

---

Subject: FFT with NaNs in an array  
Posted by [adisn123](#) on Thu, 20 Jul 2006 18:00:59 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Hi,

I'm trying to fourier transform a spacial domain image to frequency domain using FFT function in IDL.

My image has quite a bit of NaNs in an array, about 5%.

When I use FFT into the image, it doesn't give me any errors, but when I inversely fourier transform after

filtering, it gives a little funky result.

How do I make FFT ignore NaNs in their job or filtering?

---

---

Subject: Re: FFT with NaNs in an array  
Posted by [Kenneth P. Bowman](#) on Thu, 20 Jul 2006 19:24:13 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

In article <1153418459.663699.178520@i42g2000cwa.googlegroups.com>, [adisn123@yahoo.com](#) wrote:

> Hi,  
>  
> I'm trying to fourier transform a spacial domain image to frequency  
> domain using FFT function in IDL.  
>  
> My image has quite a bit of NaNs in an array, about 5%.  
>  
> When I use FFT into the image, it doesn't give me any errors, but when  
> I inversely fourier transform after  
>  
> filtering, it gives a little funky result.  
>  
> How do I make FFT ignore NaNs in their job or filtering?

FFTs assume no missing data. Missing data destroys the symmetry that makes the "Fast" part of FFT work.

You haven't indicated your purpose, but basically, your options are:

1) Interpolate to fill the gaps before FFTing. If your gaps are irregular, this can be tedious; and you should do some tests with known data to evaluate the effects of the interpolation. For example, take a

complete image with properties similar to your data and delete 5% of the data. Interpolate the gaps and then compare the FFTs of the original and interpolated images.

2) Use a curve-fitting routine, such as REGRESS, to fit sines and cosines. (Least-squares fitting with sines and cosines is equivalent to the FFT when there is no missing data.) This is much slower than an FFT, but if your data size is not too large, you may not notice or care. If you know ahead of time that you only want to keep a few frequencies, then this could be as fast as an FFT.

Ken Bowman

---

---

Subject: Re: FFT with NaNs in an array  
Posted by [Steve Eddins](#) on Thu, 20 Jul 2006 19:42:30 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

adisn123@yahoo.com wrote:

> Hi,  
>  
> I'm trying to fourier transform a spacial domain image to frequency  
> domain using FFT function in IDL.  
>  
> My image has quite a bit of NaNs in an array, about 5%.  
>  
> When I use FFT into the image, it doesn't give me any errors, but when  
> I inversely fourier transform after  
>  
> filtering, it gives a little funky result.

I would have expected you to get a VERY funky result. Since every output element of an FFT depends on every input element, I'd expect every output element of your result to be NaN.

> How do I make FFT ignore NaNs in their job or filtering?

I think you'll need to explicitly replace the NaNs with 0s, like this:

```
A(isnan(A)) = 0;
```

--

Steve Eddins  
<http://blogs.mathworks.com/steve>

---

---

Subject: Re: FFT with NaNs in an array

Posted by [Steve Eddins](#) on Thu, 20 Jul 2006 19:44:46 GMT

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

---

Steve Eddins wrote:

> adisn123@yahoo.com wrote:

>> Hi,

>>

>> I'm trying to fourier transform a spacial domain image to frequency

>> domain using FFT function in IDL.

>>

>> My image has quite a bit of NaNs in an array, about 5%.

>>

>> When I use FFT into the image, it doesn't give me any errors, but when

>> I inversely fourier transform after

>>

>> filtering, it gives a little funky result.

>

> I would have expected you to get a VERY funky result. Since every  
> output element of an FFT depends on every input element, I'd expect  
> every output element of your result to be NaN.

>

>> How do I make FFT ignore NaNs in their job or filtering?

>

> I think you'll need to explicitly replace the NaNs with 0s, like this:

>

> A(isnan(A)) = 0;

Whoops, forgot which newsgroup I was in, sorry. The above code line is  
MATLAB syntax. Replace it with suitable IDL syntax.

--

Steve Eddins

<http://blogs.mathworks.com/steve>

---

---

Subject: Re: FFT with NaNs in an array

Posted by [adisn123](#) on Fri, 21 Jul 2006 00:08:47 GMT

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

---

^^;

Thanks for the tip.

Steve Eddins wrote:

> Steve Eddins wrote:

>> adisn123@yahoo.com wrote:

>>> Hi,

>>>

>>> I'm trying to fourier transform a spacial domain image to frequency

>>> domain using FFT function in IDL.  
>>>  
>>> My image has quite a bit of NaNs in an array, about 5%.  
>>>  
>>> When I use FFT into the image, it doesn't give me any errors, but when  
>>> I inversely fourier transform after  
>>>  
>>> filtering, it gives a little funky result.  
>>  
>> I would have expected you to get a VERY funky result. Since every  
>> output element of an FFT depends on every input element, I'd expect  
>> every output element of your result to be NaN.  
>>  
>>> How do I make FFT ignore NaNs in their job or filtering?  
>>  
>> I think you'll need to explicitly replace the NaNs with 0s, like this:  
>>  
>> A(isnan(A)) = 0;  
>  
> Whoops, forgot which newsgroup I was in, sorry. The above code line is  
> MATLAB syntax. Replace it with suitable IDL syntax.  
>  
> --  
> Steve Eddins  
> <http://blogs.mathworks.com/steve>

---

---

Subject: Re: FFT with NaNs in an array  
Posted by [Kenneth P. Bowman](#) on Fri, 21 Jul 2006 02:48:56 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

In article <1153439903.349812.299130@75g2000cwc.googlegroups.com>, adisn123@yahoo.com wrote:

> My purpose here is to get rid of low frequency by applyting high pass  
> filter into my image.  
> My image is quite huge (about 10,000 x 10,000 pixel size).  
>  
> Then, if my image has NaNs, what is the reaction of FFT into that?  
> What does FFT consider those NaNs as?  
>  
> When I do FFT, it certainly doesn't give me any errors, then does it  
> mean  
> FFT reads NaNs as some sort of data values?  
>  
> Thanks.

The FFT is a clever algorithm for performing a discrete Fourier

transform. At bottom it amounts to a lot of dot products (additions and multiplications).

The example below demonstrates the effects of having NaNs in the input data. (The IDL FFT function must turn off floating-point error notification, which is an odd thing to do.)

Since your interest is high-pass filtering, you could use a curve-fitting routine to fit a smooth function to the data and then subtract it from the original data. There are least-squares methods and nonlinear approaches, such as Craig Markwardt's MPFIT (<http://cow.physics.wisc.edu/~craigm/idl/>).

Interpolating the gaps will introduce high-frequency components and should probably be avoided.

Ken Bowman

```
n = 8
x = RANDOMN(seed, n)
xt = FFT(x)
xx = FFT(xt, /INVERSE)
PRINT, 'No NaNs'
PRINT, 'x = ', x
PRINT, 'xt = ', xt
PRINT, 'xx = ', FLOAT(xx)
x[5] = !VALUES.F_NAN
xt = FFT(x)
xx = FFT(xt, /INVERSE)
PRINT
PRINT, 'One NaN'
PRINT, 'x = ', x
PRINT, 'xt = ', xt
PRINT, 'xx = ', FLOAT(xx)
```

No NaNs

```
x = -0.303949  1.09403  0.627827  1.23538 -1.09260
-0.563133  0.00560129
  1.14969
xt = ( 0.269105,  0.00000)( 0.237482, -0.231826)(
-0.253748,  0.231771)
( -0.0403187, -0.0762694)( -0.459886, -0.00000)(
-0.0403187,  0.0762694)
( -0.253748, -0.231771)( 0.237482,  0.231826)
xx = -0.303949  1.09403  0.627827  1.23538 -1.09260
```

```
-0.563133 0.00560129
 1.14969
```

One NaN

```
x = -0.303949 1.09403 0.627827 1.23538 -1.09260
```

```
NaN 0.00560129
```

```
 1.14969
```

```
xt = ( NaN, NaN)( NaN, NaN)(
```

```
NaN, NaN)
```

```
( NaN, NaN)( NaN, NaN)(
```

```
NaN, -NaN)
```

```
( NaN, -NaN)( NaN, -NaN)
```

```
xx = NaN NaN NaN NaN NaN
```

```
NaN NaN
```

```
NaN
```

---

Subject: Re: FFT with NANs in an array

Posted by [R](#) on Wed, 26 Jul 2006 22:47:14 GMT

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

---

<adisn123@yahoo.com> wrote in message

news:1153418459.663699.178520@i42g2000cwa.googlegroups.com.. .

> Hi,

>

> I'm trying to fourier transform a spacial domain image to frequency

> domain using FFT function in IDL.

>

> My image has quite a bit of NANs in an array, about 5%.

>

> When I use FFT into the image, it doesn't give me any errors, but when

> I inversely fourier transform after

>

> filtering, it gives a little funky result.

>

> How do I make FFT ignore NANs in their job or filtering?

>

I would interpolate the the image to remove the nans,  
then FTT, apply the filter, inverse fft, and then re-insert all the nans.

Special care needs to be taken when interpolating, depending on  
how many points are missing. If they are single pixels, (i.e. mostly  
surrounded

by data) then a simple bilinear interpolation would be fine. If you have  
large

areas of nan, then you will want to handle that interpolation better.

Also look out for nans at the edge of the image, as the interpolation

might give some extreme funkiness.

Also, take care of the wrap around effects that are inherent in the FFT routines.

Do you have a time series of images? If so, perhaps you can interpolate in time as well as just in space.

Cheers,  
bob

---