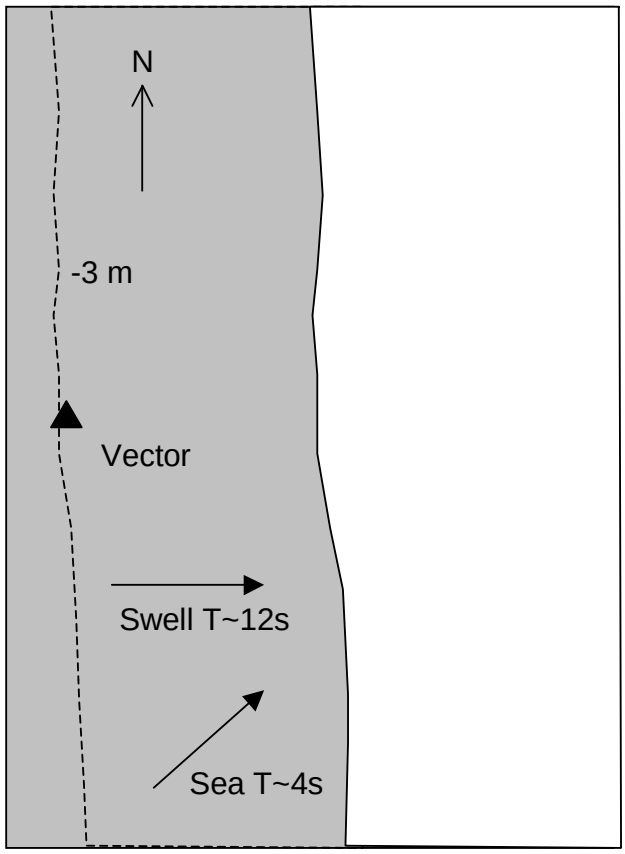
DIWASP sample analysis (Version 1.1)

This tutorial describes a sample analysis of a three component measurement. This should be read in conjunction with the DIWASP user manual, in particular the section describing the data structures used in DIWASP.

The data file

File '**sample.dat**' is data from a Nortek Vector velocimeter. The three columns are of pressure (meters of water), x-component velocity (m/s) and y-component velocity (m/s).

The sensor deployment is 20cm above the bed in approximately 3 meters of water with instrument axes oriented with +ve x towards the east and +ve y towards the north. The data is a single 10 minute burst at 2 Hz sampling rate.



Hydrodynamic conditions

The data was collected during a field experiment to investigate seabreeze activity in Perth, Western Australia. The shore runs north-south with the ocean to the west. The seabreeze blows from the SW quadrant during the afternoons at around 10-15m/s during the summer months. The sample data is from a mid-afternoon burst with the seabreeze near to its full strength and the associated sea well developed. In addition there is a longer period swell that is refracted to be nearly perpendicular to the shore at the depth of the instrument.

The analysis

Load the instrument data into the data field of the instrument data structure:

```
data = load ('sample.dat')
```

The other inputs for the ***dirspec*** function are contained in '**inputs.mat'**. This can be loaded directly into the workspace. It contains the Instrument data with data in the fields as follows :

```
ID.datatypes = {'pres' 'velx' 'vely'}
```
(these are the datatype codes)

```
ID.layout = [ 0     0     0      (these are the x positions)
              0     0     0      (these are the y positions)
            0.2   0.2   0.2]    (these are the z positions)
```

Note that the instrument positions are relative to an arbitrary horizontal origin so are all at the same (0,0) but must be fixed absolutely vertically from the bed.

```
ID.fs = 2
```
(this is the sampling frequency)

An additional required field is the depth but this can be input by using the existing pressure data to approximate the mean sensor depth:

```
ID.depth = mean(data(:,1))+ 0.2
```

Finally, put the instrument data in the data field:

```
ID.data = data
```

The instrument data structure which describes everything about the instrument data is now complete.

The second structure is the spectral matrix structure which contains fields describing the bin layout of the spectral matrix:

```
SM.freqs = [0.01:0.01:0.5]
```
(these are the frequency bins for the output)

```
SM.dirs = [-180:1:180]
```
(these are the directional bins for the output)

The other field in the spectral matrix structure is the direction of the x-axis. This can be explicitly

```
SM.xaxisdir = 90
```

This will default to 90 if not defined.

The third structure defines the parameters used in the estimation process. Here all the default values are used:

`EP.dres = 180` (this is the number of directions in the calculation)

`EP.nfft = 256` (this is the number of DFTs in the spectral estimation)

`EP.iter = 100` (this is the number of algorithm iterations)

`EP.smooth = 'ON'` (this enables smoothing of the final output)

`EP.method = 'EMEP'` (This is the best choice for high quality estimation with a 3 quantity point measurement)

Note that if the estimation parameters structure can be defined as empty:

```
EP = []
```

which also means default values are used.

Run the analysis with options to produce a polar plot and write the file to 'sampleout.spec':

```
[SMout,EPout]=dirspec(ID,SM,EP,{'PLOTTYPE',2,'FILEOUT','sampleout.spec'});
```

The returned structures are the spectral matrix with the estimated spectral density and the estimation parameters structure with values used in the estimation – any fields explicitly specified will be unchanged, otherwise the default values are present.

Output

You should get an output plot that looks the same as the plot on the front of the user manual. The spectrum shows a narrow banded, directionally tight peak centred around 0 degrees and 0.08Hz. This is the swell component of the sea. There is also a higher frequency tail associated with this down to around 0.16Hz. The second much broader peak is centred around 20-30 degrees and 0.25Hz. This is the seabreeze-generated sea that is approaching from the SW quadrant and propagating to the NE. It is much more spread in direction that is expected for a "young" sea and is supported by visual observations of short crest lengths.

Changing the parameters and the estimation parameters structure results in significant differences in the output. The parameters `EP.nfft, EP.dres` and `EP.iter` mainly govern the resolution while the estimation methods can produce significantly different results. It is worth looking at the effects of changing these

parameters on the same data to understand how each works. In particular this will demonstrate the differences between the estimation methods.

Manipulating the output file

The dirspec function will have produced a file called ‘sampleout.spec’ and returned a spectral matrix structure. The file contains the spectral matrix of the results in DIWASP format. To read a file back into a MATLAB spectral matrix structure just use:

```
[SM]=readspec('sampleout.spec')
```

which will read the spectral matrix into `SM.S`, and the frequency and directional bins into `SM.freqs` and `SM.dirs` respectively. `SM.xaxisdir` will be 90 as the axes are oriented with the x-axis pointing east.

The spectral matrix can be passed to the other functions which plot, write, and return information about the spectrum – these are all called by *dirspec.*

```
plotspec(SM,2)
```

should make the same plot as the initial dirspec function while

```
plotspec(SM,1)
```

should make a 3D plot of the same data. If you want to make a 3D plot with compass directions instead of polar coordinate angles then use:

```
plotspec(SM,3)
```

If you didn’t specify an output file in the ***dirspec*** function but wanted to write the data in the spectral matrix, `S`, to a new file use:

```
writespec(SM,'newfile.spec')
```

To get information like the significant wave height use:

```
infospec(SM,3)
```

The other functions, *makespec* and *testspec* are not used in field data analysis but are useful for testing the estimation routines.