
Subject: More Kriging Problems

Posted by [David Fanning](#) on Thu, 17 Oct 2013 01:42:26 GMT

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

Folks,

Sorry to keep bringing bad news about the new fast kriging algorithm Chris provided last week, but I don't think it is working correctly.

Aside from problems with the model functions, I think there is something wrong with the way regular grids are processed with the function.

I have been trying to use the Krig2D function to "repair" small holes in larger arrays. I've been using an 11x11 array for this process, but I am getting very strange results.

I've made the programs I am using available here:

http://www.idlcoyote.com/misc/krig_problem.zip

There are three PNG files in the zip file. One of the original data. One using the current Krig2D method in IDL 8.2.3 (which is basically identical to the original data set, as I would expect), and one using my cgKrig2D routine (included in the zip file), which produces a surface not at all like the original data. I *try* to create a surface with the fast kriging routine Chris gave us last week, but I can't get this program to run without an error. I have renamed the functions in this program, which I call krig2d_fast and include in the zip file, so there is no confusion with the original Krig2D routine.

The fact that krig2d_fast doesn't run confuses me, because as far as I can determine my cgKrig2D program is a faithful copy of this routine. The only thing I *believe* to be different about it is that I have modified the model functions to produce what I think are the correct results.

In any case, I can't get it to run with my simple data set. Here is the main-level program I am using to investigate the problem. I recreate the values array each time, because I also suspect these programs are modifying the variable, but I haven't had time to chase that down and confirm it.

I produce the PNG files from the first three graphics commands.

```
; Original data.  
values = Dist(11)  
smdims = Size(values, /Dimension)  
cgSurface, values, Title='Original Values'
```

```
; Surface from current Krig2D program.
values = Dist(11)
sampled1 = Krig2d(values, findgen(smdims[0]), $
    findgen(smdims[1]), /Regular, NX=11, NY=11, Spherical=[5.0, 0.0])
cgSurface, sampled1, Title='Original Krig2D Values'
```

```
; Surface from my cgKrig2D program.
values = Dist(11)
sampled2 = cgKrig2d(values, findgen(smdims[0]), $
    findgen(smdims[1]), /Regular, Spherical=[5.0, 0.0])
cgSurface, sampled2, Title='cgKrig2D Values'
```

```
; Surface from the fast krig2d program of Chris Torrence, offered
; last week on newsgroup. This fails with error.
values = Dist(11)
sampled3 = Krig2d_Fast(values, findgen(smdims[0]), $
    findgen(smdims[1]), /Regular, Spherical=[5.0, 0.0])
cgSurface, sampled3, Title='Fast Krig2D Values'
END
```

I spent the whole day on this, and I'm feeling frustrated. Any ideas gratefully accepted. :-)

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.")

Subject: Re: More Kriging Problems
Posted by chris_torrence@NOSPAM on Thu, 17 Oct 2013 04:22:31 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi David,

I think I see the problem. In my latest krig2d.pro code, I am incorrectly calculating the X and Y arrays for the /REGULAR grid case. Near the end of the code are the following lines:

```
xx = (SIZE(x, /N_DIM) eq 2) ? x : REBIN(TRANSPPOSE(x), nx, n)
yy = (SIZE(y, /N_DIM) eq 2) ? y : REBIN(TRANSPPOSE(y), nx, n)
```

They should be:

```
xx = REBIN(REFORM(x, 1, n), nx, n)
```

```
yy = REBIN(REFORM(y, 1, n), nx, n)
```

Do you want to make the changes to the code, and see what kind of results you get? Also, I would be curious as to your speed test results. I tried your cgkrig2d code, and it seemed about the same or maybe slightly slower. Finally, the cgkrig2d results didn't quite match what I was expecting.

Here is the code I used:

```
values = Dist(11)
smdims = Size(values, /Dimension)
sampled2 = cgKrig2d(values, findgen(smdims[0]), $
  findgen(smdims[1]), /Regular, Spherical=[5.0, 0.0])
sampled3 = Krig2d(values, findgen(smdims[0]), $
  findgen(smdims[1]), /Regular, Spherical=[5.0, 0.0])
s2 = surface(sampled2, window_title='cgKrig2D')
s3 = surface(sampled3, window_title='new Krig2D')
```

Note that the call to Krig2D is my latest fast code with the 2-line fix mentioned above.

Thanks for catching this!

-Chris

VIS

Subject: Re: More Kriging Problems

Posted by [David Fanning](#) on Thu, 17 Oct 2013 04:25:12 GMT

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

Chris Torrence writes:

> I think I see the problem. In my latest krig2d.pro code, I am incorrectly calculating the X and Y arrays for the /REGULAR grid case. Near the end of the code are the following lines:

> xx = (SIZE(x, /N_DIM) eq 2) ? x : REBIN(TRANSPPOSE(x), nx, n)

> yy = (SIZE(y, /N_DIM) eq 2) ? y : REBIN(TRANSPPOSE(y), nx, n)

>

> They should be:

> xx = REBIN(REFORM(x, 1, n), nx, n)

> yy = REBIN(REFORM(y, 1, n), nx, n)

>

> Do you want to make the changes to the code, and see what kind of results you get? Also, I would be curious as to your speed test results. I tried your cgkrig2d code, and it seemed about the same or maybe slightly slower. Finally, the cgkrig2d results didn't quite match what I was expecting. Here is the code I used:

> values = Dist(11)

> smdims = Size(values, /Dimension)

> sampled2 = cgKrig2d(values, findgen(smdims[0]), \$

> findgen(smdims[1]), /Regular, Spherical=[5.0, 0.0])

> sampled3 = Krig2d(values, findgen(smdims[0]), \$

> findgen(smdims[1]), /Regular, Spherical=[5.0, 0.0])

> s2 = surface(sampled2, window_title='cgKrig2D')

> s3 = surface(sampled3, window_title='new Krig2D')
>
> Note that the call to Krig2D is my latest fast code with the 2-line fix mentioned above.

Thanks, Chris. I'm bushed, but I'm notorious for not being a multi-tasker. I'm sure I'll be back at this first thing in the morning. I'll let you know what I find out. :-)

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.")

Subject: Re: More Kriging Problems
Posted by [David Fanning](#) on Thu, 17 Oct 2013 13:01:24 GMT
[View Forum Message](#) <> [Reply to Message](#)

Chris Torrence writes:

> I think I see the problem. In my latest krig2d.pro code, I am
incorrectly calculating the X and Y arrays for the /REGULAR grid case.
Near the end of the code are the following lines:
> xx = (SIZE(x, /N_DIM) eq 2) ? x : REBIN(TRANSPPOSE(x), nx, n)
> yy = (SIZE(y, /N_DIM) eq 2) ? y : REBIN(TRANSPPOSE(y), nx, n)
>
> They should be:
> xx = REBIN(REFORM(x, 1, n), nx, n)
> yy = REBIN(REFORM(y, 1, n), nx, n)

Yes, this appears to solve the problem nicely. Thanks!

> Do you want to make the changes to the code, and see what kind of
> results you get?

OK, I made the above changes to my cgKrig2D code and to your new Krig2D code, which I call Krig2D_Fast, to distinguish it from the normal Krig2D code.

Here is my test code:

```
.*****  
,
```

```

number = 21
values = Dist(number)
cgSurface, values, Title='Original Values'

values = Dist(number)
tic
sampled1 = Krig2d(values, Spherical=[5.0, 0.0])
toc
cgSurface, sampled1, Title='Original Krig2D Values'

values = Dist(number)
tic
sampled2 = cgKrig2d(values, Spherical=[5.0, 0.0])
toc
cgSurface, sampled2, Title='cgKrig2D Values'

values = Dist(number)
tic
sampled3 = Krig2d_Fast(values, Spherical=[5.0, 0.0])
toc
cgSurface, sampled3, Title='Fast Krig2D Values'
.*****
,

```

Here are the times I get on my machine:

```

Elapsed Time: 0.952000 ; Original Krig2D code
Elapsed Time: 0.187000 ; cgKrig2D
Elapsed Time: 0.202000 ; Krig2D_Fast

```

I've run this several times and cgKrig2D is always *slightly* faster than Krig2D_Fast, but essentially the same, even if I set the "number" variable to different values.

There are a couple of other differences (which may explain the time difference). cgKrig2D preserves the grid size of the original data set, while the original Krig2D code and the Krig2D_Fast code default to their regular 26x26 size grid. (Humm, if this is true, then if I make the grid 31x31 then cgKrig2D should be slower than Krig2D_Fast, and this in fact happens.)

```

Elapsed Time: 5.616000 ; Original Krig2D code
Elapsed Time: 2.137000 ; cgKrig2D
Elapsed Time: 2.106000 ; Krig2D_Fast

```

Also, now the original Krig2D routine produces a CRAZY surface! Not at all like the original surface. This wasn't happening last night, but you know how things get when it is late in the day and you have been working 12 hours straight trying to make sense of things. All kinds of

strangeness sets in. Anyway, based on today's tests, I wouldn't use the original Krig2D routine.

In terms of fidelity to the original data, here are the results I get when I use the 21x21 data set. I still don't totally understand the difference between a semivariogram and a covariance matrix, but I guess cgKrig2D is using the former and Krig2D_Fast is using the latter.

```
MinMax: 0.000000 14.1421 ; Original data
MinMax: -2.16058 15.0750 ; Original Krig2D code
MinMax: -2.38419e-006 14.1422 ; cgKrig2D
MinMax: -7.15256e-006 14.2410 ; Krig2D_Fast
```

I did add other kriging models to cgKrig2D, but I couldn't make sense of them yesterday. I'll look at this again today in light of the changes we have made to the code.

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.")

Subject: Re: More Kriging Problems

Posted by [David Fanning](#) on Thu, 17 Oct 2013 13:24:36 GMT

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

David Fanning writes:

```
> There are a couple of other differences (which may explain the time
> difference). cgKrig2D preserves the grid size of the original data set,
> while the original Krig2D code and the Krig2D_Fast code default to their
> regular 26x26 size grid. (Humm, if this is true, then if I make the grid
> 31x31 then cgKrig2D should be slower than Krig2D_Fast, and this in fact
> happens.)
>
> Elapsed Time: 5.616000 ; Original Krig2D code
> Elapsed Time: 2.137000 ; cgKrig2D
> Elapsed Time: 2.106000 ; Krig2D_Fast
>
> Also, now the original Krig2D routine produces a CRAZY surface! Not at
> all like the original surface. This wasn't happening last night, but you
```

- > know how things get when it is late in the day and you have been working
- > 12 hours straight trying to make sense of things. All kinds of
- > strangeness sets in. Anyway, based on today's tests, I wouldn't use the
- > original Krig2D routine.

Well, HERE is an interesting thing!

Just for the hell of it, I changed the number in my test program to 26, to make a 26x26 array. I wanted to see if cgKrig2D and Krig2D_Fast would be exactly the same speed in that case. In fact, they are, pretty much:

Elapsed Time: 2.482000 ; Original Krig2D code

Elapsed Time: 0.708000 ; cgKrig2D

Elapsed Time: 0.710000 ; Krig2D_Fast

But, the interesting thing here is that NOW the surface plot of the original Krig2D plot is NORMAL!!!! Not crazy at all. Apparently, this code only goes crazy if there is a mismatch between the input data size and the output data size. Who woulda known! ;-)

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.")
