Four Forms of the Fourier Transform
— for Freshmen, using MATLAB

Frederik J Simons

Adam C. Maloof

Princeton University

Explain the basics — I

1. `help`
2. `lookfor`
3. `type`
4. `who, whos, which`
7. `diary`

Explain the basics — II

7. `plot`
8. `xlabel, ylabel, title`
11. `hold on, hold off`
13. `sprintf`
14. `print`
15. `load, imread`

Make a cheat sheet

Name	Function	Example
`imread`	Reads an image file	`ix=imread('filename');`
`fullfile`	Constructs a valid path name	`ff=fullfile('dirn','fname');`
`size`	Queries the size of a variable	`s=size(ix);`
`plot`	Plots (x, y) values on a graph	`x=[1 2 3]; y=[10 20 30];` `plot(x,y,'o')`
`xlabel`	Uses a quoted string for an x-axis label	`xlabel('elevation [m]')`
`ylabel`	Uses a quoted string for a y-axis label	`xlabel('roughness')`

`hold on`	Keeps current axes for next time you plot anything	`x=[1 2 pi]; y=[10 20 30];`
		`plot(x,y,'bo'); hold on;`
		`plot(10*x,3*y,'rs')`

`linspace`	Makes an array of N evenly spaced values between a and b	`x=linspace(-3,3,100)`
`reshape`	Changes the dimensions of an array x to a rows and b columns	`x=linspace(-3,3,100);` `xr=reshape(x,20,5)`
`hist`	Makes a histogram (and plots it)	`x=linspace(-3,3,10);` `hist(x)`
`bar`		
`axis xy`		
`axis ij`		

Explain the basics — III

Addressing:

rows, columns, dimensions, range

17. `size`

18. `transpose`

19. `colon`

20. `linspace`

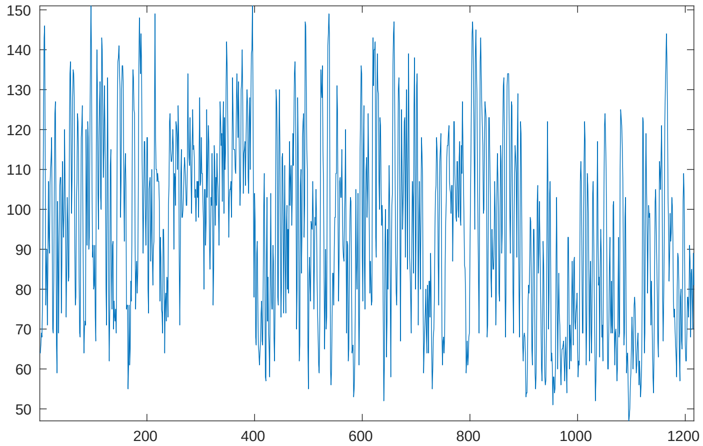
Logic:

logical, character, string, double

21. `<`, `>`, `==`, $\sim$, `&`, `|`

Practice on the *command line*

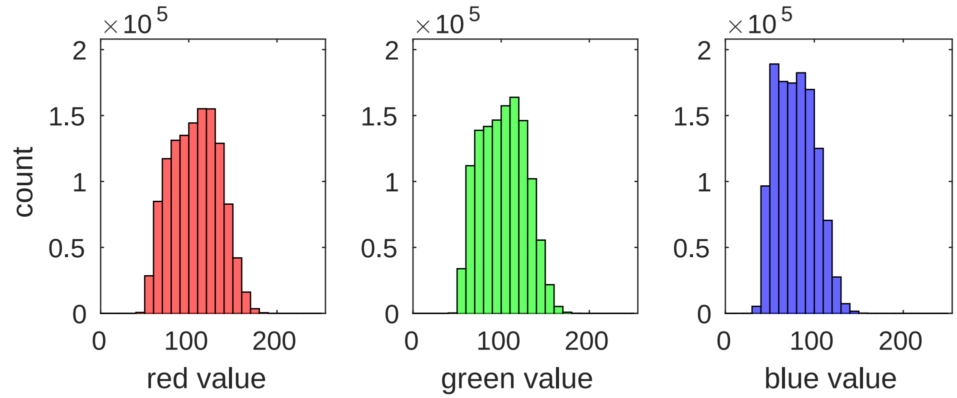
→




```
% Specify the path - directory string to where the 'fname' is
diro='/u/fjsimons/CLASSES/FRS-Spain/MatlabDemos/Matlab-Lec01/';
fname='H1W-18_35-test2-small.jpg';

% Read in the image using a canned Matlab function
rgb=imread(fullfile(diro,fname));

% Convert to  grey scale
red=rgb(:,:,1); green=rgb(:,:,2); blue =rgb(:,:,3);
% These values are unsigned 8-bit integer (from 0-255) so
% they require special attention to convert to grey scales
grae=uint8(round([double(red)+double(green)+double(blue)]/3));

% Make a picture
plot(grae(randi(size(grae,1),1),:)); axis tight
```

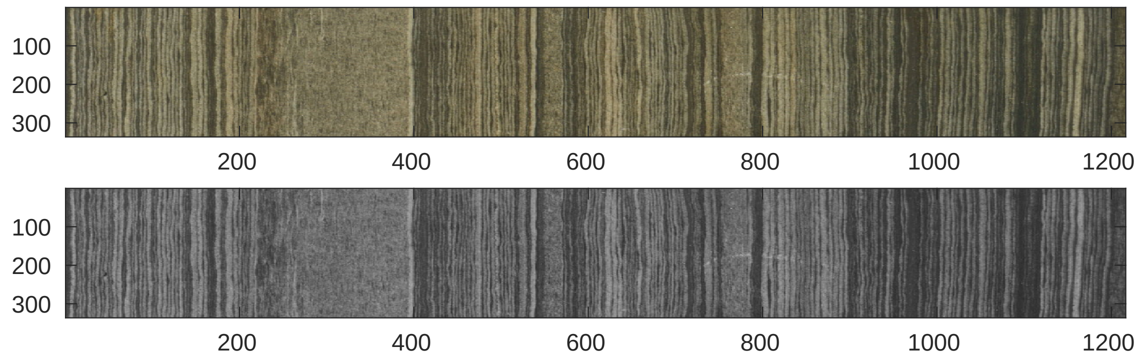
Write the first *script*

→




```
diro='/u/fjsimons/CLASSES/FRS-Spain/MatlabDemos/Matlab-Lec01/';
fname='H1W-18_35-test2-small.jpg';
rgb=imread(fullfile(diro,fname));
red=rgb(:,:,1); green=rgb(:,:,2); blue =rgb(:,:,3);

% Make histograms of the colors and annotate
ah(1)=subplot(231); h(1)=histogram(red,0:10:intmax('uint8')); xlabel('red value')
ah(2)=subplot(232); h(2)=histogram(green,0:10:intmax('uint8')); xlabel('green value')
ah(3)=subplot(233); h(3)=histogram(blue,0:10:intmax('uint8')); xlabel('blue value')

% Annotate; set the x-axes to reason and y-axes to the same limits
ah(1).YLabel.String='count'; h(1).FaceColor='red'; h(2).FaceColor='green'; h(3).FaceColor='blue';
set(ah(:),'XLim',[0 intmax('uint8')],'YLim',[0 1.1*max([h(:).Values])])

% Print the picture for inclusion in a report
print -dpdf madison02
```

Write the first *function*




```
function madison03(diro,fname,sizo)
% MADISON03(diro,fname,sizo)
% diro       directory string
% fname      image filename string
% sizo       size divisor

rgb=imread(fullfile(diro,fname));
red=rgb(:,:,1); green=rgb(:,:,2); blue =rgb(:,:,3);
grae=uint8(round([double(red)+double(green)+double(blue)]/3));

% Plot the sizoth part of the image in color
subplot(411); image(rgb(1:round(size(rgb,1)/sizo),:,:))
% And plot the same sizoth fraction in gray scale
subplot(412); image(repmat(grae(1:round(size(grae,1)/sizo),:),[1 1 3]))

% Print to a file with a filename that you learn from the function name
print('-dpdf',mfilename)
```

Code hygiene

1. "If you type it *twice*, you need to use a *variable*"

2. "If you say it in the *absolute*, you need to reformulate to the *relative*"

3. "Annotate all *graphs* completely, and give them *meaningful* names "

4. "Annotate all *code* completely, to a *ridiculous* degree"

Documentation/Help/Date

Input/Output

Computation/Algorithm

Figures/Embellishment

Variable Output

Five ways to do spectral analysis on time series

Inspection

Play with the plot, the axes, the grid lines, the annotations, etc.

Correlation

Guess a period, generate a synthetic, calculate how well they agree

Inversion

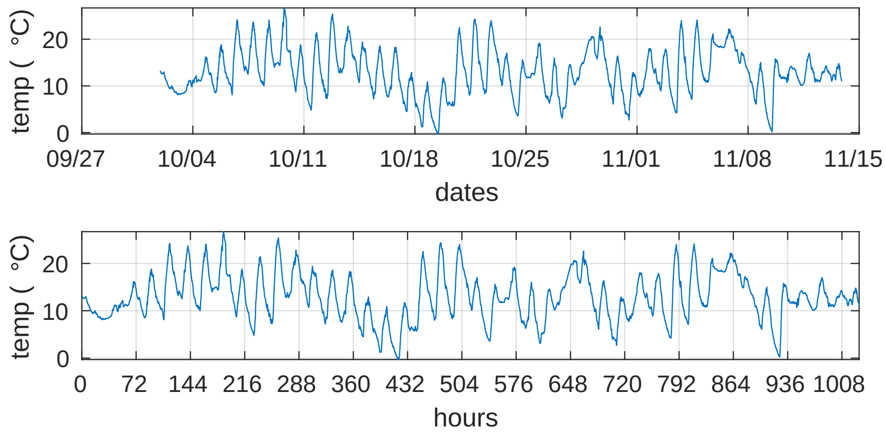
Guess a period and conduct a formal regression, calculate how well it fits

Stacking

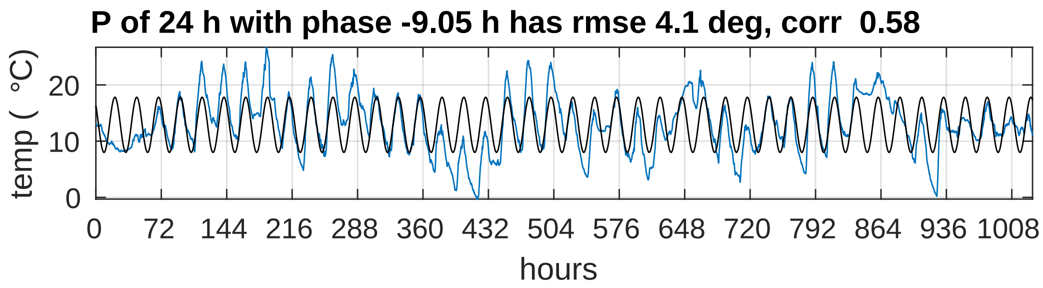
Guess a stacking length and see how well the result fits a meaningful oscillation

Fourier analysis

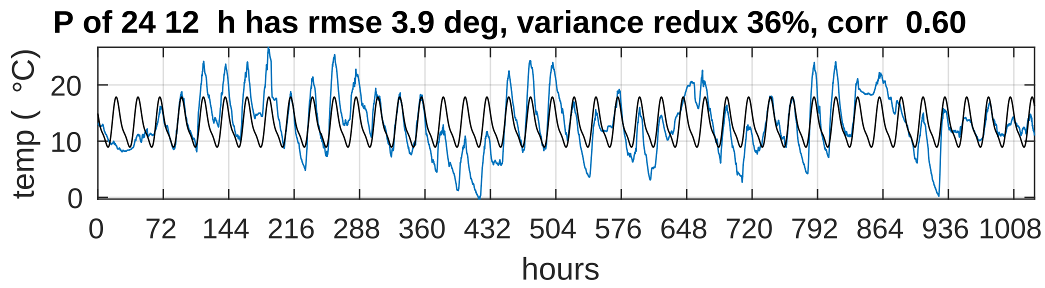
Do as the adults do! Inspect, interpret, and understand the results



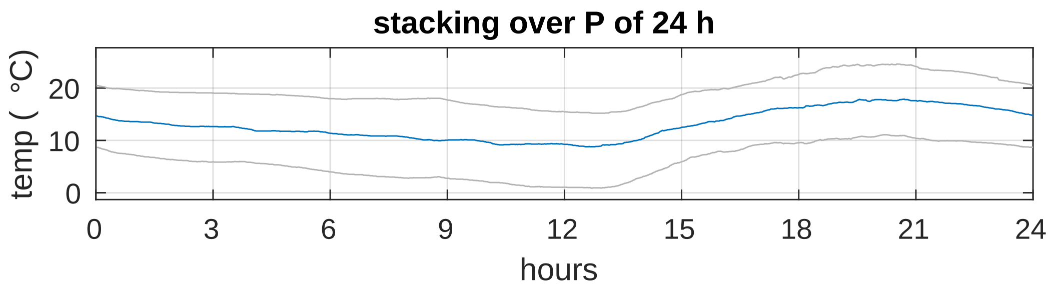

```
function madison04
% Read the data and identify the variables
w=webread('http://geoweb.princeton.edu/people/simons/weather_data.csv');
time=w.Var1; temp=w.Var5;

% Convert and plot in a human-intelligible format
dates=time/24/60/60+datenum(1970,1,1,0,0,0);
subplot(311)
plot(dates,temp); axis tight; grid on
datetick('x',6); xlabel('dates'); ylabel(sprintf('temp (%sC)','\circ'))

% Convert and plot in hours since the first sample which is last
hours=[dates-datenum(dates(end))]*24;
subplot(312); plot(hours,temp)
set(gca,'xtick',0:3*24:hours(1)); axis tight; grid on
xlabel('hours'); ylabel(sprintf('temp (%sC)','\circ'))
```



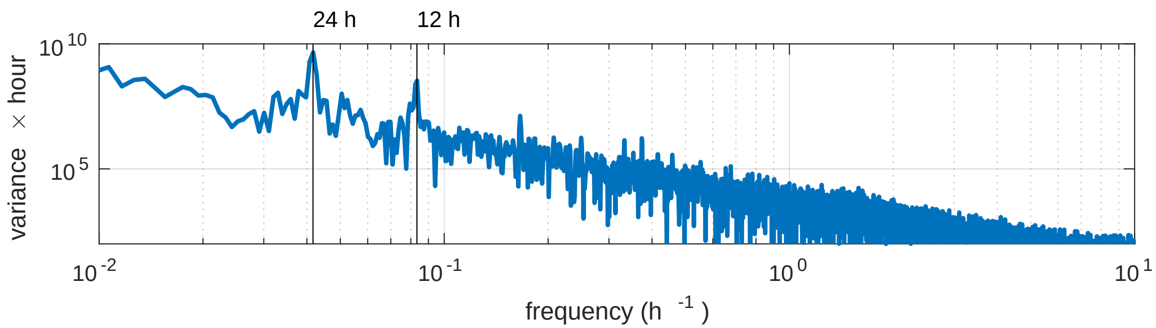

```
function madison05(P,ph)
% MADISON05(P,ph)
% P     Trial period (h)
% ph    Trial phase (h)

% Read the data, identify the variables and plot exactly as it was done by MADISON04
w=webread('http://geoweb.princeton.edu/people/simons/weather_data.csv'); time=w.Var1; temp=w.Var5;
dates=time/24/60/60+datenum(1970,1,1,0,0,0); hours=[dates-datenum(dates(end))]*24;
subplot(311); plot(hours,temp); set(gca,'xtick',0:3*24:hours(1)); axis tight; grid on
xlabel('hours'); ylabel(sprintf('temp (%sC)','\circ'))

% Make a 24-hour sinusoid to overlay on the data, calculate RESIDUAL and ROOT-MEAN-SQUARED ERROR
A=std(temp); m=mean(temp); tpred=m+A*sin(2*pi*[hours-ph]/P);
resd=temp-tpred; rmser=sqrt(sum(resd.^2)/length(resd));
hold on; p=plot(hours,tpred,'k'); hold off

% Compute the CORRELATION COEFFICIENT between the synthetic and the data and finish the plot
[r,pval]=corrcoef(temp,tpred);
title(sprintf('P of %g h with phase %g h has rmse %3.1f deg, corr %5.2f',P,ph,rmser,r(2)))
```




```
function madison06(P)
% MADISON06(P)
% P    A vector of trial periods (h)

% Read the data, identify the variables as in MADISON04
w=webread('http://geoweb.princeton.edu/people/simons/weather_data.csv'); time=w.Var1; temp=w.Var5;
dates=time/24/60/60+datenum(1970,1,1,0,0,0); hours=[dates-datenum(dates(end))]*24;
subplot(311); plot(hours,temp); set(gca,'xtick',0:3*24:hours(1)); axis tight; grid on;
xlabel('hours'); ylabel(sprintf('temp (%sC)','\circ'))

% Do the INVERSION, PREDICTION, RESIDUAL, ROOT-MEAN-SQUARED ERROR, VARIANCE REDUCTION, CORRELATION
argm=[1./P(:)*hours(:)']'; F=[ones(size(temp)) cos(2*pi*argm) sin(2*pi*argm)];
A=pinv(F)*temp; tpred=F*A; resd=temp-tpred; [r,pval]=corrcoef(temp,tpred)
rmser=sqrt(sum(resd.^2)/length(resd)); vard=100*[1-var(resd)/var(temp)];

% Now finish the plot by plotting the prediction right on top of the data
hold on; p=plot(hours,tpred,'k'); hold off; title(sprintf(sprintf(...
    'P of %s h has rmse %s, variance redux %s, corr %s',repmat('%g ',size(P)),...
    '%3.1f deg','%g%%','%3.1f'),P,rmser,round(vard),r(2)))
```




```
function madison07(P)
%%%%%%%%%%%%%%%%%%%  Data loading etc %%%%%%%%%%%%%%%%%%
% Remove duplicates since we will be interpolating later
[hours,isort]=unique(hours); temp=temp(isort);

% About how often did we sample and how many blocks of length P can we identify? Interpolate
sinth=1/round(1/abs(median(diff(hours)))); mblock=floor(max(hours)/P)*P;
hoursi=[sinth:sinth:mblock]'; tempi=interp1(hours,temp,hoursi,'linear');

% Now we are ready to 'stack', all of these are segments of exactly P in length
hoursd=reshape(hoursi,P/sinth,[]); tempd=reshape(tempi,P/sinth,[]);

% With a little luck these are ALL the same modulo P!
if all(all(diff(mod(hoursd,P),[],2)<1000*eps)); hoursu=hoursd(:,1); end

% Now comes the 'median stack' and some idea of the variability
tempu=nanmedian(tempd,2); tempup=prctile(tempd,95,2); tempdn=prctile(tempd,05,2);

%%%%%%%%%%%%%%%%%%% Plotting etc %%%%%%%%%%%%%%%%%%%%%%%%
```




```
function madison08
% Read the data, identify the variables as in MADISON04
w=webread('http://geoweb.princeton.edu/people/simons/weather_data.csv');
time=w.Var1; temp=w.Var5; dates=time/24/60/60+datenum(1970,1,1,0,0,0);
hours=[dates-datenum(dates(end))]*24; subplot(311);

% Make a frequency axis ahead of time
nfft=length(hours); physl=hours(1);
selekt=1:floor(nfft/2)+1; fax=(selekt-1)'/physl*(length(temp)-1)/nfft;

% Compute a naive spectral estimate, the periodogram, and plot it
S=abs(fft(hanning(length(temp)).*[temp-mean(temp)],nfft)).^2;
loglog(fax,S(selekt),'linew',2); grid on
xlabel('frequency (h^{-1})'); ylabel(sprintf('variance %s hour','\times'))

% Annotate
P=[24 12]; hold on ; plot(1./[P(:) P(:)]',repmat(ylim,length(P),1)','k'); hold off
text(1/P(1),1e11,sprintf('%i h',P(1))); text(1/P(2),1e11,sprintf('%i h',P(2)))
xlim([1e-2 10]); ylim([1e2 1e10])
```

Conclusions

- All Earth scientists coming of age in this century will write computer code
- Computational analysis need not be scary if you take it slow!
- Learning to mix paint won't turn us into Rembrandts, but *painting* might!
- Mastering the Java 'while-loop' hasn't taught anyone the scientific method
- Students will *want* to program the computer when they have data of their own
- Home-grown data can come from smartphones, local weather stations, etc
- Matlab is a sophisticated, yet low-threshold language with staying power
- Geosciences are a *gentle first step* into a life of the programming scientist
- Coding enables students to *take control* and is a tool for *diversity*
- Home and student licenses need not be costly, open-source clones exist