Subject: Re: Discrete sine transform
Posted by Dominik[1] on Thu, 02 May 2002 09:38:50 GMT
View Forum Message <> Reply to Message

I am not 100% sure about it, but I think the sin transform is just the
imaginary part of the Fourie Transformation.

Dom

"Matthew Angling" <mjangling@QinetiQ.com> schrieb im Newsbeitrag
news:aaqvkg$85k$1@hamble.qinetiq.com...
> Hello all,
>
> I'm another lurker drawn out to ask for help. I think this is more of
> a maths question rather than IDL, but as I'm working with IDL and this
> seems to be the most friendly newsgroup I know of, I thought I'd try
> here!
>
> Does anybody have the algorithm for the discrete sine tranform coded
> in IDL. If I understand correctly it can be done (with some
> redundency) by using the FFT - but I'm not sure how!
>
> Thanks in advance for your help,
>
> Matt
>
> --
> Dr. M. J. Angling
> Centre for RF Propagation and Atmospheric Research
> QinetiQ Malvern
> mjangling@QinetiQ.com
>
>
>

Subject: Re: Discrete sine transform
Posted by Kenneth P. Bowman on Thu, 02 May 2002 11:16:46 GMT
View Forum Message <> Reply to Message

In article <aar135hqq1@n.ruf.uni-freiburg.de>,
 "Dominik Paul" <dpaul@ukl.uni-freiburg.de> wrote:

> I am not 100% sure about it, but I think the sin transform is just the
> imaginary part of the Fourie Transformation.
>

>> Does anybody have the algorithm for the discrete sine tranform coded

>> in IDL. If I understand correctly it can be done (with some
>> redundency) by using the FFT - but I'm not sure how!

Dominik Paul is correct, but the discrete sine transform is usually defined from (0, pi) rather than (0, 2 pi). So if your series is of length N, I think that you need to double the length of the series by adding N zeros. The coefficients will also need to be adjusted by a factor of 2 (because your are integrating over a domain that is twice as large).

Hope that helps,

Ken Bowman

Subject: Re: Discrete sine transform
Posted by mvukovic on Thu, 02 May 2002 15:36:44 GMT
View Forum Message <> Reply to Message

"Dominik Paul" <dpaul@ukl.uni-freiburg.de> wrote in message news:<aar135hqq1@n.ruf.uni-freiburg.de>...
> I am not 100% sure about it, but I think the sin transform is just the
> imaginary part of the Fourie Transformation.
>
> Dom
>
> "Matthew Angling" <mjangling@QinetiQ.com> schrieb im Newsbeitrag
> news:aaqvkg$85k$1@hamble.qinetiq.com...
>> Hello all,
>>
>> I'm another lurker drawn out to ask for help. I think this is more of
>> a maths question rather than IDL, but as I'm working with IDL and this
>> seems to be the most friendly newsgroup I know of, I thought I'd try
>> here!
>>
>> Does anybody have the algorithm for the discrete sine tranform coded
>> in IDL. If I understand correctly it can be done (with some
>> redundency) by using the FFT - but I'm not sure how!
>>
>> Thanks in advance for your help,
>>
>> Matt
>>
>> --
>> Dr. M. J. Angling
>> Centre for RF Propagation and Atmospheric Research
>> QinetiQ Malvern
>> mjangling@QinetiQ.com

>>
>>
>>

nope! Funny, I was just leafing Numerical Recipes on that topic yesterday. By way of explanation (and probably quoting the Book and other authorities), the sine transform is a ``complete'' transform of the data, i.e., you can get the data back by doing an inverse transform. If you were just to take the imaginary part of your FFT, you would loose half the information, and could not recover it when doing the inverse.

The Book does provide an algorithm to do it using the FFT. You have to make an anti-symmetric series out of your data by extending the data series. The first N samples stay the same. The other N samples are anti-symmetric about N+1 where the N+1 sample must be zero. I've never done it, and advise going to the Book itself to make sure you got everything right. The topic is covered in Ch. 12 of the second edition.

Mirko

Subject: Re: Discrete sine transform
Posted by Matthew Angling on Fri, 03 May 2002 07:41:15 GMT
View Forum Message <> Reply to Message

Mirko Vukovic <mvukovic@taz.telusa.com> wrote in message news:d96c8f7c.0205020736.6eceab0d@posting.google.com...
> "Dominik Paul" <dpaul@ukl.uni-freiburg.de> wrote in message news:<aar135hqq1@n.ruf.uni-freiburg.de>...
>> I am not 100% sure about it, but I think the sin transform is just the
>> imaginary part of the Fourie Transformation.
>>
>> Dom
>>
>> "Matthew Angling" <mjangling@QinetiQ.com> schrieb im Newsbeitrag
>> news:aaqvkg$85k$1@hamble.qinetiq.com...
>>> Hello all,
>>>
>>> I'm another lurker drawn out to ask for help. I think this is more of
>>> a maths question rather than IDL, but as I'm working with IDL and this
>>> seems to be the most friendly newsgroup I know of, I thought I'd try
>>> here!

>>>
>>> Does anybody have the algorithm for the discrete sine tranform
coded
>>> in IDL. If I understand correctly it can be done (with some
>>> redundency) by using the FFT - but I'm not sure how!
>>>
>>> Thanks in advance for your help,
>>>
>>> Matt
>>>
>>> --
>
> nope! Funny, I was just leafing Numerical Recipes on that topic
> yesterday. By way of explanation (and probably quoting the Book and
> other authorities), the sine transform is a ``complete'' transform
of
> the data, i.e., you can get the data back by doing an inverse
> transform. If you were just to take the imaginary part of your FFT,
> you would loose half the information, and could not recover it when
> doing the inverse.
>
> The Book does provide an algorithm to do it using the FFT. You have
> to make an anti-symmetric series out of your data by extending the
> data series. The first N samples stay the same. The other N
samples
> are anti-symmetric about N+1 where the N+1 sample must be zero.
I've
> never done it, and advise going to the Book itself to make sure you
> got everything right. The topic is covered in Ch. 12 of the second
> edition.
>
> Mirko

Hi!

Thanks for everybody's comments. If anybody is interested I have
attached my dst function below. Its a translation of the matlab dst
routine.

Matt

--
Dr. M. J. Angling
Centre for RF Propagation and Atmospheric Research
QinetiQ Malvern

Tel: +44 (0)1684 896460
Fax: +44 (0)1684 894657

email: mjangling@QinetiQ.com

```
Function mldst,a
;function to calculate the discrete sine transform
;translated from the matlab routine dst
;normalisation is split between the forward
;and inverse transforms
;i.e. mldst(mldst(test) is equal to test
;
;NB to take advantage of efficient FFTs the input vector
;should be length 2^m-1

n=n_elements(a)
i=complex(0,1)

;set up antisymmetric vector
y=[0,a,0,-1.*reverse(a)]

;do fft and normalise
yy=2.*sqrt(2*(n+1))*fft(y)
out=yy[1:n]/(-2.*i)

;check type
type=size(out,/type)

;return real part if input was real
if type ne 6 and type ne 9 then out=float(out)

return,out

end
```
