

Changing scales with Fourier transformation [Lesson 3 of matrix algebra (matrix multiplication)]

A.M.C. Davies

Norwich Near Infrared Consultancy, 75 Intwood Road, Cringleford, Norwich NR4 6AA, UK

Tom Fearn

Department of Statistical Science, University College London, Gower Street, London WC1E 6BT, UK

In the last column¹ we showed how we could perform Fourier transformation (FT) of a near infrared (NIR) spectrum in a few lines of matrix algebra and said that in this column we would use it in a novel way. The task we are going to perform is that of changing scales of spectroscopic (NIR) data. This may be novel, we are not aware that anyone else does it this way, but of course instrument manufacturers sometimes like to be silent about the methods they employ.

NIR spectra may be measured by a dispersive or an FT instrument.* Data points in an NIR dispersive spectra are measured at equal intervals of wavelength (λ , nm) while in a spectrum generated by an FT instrument the data points are at equal wavenumber intervals ($\bar{\nu}$, cm^{-1}). The relationship between wavelength and wavenumber is $\lambda = 10^7/\bar{\nu}$. The problem of switching scales is not trivial because of this reciprocal relationship between them; wavenumbers at equally spaced intervals will not be at equal intervals when converted to nanometers. Plotting programs normally assume that the data are spaced at equal intervals and data files will normally contain only an initial wavelength or wavenumber and the interval between data points rather than a specification of every data point. As far as we are aware, scale conversion from wavenumber to wavelength is normally done by fitting a function through data in a small window. The method that we are describing in this article was almost suggested by Fred McClure in the original paper² on the use of FT in NIR data compression. He said "... *In fact the Fourier transform may be searched to compute the magnitude of the computed spectrum at the required wavelengths rather than re-computing the entire spectrum*".

We are going to compute the spectrum at a set of 700 wavenumber values which happen to translate to a series of points at 2 nm intervals in a wavelength scale!

Test spectra

To illustrate the idea we are using spectra provided by our friend and NIR colleague, Professor Heinz Siesler of the University of Essen, Germany. These are of a sample of a mixture of rare earths which was measured on an FT and on a dispersive instrument. We will transform the wavenumber spectrum to wavelength and compare it with the dispersive

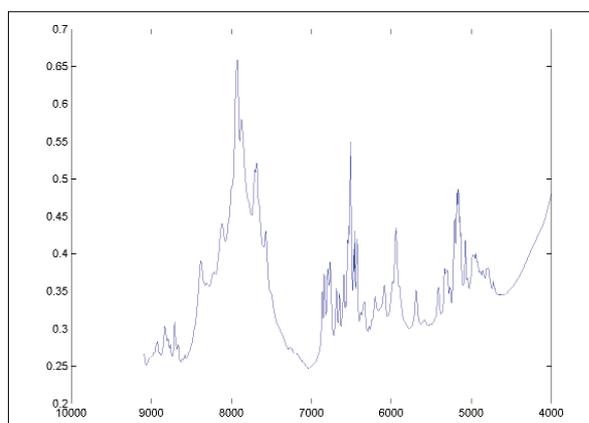


Figure 1. NIR spectrum of a sample of a mixture of rare earths measured on an FT instrument; $\log 1/R$ against wavenumber (cm^{-1}).

spectrum. The initial wavenumber spectrum is shown in Figure 1.

Program

The program required for this scale change is very similar to the TDeious Fourier transform program¹ except in the calculation of the reconstructed spectrum and so we will repeat it with a few additional comments. To assist in understanding the program we include Table 1, which is a list of all the variables.

Instructions are in red while comments are in green. If you have Matlab and wish to try the program, you do not need to type in the code, it is available on the *Spectroscopy Europe* web site at http://www.spectroscopyeurope.com/td_col.html.

```
% TDeiousTwo Fourier Transformation (Scale
% Changer) Converts from wavenumber to
% wavelength
% Get wavenumber spectrum

load HS1;
d=(S_XV(2,1:2640))';

% This spectrum was measured from 9090 wn to
% 3812 wn for a total of 2640 points

dp=2640;
wnmax=9090.89;
wngap=1.92776;
```

*Although they are popularly known as "FT" instruments this is not a very accurate description. The instruments are interferometers, which produces interferograms. The interferograms are converted to spectra using the fast Fourier transform.³ However, we will use the popular terminology.

Table 1. Dimensions and size of all the variables used in TediousTwo.

Name	Size	Name	Size
C	700×1320	npairs	1×1
S	700×1319	p	1×1320
S_XV	2×700	s	1×1
X	2640×2640	t	2640×1
Xi	700×2640	ti	700×1
d	2640×1	w	700×1
dd	700×1	wn	700×1
div	2640×1	wngap	1×1
dm	1×1	wnmax	1×1
dp	1×1	wnmin	1×1
e	2640×1	wnr	1×2640
e1	2640×1	wpt	700×1320
ei	700×1	xfit	700×1
fc	2640×1	xfits	1×700
fcs	1×1	y	2640×1
k	1×1		
n	1×1		
ni	1×1		

```
wnmin=wnmax-(dp-1)*wngap;
wnr=[wnmax:-wngap:wnmin];
figure
plot(wnr',d) % Figure 1
```

```
set(gca,'XDir','reverse') % this reverses the
% scale on the wavenumber axis

% Calculate FT using equations on p186 of
% McClure's article in Handbook of NIRA

% Tilt spectrum to make ends equal
dm=dp-1;

% calculate slope
s=(d(dp)-d(1))/dm;
e=s*[0:dm]';

% adjust data by slope
e1=d-e;
y=e1;

n=size(y,1); % number of data points
k=n/2; % half number of data points
t=[1:n]'; % n x 1 col vector 1,2,...,n
p=[1:k]; % 1 x k row vector 1,2,...,k
wpt=(2*pi/n)*(t*p);

% n x k matrix with ij'th element 2(pi)ij/n
C=cos(wpt); % n x k matrix of cosines
S=sin(wpt(:,1:(k-1)));

% n x k-1 matrix of sines (omit last one
% because it would be a col of zeros anyway)
X=[ones(n,1) C S];
```

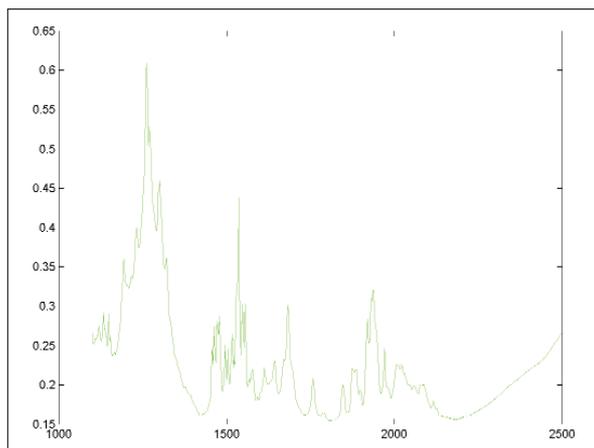


Figure 2. Wavelength spectrum of the rare earth mixture calculated by Fourier transformation from the data in Figure 1; log 1/R against wavelength (nm).

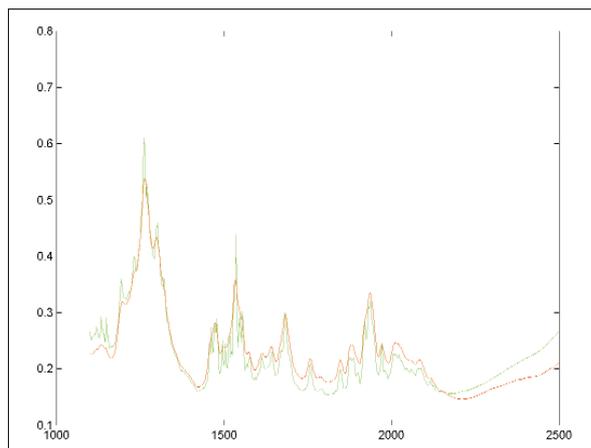


Figure 3. Calculated spectrum (green) compared to the wavelength spectrum (red) measured on a dispersive instrument; log 1/R against wavelength (nm).

```
% put together into n x 2M matrix
div=n*[1 (1/2)*ones(1,k-1) 1 (1/2)*ones(1,k-1)]';

% 2k x 1 col vector of divisors fc are the
% Fourier coefficients order is
% a0,a1,...,aM,b1,...,b(k-1) Fred McClure
% orders his differently

fc=(X'*y)./div;

% This does the summations

% THIS IS THE NEW SECTION
% now we invert the transform, not at the
% equally spaced wavenumber points we started
% from, but at equally spaced wavelength
% points this will involve some interpolation,
% so we have to smooth a little by omitting
% the very high order coefficients

npairs=330; % Number of pairs of Fourier
% coefficients to use when inverting

fcs=npairs+1;

% make unrequired coefficients 0
fc(fcs:k)=0;
fc(k+fcs:n)=0;

% set up vector with points at which to invert
w=[1100:2:2498]';

% convert this to wavenumbers
wn=(1e7)./w;

% now convert these to a scale from 1 to 2640
ti=1+(wnmax-wn)/wngap;

% now create an X matrix just like the one
% used for the FT, but with different t. i.e.
% with sines and cosines calculated at the
% points at which we wish to invert

ni=size(ti,1);
wpt=(2*pi/n)*(ti*p);
C=cos(wpt);
S=sin(wpt(:,1:(k-1)));
Xi=[ones(ni,1) C S];

% now invert, but using Xi instead of X
```

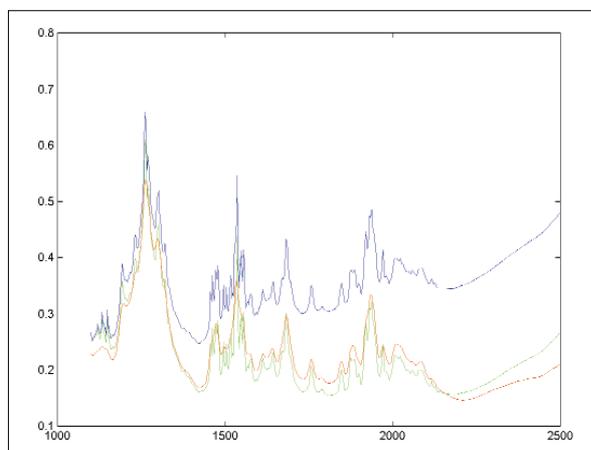


Figure 4. As Figure 3 with the tilt correction applied to the recalculated spectrum (blue); log 1/R against wavelength (nm).

```
xfit=Xi*fc; % THIS IS THE CRUCIAL LINE OF
% MATRIX MULTIPLICATION!!

plot(w,xfit,'g'); % Figure 2

hold on

% get dispersive spectrum and plot for
% comparison

load HS2;
dd=(S_XV(2,:))';

plot(w,dd,'r'); % Figure 3

pause

% Un-tilt the spectrum (not really required in
% this case differences due to technology;
% dispersive spectrum is almost flat) be
% careful, need to convert to 1-2640 scale on
% which linear baseline was subtracted

ei=s*(ti-1);
xfits=xfit'+ei';

figure
plot(w,xfits); % Figure 4

hold off
```

Discussion

The peaks in the switched spectrum appear to line up with the peaks in the dispersive spectrum while retaining much of the high-resolution information seen in the wavenumber version. There is no obvious way of determining the number of pairs of Fourier coefficients to retain but 330 is a quarter of the original number and should contain

transformation but the switch to wavelengths has got to use the TDeious method.

Matrix multiplication

This article has not introduced any new matrix algebra but (after feed-back from the previous article) we would

like to make quite sure that everyone understands what happens in matrix multiplication, see Box 1. For matrix multiplications the number of columns in the first matrix must be equal to the number of rows in the second. In matrix **A** the background is white and the numbers are colour coded by row, while in matrix **B** the numbers are white and the background is colour coded by columns. This should enable you to see where all the products come from before they are summed in the appropriate element of the new matrix **C**. Note that in matrix algebra, $\mathbf{A} \times \mathbf{B}$ is not the same as $\mathbf{B} \times \mathbf{A}$. In the program, as it stands at the moment, you will not find an example of the multiplication of two matrices. In the "Crucial line" $\mathbf{f}\mathbf{c}$ is a column vector

A × **B**

6	7
3	5
2	1

1	4	6
2	8	9

C

=	$(6 \times 1) + (7 \times 2)$	$(6 \times 4) + (7 \times 8)$	$(6 \times 6) + (7 \times 9)$
	$(3 \times 1) + (5 \times 2)$	$(3 \times 4) + (5 \times 8)$	$(3 \times 6) + (5 \times 9)$
	$(2 \times 1) + (1 \times 2)$	$(2 \times 4) + (1 \times 8)$	$(2 \times 6) + (1 \times 9)$

C

=	20	80	99
	13	52	63
	4	16	21

B × **A**

1	4	6
2	8	9

6	7
3	5
2	1

D

=	$(1 \times 6) + (4 \times 3) + (6 \times 2)$	$(1 \times 7) + (4 \times 5) + (6 \times 1)$
	$(2 \times 6) + (8 \times 3) + (9 \times 2)$	$(2 \times 7) + (8 \times 5) + (9 \times 1)$

D

=	30	33
	54	63

Box 1.

almost all of the useful information without running too much of a risk of interpolating between points in high-frequency waves (which could give rise to large errors). This example was deliberately used because it contains some sharp peaks, due to electronic absorptions, which are resolved by the higher-resolution (FT) instrument. Absorptions due to molecular vibrations in the NIR region are usually much broader and the resolution difference between the different systems is much less obvious. The program, with slight modifications, could be used to convert wavelength spectra to wavenumber spectra but it is not very likely that this would ever be required. The common format is wavelength and converting from wavelength to wavenumber by any method cannot "discover" the higher resolution which is normally available in FT systems. We are not sure if there are any advantages of this FT method, which was produced for this demonstration. As written it is quite slow; this could be speeded by using the FFT for the first

transform because we have only one spectrum. If we were considering several spectra then $\mathbf{f}\mathbf{c}$ would become a matrix *without any change in the code!* This is one of the great advantages of well-constructed matrix algebra coding.

The next column

This is not the end of matrix algebra! However, the first (chemometric) TD column of 2001 will be concerned with another topic. We will return to matrix algebra later in the year.

Acknowledgement

We are very grateful to Heinz Siesler for allowing the use of his test spectra.

References

1. A.M.C. Davies and T. Fearn, *Spectrosc. Europe* **12**(4), 28 (2000).
2. F.G. Giesbrecht, W.F. McClure and A. Hamid, *Appl. Spectrosc.* **35**, 210 (1981).
3. J.W. Cooley and J.W. Tukey, *Maths. Comput.* **19**, 297 (1965).