Fanning](#) on Fri, 23 Jan 2015 20:35:56 GMT

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

---

Chris Torrence writes:

> Whew! Okay, balance is restored to the universe. Thanks Amin for checking. Good luck with your research.

If I get some time I'll add this to the Sky is Falling annals. ;-)

Cheers,



David

--

David Fanning, Ph.D.

Fanning Software Consulting, Inc.

Coyote's Guide to IDL Programming: <http://www.idlcoyote.com/>

Sepore ma de ni thue. ("Perhaps thou speakest truth.")

---