

National Oceanography Centre, Southampton

**Addendum to
Cruise Report No. 23**

RRS Discovery Cruise 321
24 JUL – 23 AUG 2007

The SUV-6 Dissolved Nitrate Sensor

Authors

R E M Pidcock
M Srokosz

Principal Scientist

J T Allen

Editor

S C Painter

2008

National Oceanography Centre, Southampton
University of Southampton
Waterfront Campus
European Way
Southampton
Hants SO14 3ZH UK
Tel: +44 (0)023 8059 6429
Fax: +44 (0)023 8059 6472
Email: jta@noc.soton.ac

Contents

	Page
FOREWORD	5
1. INITIAL PROCESSING	5
2. POST-CRUISE PROCESSING AND CALIBRATION	10
Processing Failure and Data Recovery	11
Existence of ‘Ramp Up’ Periods	12
Shifting Calibrations	14
Full-Depth Calibration	16
References	20
Appendix	20

The SUV-6 Dissolved Nitrate Sensor

Foreword

The first part of this addendum is the original section of cruise report written by Meric Srokosz, detailing the processing of the raw SUV-6 data files on D321. No amendments have been made to this part of the report and it stands in its original form. The second part of the report details the post-cruise processing and calibration procedure. Footnotes appear in the first part of the report alongside statements or speculations that, following the post-cruise processing, require additional comment. In these cases, the footnotes refer to specific sections in the second part of the report. Where footnotes existed in the original text of the first part of the report, they remain unchanged.

1. Initial Processing

Meric Srokosz

The SUV6 was flown on SeaSoar and the data acquired and logged through Penguin. The data files were numbered randomly `suv6nnn.000` or `suv6nnnn.000` (n.b. the corresponding minipack and OPC files produced by Penguin are also randomly numbered, so matching files to merge data is problematic). These were transferred to the directory `/data32/d321/suv6` for further processing. The data files are structured:

JDAY CH1 CH2 CH3 CH4 CH5 CH6 CH7

Where JDAY is the decimal Julian day, CH1 to CH6 are the measurements from the 6 channels (see below), and CH7 is a duplicate of CH6. The data are logged once per second and every 60 seconds of data contains 45 measurements and 15 internal calibration loop measurements. Note that the files can start and end anywhere in this sequence of 60 measurements.

The SUV6 produces measurements in 6 UV channels (205, 220, 235, 250, 265, 280nm) and the nitrate measurement is made from the relative absorbance in the 220nm and 280nm channels. The 205nm channel is sensitive to salinity. The data were analysed using Matlab code provided by Ralf Prien (ex-NOC) for use during a previous cruise (MadEx, D288; see Quartly, 2006). This code had been amended to fix a problem with array indexing. It was further amended here to change the calibration threshold from 2000 to 1400, as this seemed to be what was output by the instrument. The code is provided in the appendix (below).

When running the code it would fail if the data file did not end with a series of 3 calibration measurements. This was cured easily by editing out a few lines at the end of the data files to ensure that the last 3 lines consisted of calibration data (identified by CH1 value being greater than 1400). Despite these changes to the code, some files caused the Matlab code to crash due to an array-indexing problem. It was not possible to fix this during the cruise as it was unclear why this was happening (whether it was a problem with the data file or with the code).¹

A table of the files processed during the cruise is given in Table 1². Note that the files cannot be concatenated as programme assumes that you get measurements in batches of 60 lines (45 data, 15 calibration) and individual files start and end at random points in the sequence (as noted above).

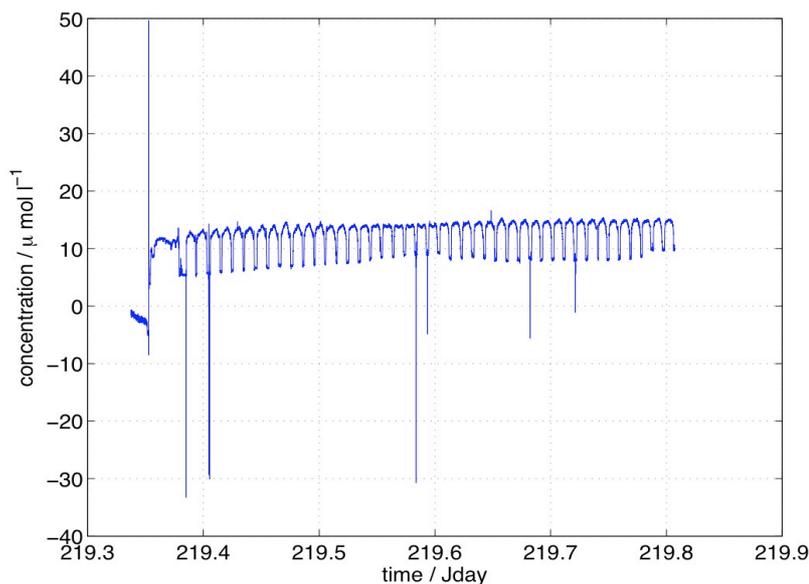


Figure 1 Nitrate concentration ($\mu\text{mol/litre}$) against time, for data processed from file suv6376.000, before de-spiking.

The Matlab code produced nitrate concentration as a function of Jday for each file. It also plotted the data (see example in Figure 1). As a first step the data that the code saved in Matlab format were merged with raw SeaSoar minipack data for pressure, temperature,

¹ This was later identified as a problem due to poor coding. It was fully rectified, enabling all original data files to be processed. Refer to section 2.1 for details.

² To avoid duplication, Table 1 has been moved to the second section of the report where it has been updated to include the files processed on our return to NOCS.

fluorescence and PAR. This allowed simple plots to be produced to see whether the SUV6 was giving sensible nitrate data.³ Visual inspection of the nitrate results from the first SeaSoar survey suggested that it lay in the range $\sim 4\text{-}18 \mu\text{mol/litre}$, so spikes outside this range were edited out. An example of the processed data from file `suv6376.000` is shown in Figure 2 (below).

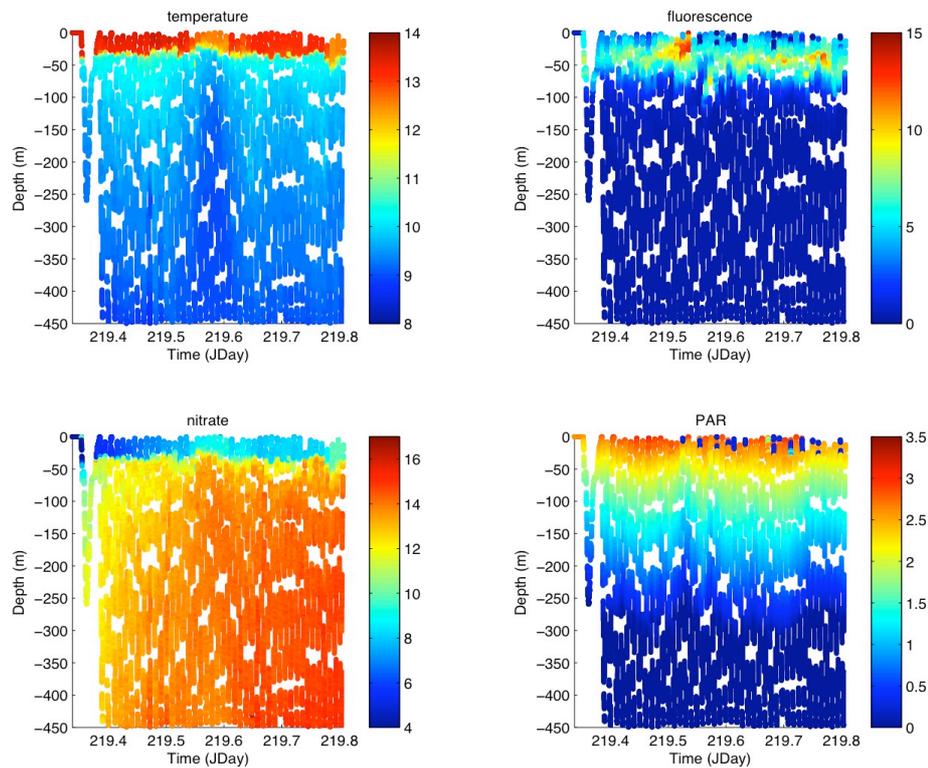


Figure 2 Uncalibrated temperature, fluorescence, and PAR with SUV6 nitrate ($\mu\text{mol/l}$) from the SeaSoar tow on Jday 219 (data from `suv6376.000` and `minipack353.000`). Note that these plots are simply the ups and downs of SeaSoar plotted against time, hence their “gappy” nature. They also contain various “glitches” as they have not been cleaned up.

It can be seen that the SUV6 gives nitrate values that lie in a reasonable range, but the calibration (which was based on laboratory measurements prior to the MadEx cruise in early 2005) needs adjusting. Surface nitrate values from water samples during the cruise lay in the range $\sim 1\text{-}6 \mu\text{mol/litre}$, while those at 400m depth lay in the range $\sim 15\text{-}20 \mu\text{mol/litre}$. From

³ Used the Matlab routine `nogrid_plot2d.m` by Paolo Cipollini, which allows data to be plotted without first being gridded.

Figure 2 this suggested that the SUV6 could be overestimating surface values, though the values at depth might be o.k.⁴

Figure 2 clearly shows that the nitrate is low in the mixed layer and that the nutricline is at the base of the mixed layer (as might be expected – seen by comparing nitrate and temperature). There is evidence that the cold core eddy traversed around Jday 219.6 is reflected in the nitrate distribution, and that nitrate is higher in the cold water at the surface above the eddy. In addition, the high fluorescence patch is associated with locally lower nitrate values, as might be expected if the phytoplankton were taking up nitrate.

The question that the figure raises is whether there is a period at the beginning of the deployment when the SUV6 is adjusting to it's surrounding and the sensor drifted (period affected up to around Jday 219.55). The first dip has higher nitrate values, but this could be due to horizontal gradients in nitrate in the water as well. It is not possible on the basis of the SUV6 data alone to differentiate between these causes. As a comparison, Figure 3 shows the results from later in the same SeaSoar tow and there is no evidence of drift in the nitrate data. Therefore, if the sensor was drifting initially on deployment it seems to have stopped doing so, presumably as it acclimatised to its surroundings. On a shorter deployment (suv6478.000) the SUV6 showed similar behaviour (results not shown).⁵

Further deployments (on the second SeaSoar survey) revealed another problem; as illustrated in Figure 4. As well as the drift seen at the start of each of the earlier deployments, and still apparent here, there also appears to be a shift in the actual values of nitrate being estimated. On the earlier deployments the range was ~4-18 $\mu\text{mol} / \text{litre}$ (see Figures 2 and 3), but on the later deployments the range was ~8-23 $\mu\text{mol} / \text{litre}$ (see Figure 4).⁶

⁴ This was the result of a cursory examination during the cruise. It was later shown through comparison with both the ships non-toxic underway supply and CTD measurements that the data would need calibrating to both surface and at-depth values. Refer to sections 2.3 and 2.4 for full details.

⁵ Further examination of the data revealed that the instrument experienced a period of acclimatisation at the beginning of each tow, where the lamp appeared to 'warm up'. An internal reference path in the instrument monitors these changes, but deficiencies in the processing code mean that it is not fully corrected for. See section 2.2 for details.

⁶ On detailed analysis of the surface (4-5m) data, a drift was uncovered towards increasingly higher nitrate values with each deployment of the instrument. This required each tow to be calibrated individually to underway values. See section 2.3 for details.

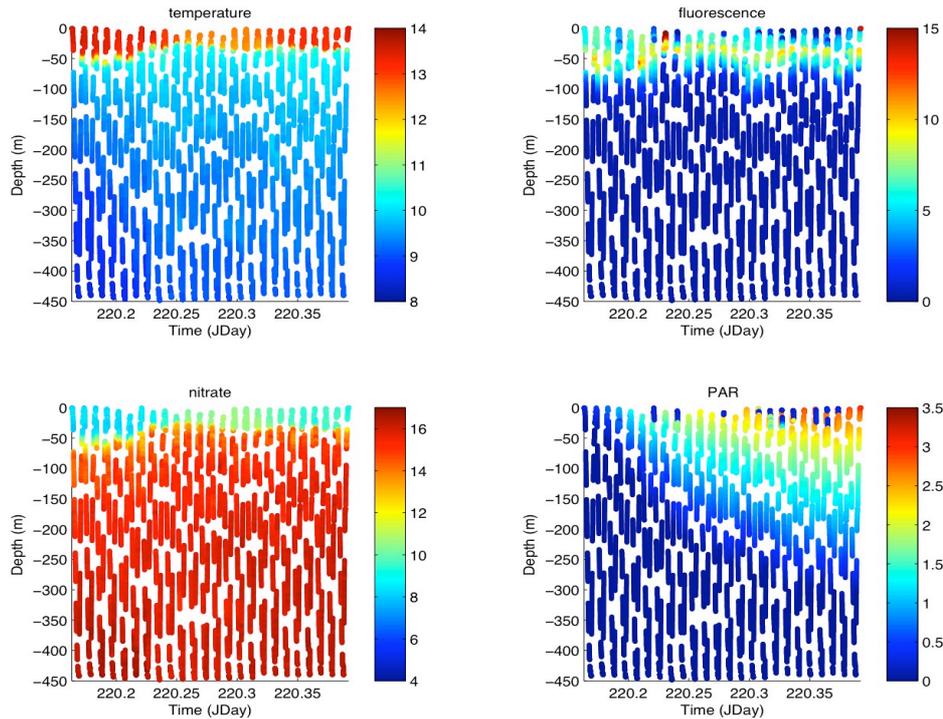


Figure 3 Uncalibrated temperature, fluorescence and PAR with SUV6 nitrate ($\mu\text{mol/l}$) from the SeaSoar tow on Jday 220 (data from suv6347.000 and minipack324.000).

The initial intention had been to calibrate the SUV6 nitrate values against the Auto analysed one-hourly underway nitrate samples. Unfortunately, due to the drift of the sensor during a deployment and the apparent shift in calibration between deployments, there were too few underway nitrate values to characterise this behaviour. A further option considered was to characterise the sensor on deck by placing it in buckets of different strength nitrate solution and recording data via Penguin (as is done on the SeaSoar tows). Since this would need to have been done at the end of each tow, due to the sensor drifts described above, it was impractical to carry out. Removing the SUV6 and Penguin from SeaSoar is complex. As the final SeaSoar tow was not completed until late in the cruise, there wasn't sufficient time to do a final on deck calibration. Clearly the drift during deployments and shift in nitrate values between deployments identified here need to be explained and corrected for if the SUV6 is to be used routinely on SeaSoar. Whether these sensor characteristics can be reproduced and characterised in the laboratory after the cruise remains to be seen. A final point to note is that the nitrate values obtained here are based on only two of the six SUV6 channels. Therefore it may be possible to use the information in the other channels to make the necessary corrections.

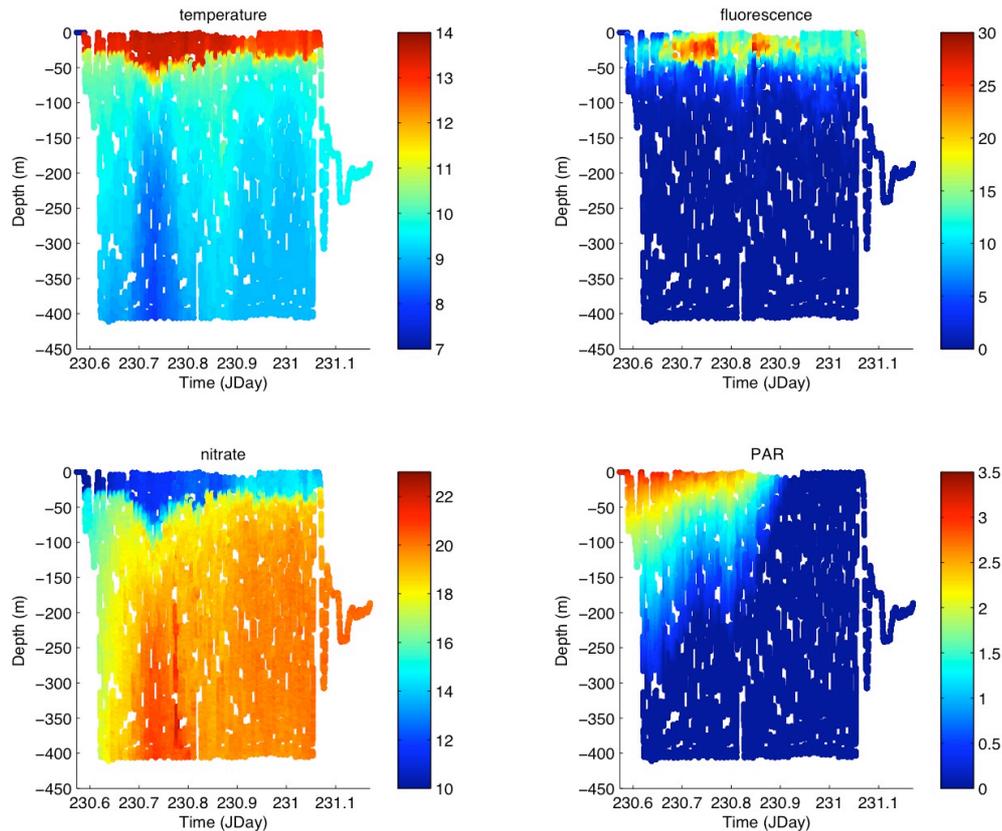


Figure 4 Uncalibrated temperature, fluorescence and PAR with SUV6 nitrate ($\mu\text{mol/l}$) from the SeaSoar tow on Jday 230-231 (data from `suv6525.000` and `minipack497.000` plus `minipack25724.000`). Note change in scale for nitrate, as compare to that in Figures 2 and 3.

2. Post-Cruise Processing and Calibration (November 2008)

Roz Pidcock

Several issues were raised during the on-board processing requiring investigation on our return to NOCS.

- (i) The reasons for the indexing array problems that caused the on-board processing of two SUV6 data files to fail.
- (ii) The existence of apparent ‘ramp up’ periods upon deployment of the instrument where the data experienced an initial drift before stabilising.
- (iii) The apparent shift in calibration values between deployments, towards increasingly higher nitrate concentrations.

Processing Failure and Data Recovery

It was found that the matlab code failed to process two files on-board because of poor coding. The programme as provided was unable to distinguish between genuine data gaps and those created by stripping out the internal calibration periods, leading to a mismatch between the number of data periods and calibration periods. This was rectified in the code and the remaining two files were successfully processed.

Table 1 below details the original files processed on board D321. It has been amended from the version described in the first part of this report to include the files processed on our return to NOCS. The total SeaSoar survey period consisted of 5 separate tows, the first two as part of the first grid-box survey (hereafter referred to as 1a and 1b) and the third, fourth and fifth tows as part of the targeted second survey, (hereafter referred to as 2a, 2b and 2c).

Table of raw Penguin SUV6 files and corresponding processed file numbers

	SeaSoar survey no.	Raw SUV6 file no. (chronologically)	Minipack file no. (used to merge)	Processed files
1a	1	suv6478.000	mini pack464.000	Nitr_suv6478.mat channels_suv6478.mat Nitr478. mat
	1	suv6387.000	too short to merge with minipack data	Nitr_suv6387.mat channels_suv6387.mat
1b	1	suv6376.000	mini pack353.00	Nitr_suv6376.mat channels_suv6376.mat Nitr376. mat
	1	suv62280.000	mini pack2248.00	<i>Nitr_suv62280.mat</i> <i>channels_suv62280.mat*</i>
	1	suv6347.000	mini pack324.000	Nitr_suv6347.mat channels_suv6347.mat Nitr347. mat
2a	2	suv6344.000	mini pack333.00 mini pack24678.000	Nitr_suv6344.mat channels_suv6344.mat Nitr344. mat
2b	2	suv6525.000	mini pack497.000 mini pack25724.000	Nitr_suv6525.mat channels_suv6525.mat Nitr525. mat
2c	2	suv6516.000	mini pack497.001	<i>Nitr_suv6516.mat</i> <i>channels_suv6516.mat*</i>

* These are the files that the Matlab code failed to process whilst on board. On our return to NOCS the coding glitch was fixed and the files could be processed. The Nitr_suv6nnn.mat and the channels_suv6nnn.mat files were produced by the code as in the other files. The Nitrnnn.mat files were not created as the merging with minipack data was done later in Pstar.

n.b. see code in appendix for information on what is in files Nitr_suv6nnn.mat and channels_suv6nnn.mat. Files Nitrnnn.mat contain the variables:

time press nitr fluor Tfluor PAR temp

i.e. *time* and *nitr* from the SUV6 processing merged with the uncalibrated SeaSoar Minipack data for *press fluor Tfluor PAR temp*. Minipack data interpolated on *time* to SUV6 *time*

Existence of 'Ramp up' Periods

Having run the initial processing code on all the available SUV6 data files in Matlab, each one was output as an ASCII data file and read into Pstar, using a script called *matlabconvert.m* to replace the Matlab absent data value (NaN) with the Pstar absent data value (-999). An extra variable was also created to record the time in seconds from the beginning of the year. The individual files were then appended in time order.

To begin calibration, SUV6 data from between 4-5m were extracted using the Pstar *datpik* routine and merged with data from auto-analysed bottle samples taken from the ship's non-toxic supply at hourly intervals throughout the cruise. A scatter plot of surface (4-5m) SUV6 data for the separate tows against the underway data showed a set of reasonably straight lines staggered in space. This showed some potential for applying linear calibrations to the data, but confirmed that the calibration changed each time the instrument was deployed, recording increasingly higher surface concentrations with each deployment.

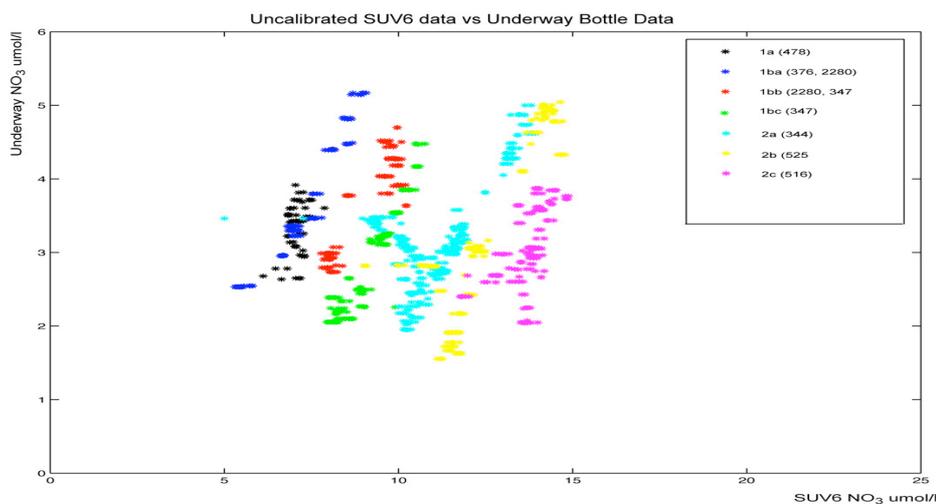


Figure 5. SUV6 data plotted according to where different calibrations exist. These correspond to the distinct files except in tow 1b where 3 separate calibrations exist, overlapping the file breaks (1ba, 1bb and 1bc).

A further complication revealed by the scatter plot was that even within one of the tows, (1b), three separate calibrations existed, which furthermore, did not correspond to the file breaks. These will be referred to as 1ba, 1bb and 1bc. The different offsets in each tow are dealt with later in this report.

A similar scatter plot to that described above but with the data divided up according to where the calibration changed, rather than simply by tow revealed a similar but cleaner succession of straight lines (Fig. 5). A noticeable deviation from this can be seen at the very beginning of each deployment during which time nitrate values started considerably lower than expected but then gradually recovered. This ‘ramp up’ period ranged from 3.5 hours to 7.5 hours between the individual files. The ramps are most obvious as an arc shape in the scatter at the beginning of files 2a and 2b (see Fig. 5). They do not appear in files 1bb and 1bc as these are within a tow, by which time the instrument had stabilised. They are masked in 1a and 2c by an overall drift in the data with time; this will be dealt with later in this report.

Table 2. Details of how each section was treated to calibrate the data to surface underway data.

Section	Calibration Details
1a	3.5 hrs removed from start Linear time drift removed
1ba and 1bb together	3.5 hrs removed from start Linear time drift through both sections removed
1bc	No ramp up period Offset of -6.27
2a	7 hours removed from start Offset of -8.33
2b	4.5 hrs removed from start Offset of -9.32
2c	Linear time drift removed

The SUV6 employs an internal reference path to monitor changes in the lamp spectrum over time. Periods of ‘ramp up’ in the NO₃ data coincided with a slow curve towards a stable constant when the internal reference data was plotted against time (not shown). Equally, the internal path showed a straight line when ramp up periods were not experienced in the data. On close inspection of the Matlab processing code, it was found that a step to normalise the measured nitrate absorbance at each data point to the internal reference value was reducing the effect of the ramp up period, but not correcting it fully. A series of tests showed that a

quadratic or logarithmic function did not correct the data better than the existing linear function. In any case, a linear function should be the most appropriate correction to apply given the linear relationship between UV absorbance and nitrate concentration. This work is on going and will require consultation with the engineering side of the NOCS SUV6 team. For the time being and for the purposes of moving forward, the ‘ramp up’ periods have been removed from the data altogether (detailed in Table 2). The lengths of the ramp up periods for each tow were consistently obtained from the internal reference data, the criteria being the time taken to achieve a stable constant value. The calibration of data as described in the remainder of this report concerns only the data within the ‘good’ periods after the lamp had stabilised.

Shifting Calibrations

After removing the ‘ramp up’ periods, plots of surface SUV6 nitrate, discrete underway (uw) nitrate samples and the offset between them (SUV-uw) were created for each tow. From these, an average SUV-uw offset was determined for each tow. For two of the five tows, a simple offset could be applied to the SUV6 data to calibrate it to surface underway values. The SUV6 nitrate data in another three of the tows appeared to show a linear drift with time, which, once identified, was removed using a series of standard Pstar routines. It was observed that within the longest of all the tows (1b), three separate calibrations existed; seemingly a constant offset at each end, (1ba and 1bc), and a linear time drift in the middle section (1bb). However, with very few surface data points, the transition between the offset in 1ba and the start of the linear drift in 1bb was fairly severe. This is likely to have caused problems when it came to contouring the data in later stages. With this in mind and together with the fact that a large proportion of 1ba (3.5 hrs) had to be removed as a ‘ramp up’ period leaving very little data behind, another look at the data determined that a single drift could cover the ‘useable’ data in both 1ba and 1bb.

Following their separate calibrations, SUV6 surface data from all five tows fitted a 1:1 relationship with the underway data, (Figure 6). The individual calibrations were then applied to the full depth data for each section. Calibration of the SUV-6 data to surface underway data by the method described only involved a very small range of nitrate values, (surface values ~2-6 $\mu\text{mol/l}$). It was therefore necessary to check that the calibration was appropriate for the full depth data before accepting it. Unfortunately, when all the SUV6 nitrate data with the surface calibrations applied were plotted against CTD bottle nitrate data down to 450m, it

was clear that this was not the case. The range of nitrate concentrations over which the SUV6 measured was significantly smaller than that for the CTD bottle samples, (see Figure 7). That this should be the case is perhaps not surprising given that the most recent laboratory calibration prior to D321 was carried out before the MadEx D288 cruise in 2006. Bearing in mind the apparent instability of the instrument, it is more than likely that the time lapse between the lab calibration and its use on our cruise render the existing calibration coefficients useless.

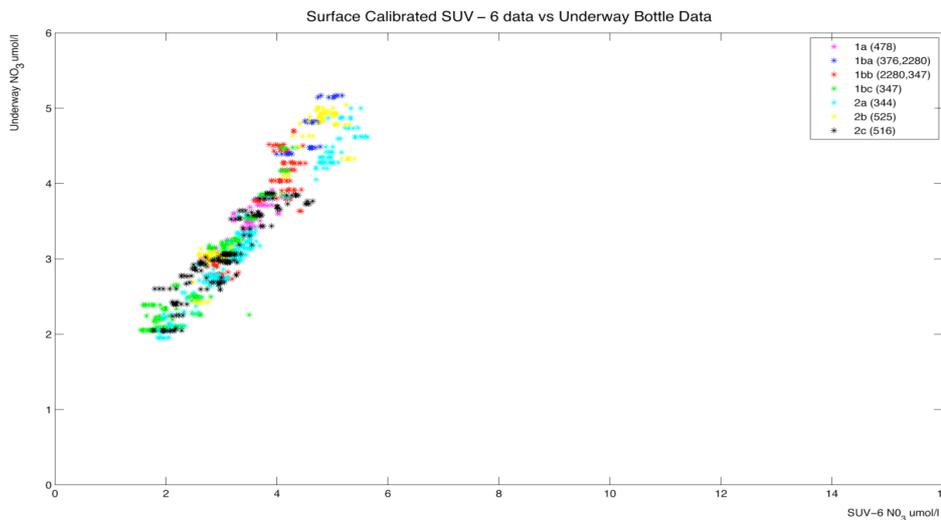


Figure 6. Following individual calibrations to surface data, SUV-6 data from all tows fit a 1:1 relationship with the underway data.

It has been confirmed that a dedicated effort towards further development and optimisation of the SUV6 instrument will take place in early 2009. Important starting points will be to find out more about how the original calibration coefficients were achieved and to carry out a lab calibration ourselves to test instrument's stability in different conditions and to attempt to replicate some of the characteristics experienced in the D321 dataset. A second important area will be to modify the processing in such a way as to make use of all six channels recorded by the instrument, as the current processing code does not currently maximise the instrument's capabilities. Whilst there is clearly much room for development and improvement of the instrument for future deployments, in order to make use of the current dataset a second calibration of the data against the CTD data to full depth was necessary.

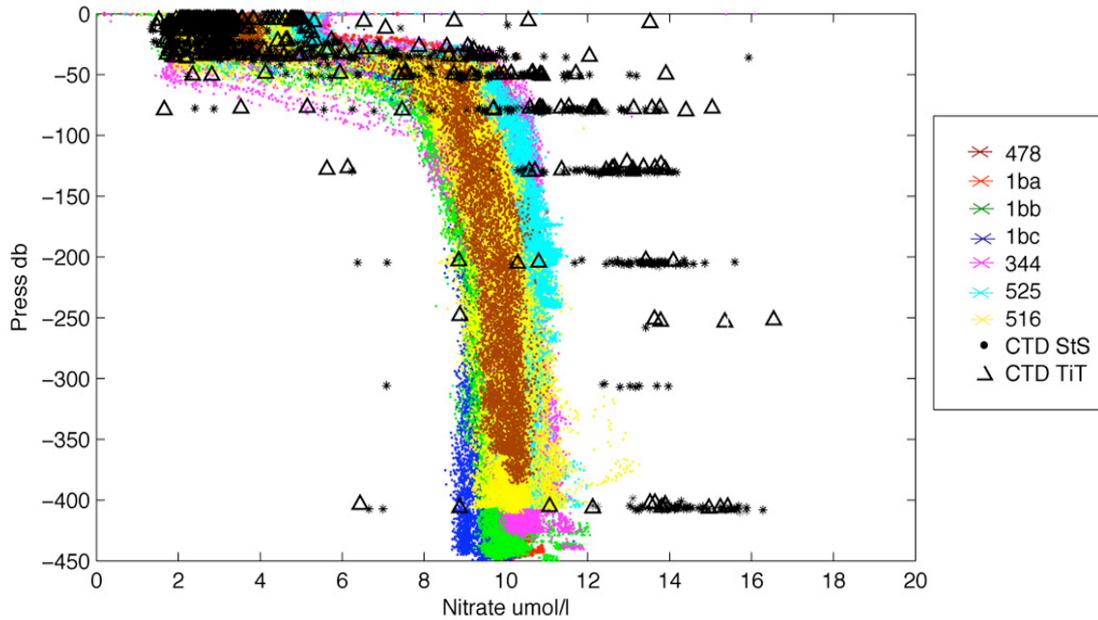


Figure 7. Surface calibrated SUV6 data plotted against CTD nitrate data down to 450m. SUV6 data is colour-coded by tow according to the key. Stainless steel CTDs are marked with filled circles, titanium casts are marked with triangles.

Full Depth Calibration of SUV-6 Data

The first step in the full-depth calibration was to ascertain the amount of scatter in the SUV6 data covered by all 5 sections compared to the scatter in the CTD data. Scatter plots of surface calibrated SUV6 nitrate concentration against depth for each section were first overlaid on to each other, and then the data from all the tows were overlaid onto the CTD bottle nitrate data. The datasets fitted well at the surface, not surprisingly, but below a depth of ~40m the SUV6 recorded significantly lower nitrate values than observed in the CTD bottle samples. However, the range of the scatter in the SUV6 data was less than that in the CTD bottle sample data. This demonstrated the potential and justification for applying a nitrate-based function to fit a mean SUV6 profile to a mean CTD profile and using it to calibrate the rest of the SUV6 data.

A small complication was that within the SUV6 scatter there appeared to be at least 3 different overall patterns to the profiles. These were deemed to be real differences in the data, not artefacts of the offsetting in calibration as the shapes of the curves differed somewhat; one group of profiles even showing a tendency towards maximum nitrate values at ~200m

which then decreasing between 200-450m. Only one CTD sample profile was found which also exhibited this shape, however. The majority of CTD sample profiles fitted in with the main trunk of SUV6 data that recorded a relatively constant concentration with depth below 100m. Had there been a significant number of CTD profiles displaying each of the patterns seen in the SUV6 data, it may have been possible to fit each set of SUV6 profiles to independent mean CTD profiles. thus preserving their shape. This, however, was not the case.

Twenty points ranging from 450m to the surface were chosen within the SUV6 nitrate vs depth scatter that best represented a mean profile. It was important to include at least two or three points between 30-50m where the gradient in nitrate concentration was greatest. The same was done with the CTD sample nitrate vs depth scatter, using the same depths for the location of the points, and a mean profile was also determined. Then, whilst keeping the two plots overlaid and taking one point at a time, the plots were shifted along the nitrate (x) axis until the point in the SUV6 mean profile overlaid the position of the same-depth point in the mean CTD profile.

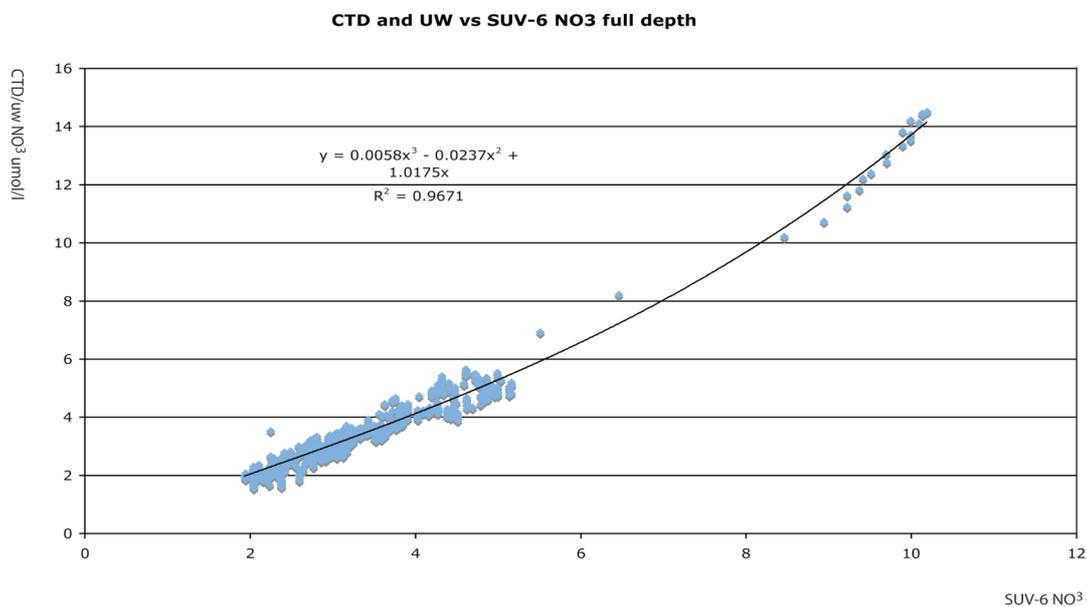


Figure 8. SUV6 nitrate vs CTD bottle nitrate at 20 points within the SUV6 nitrate mean profile, having been offset to overlay the CTD mean profile. The surface calibrated data from Figure 7 are included to increase the number of data points to which the trend line was fitted.

In this way, a pair of readings was taken for each depth point; the first was the nitrate concentration at the original position in the SUV6 mean profile (taken from x axis of the SUV6 nitrate vs depth plot). The second value was the nitrate concentration at the new position within the CTD scatter (taken from the x axis of the CTD nitrate vs depth plot). This process was repeated for each of the twenty points in the SUV6 mean profile. The pairs of points were then tabulated and plotted against each other, (see Figure 8).

The number of data points was increased by adding in the surface calibrated SUV-6 data plotted against underway nitrate concentration from Figure 7. It was found through a series of statistical tests using polynomial functions of increasing order that a cubic function best fitted the data. In view of the fact that the surface data had already been calibrated and did not need further adjustment, the cubic function was forced through zero so as not to produce an offset.

The SUV6 data was thus calibrated according to the following function, where x is the SUV6 nitrate concentration:

$$y = 0.0058 x^3 - 0.0237 x^2 + 1.0175 x$$
$$R^2 = 0.99671$$

Figure 9 shows the whole SUV-6 dataset vs depth profile following surface and depth calibrations. Also plotted are the CTD bottle data as before. The overlay is satisfactory and at this point, the SUV6 data are considered fully calibrated to a first order for scientific purposes. The fully calibrated SUV6 data was then merged on time with the master SeaSoar file to pull in corresponding Minipack CTD parameters. The master SeaSoar file had already been fully calibrated with respect to temperature, salinity, oxygen (SBE 43 and Anderaa Optode sensors) and fluorescence (Chelsea minipak fluorometer and Turner Designs Cyclops fluorometer) in post-cruise processing. Dissolved nitrate could then be plotted out in contoured sections for each of the five tows and compared with other biogeochemical and physical parameters in such a fine-scale way as has not been seen before.

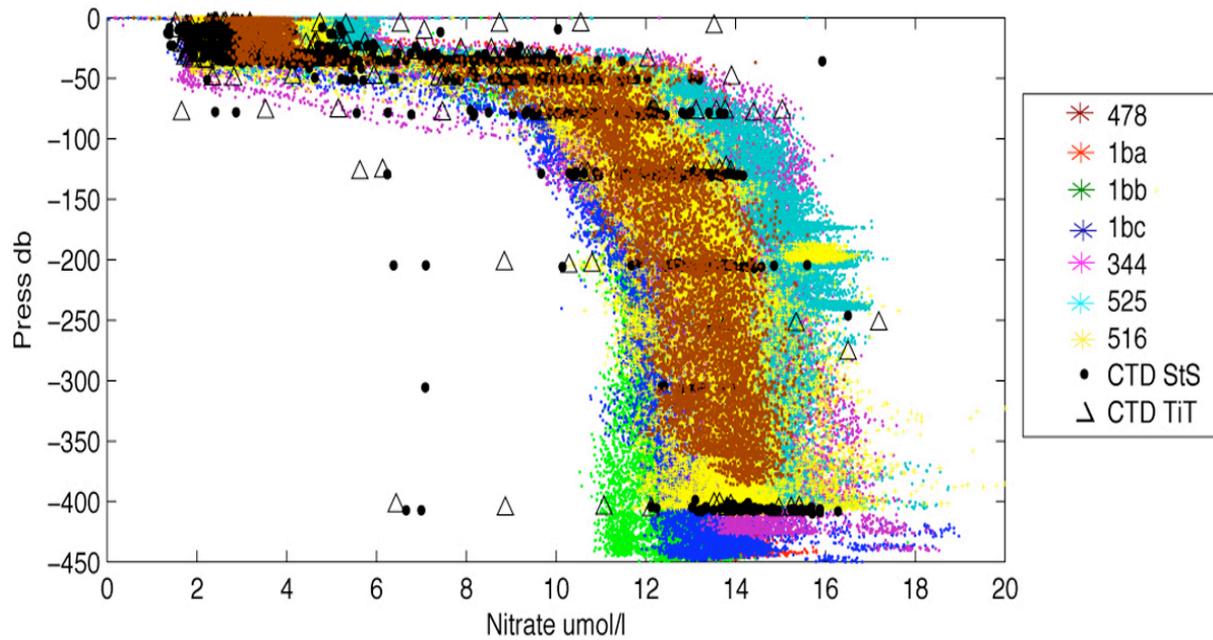


Figure 9. SUV6 data calibrated to surface and to depth plotted against CTD nitrate data down to 450m. SUV6 data is colour-coded by tow according to the key. Stainless steel CTDs are marked with circles, titanium casts are marked with triangles.

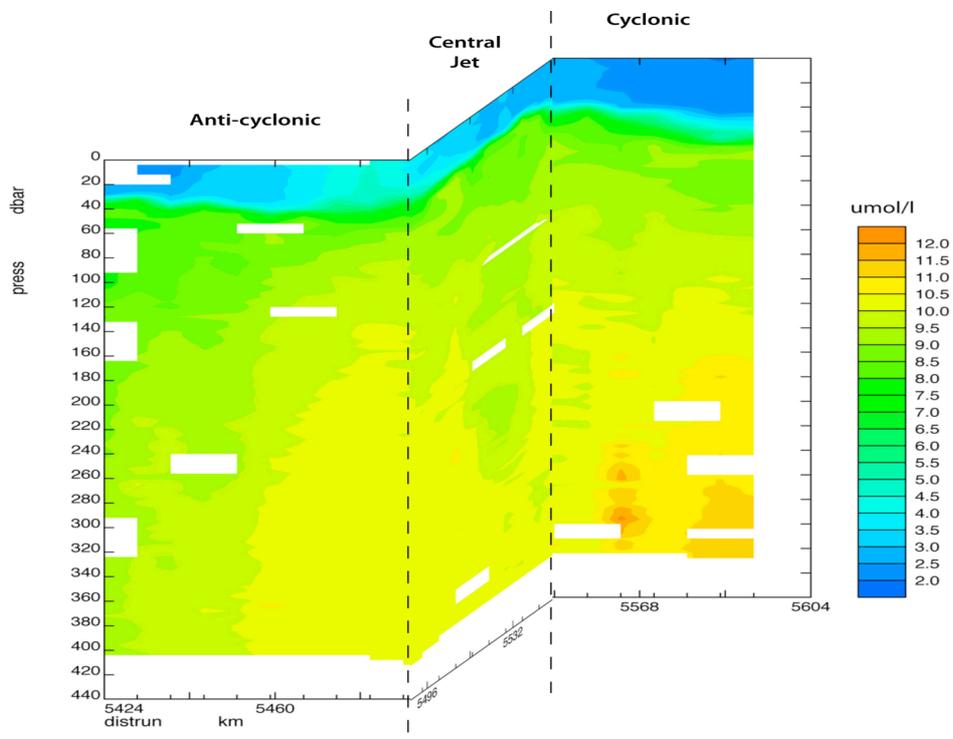


Figure 10. SUV-6 nitrate section for the dogleg tow showing cyclonic and anti-cyclonic components of the dipole and the central jet

Figure 10 shows a contoured nitrate section for the targeted dogleg tow through both eddy components of the dipole, (tow 2c). It clearly shows the fine-scale variability in the water column. Raised isonutrient surfaces characterise both eddy cores down to 400m whilst maximum concentrations of $12 \mu\text{mol l}^{-1}$ are found on the periphery of the cyclonic eddy. This may be evidence of small-scale nutrient injection resulting from instability processes.

Together with targeted net hauls and collected water samples, on-going work continues to assess the biogeochemical impacts of the dipole feature.

Reference

Quartly G.D. 2006 Madagascar Experiment (MadEx), 26 Jan – 21 Feb 2005, RRS *Discovery* Cruise 288, Cruise Report No. 8, National Oceanography Centre, Southampton, UK, 105pp.

Appendix – Matlab code used to process SUV6 data

```
% suv6dat2nitr.m - Matlab file to convert suv6 raw data to nitrate values
% R. Prien - 27.01.2005

% NB Program assumes data file to have calibration values for at least the last
%   three rows. Edit files to ensure this. EJJ 12.02.05

% modified for SeaSoar SUV6 by MAS 05.08.07

% data come in batches of 60 lines - 45 actual data and 15 calibration data
% note that data file can start and end anywhere within that batch of 60

clear
close all

files = [
'suv6516.000' % Your filename goes in here
];
nrem = 3; % number of points to remove at the
flipping edges
colors = ['m','b','c','g','y','r','k']; % colors for channels
syms = ['+', '.', 'o']; % symbols used
thres = 1400; % threshold to determine flipping - changed from 2000 to 1400 for
SeaSoar
calch = 1; % channel for detection of flipping

ind = find(files(1,:)=='');
eval(['load ',files(1,:)])
eval(['I = ',files(1,1:ind-1),';'])
eval(['clear ',files(1,1:ind-1)])

% for the files taken with Jon Campbells software the first two columns
% are year and Julian day with fraction which gives the time
% modified to remove year as not in SeaSoar data stream
% SeaSoar data stream contains JDAY and 7 SUV6 channels
```

```

jdaytime = I(:,1);
I = I(:,2:7);    %%<<<<<<< remove the day column

% for debugging only:
%jdaytime = 27.6 + (1:length(I))./24./3600;

sec = (jdaytime - jdaytime(1)).*24.*3600;

figure(1) % plot cal-channel raw data and detected flips
set(gcf,'Name','raw cal chan');
subplot(2,1,1)
plot(sec,I(:,calch),[num2str(colors(2))])
tit = [files(1,:),': ','raw data, ch. ',num2str(calch)];
title(tit)
ylabel('I / counts'),xlabel('time / s ')
hold on

% remove nrem values next to edges
% falling edges
miin = find((I(1:length(I)-1,calch)>thres)&(I(2:length(I),calch)<thres)));
ind=1:length(miin);
plot(sec(miin(ind)+1),I(miin(ind)+1,calch),'k+') % show detected falling edge
subplot(2,1,2)
plot(diff(sec(miin(ind)+1)), 'b') % show timing of detected falling edges
hold on
% remove values next to edge
for j=1:length(ind)
    I(miin(ind(j))+1:miin(ind(j))+nrem(1)+1,:)=NaN;
    sec(miin(ind(j))+1:miin(ind(j))+nrem(1)+1)=NaN;
end
ind=find(isnan(I(:,1)));
I(ind,:) = [];
ind=find(isnan(sec));
sec(ind) = [];

% rising edge
mout = find((I(1:length(I)-1,calch)<thres)&(I(2:length(I),calch)>thres)));
ind=1:length(mout);
subplot(2,1,1)
plot(sec(mout(ind)+1),I(mout(ind)+1,calch),'ko') % show detected rising edge
subplot(2,1,2)
plot(diff(sec(mout(ind)+1))+0.01,'r') % show timing of rising edges
xlabel('flip No. '),ylabel('time / s ')
title('Time between flips')
legend('falling','rising')
for j=1:length(ind)
    if( (mout(ind(j))-nrem(1)+1) > 0)
        I(mout(ind(j))-nrem(1)+1:mout(ind(j))+1,:)=NaN;
        sec(mout(ind(j))-nrem(1)+1:mout(ind(j))+1)=NaN;
    end
end
ind=find(isnan(I(:,1)));
I(ind,:) = [];
ind=find(isnan(sec));
sec(ind) = [];

subplot(2,1,1)
plot(sec,I(:,calch),'m') % show the raw-data after removal of values at edges

miin = find(I(:,calch)>thres); % time values with mirror in the path
mout = find(I(:,calch)<thres); % time values with mirror out of path

Imin = I(miin,:); % all values with mirror in the path (i.e. internal
calibration)
Imout = I(mout,:); % all values with mirror out of the path (i.e. measurement)

%%

```

```

%% Fig. 2 line-graphs for the two 'flip' positions
%%
figure(2)
set(gcf,'PaperUnits','centimeters');
set(gcf,'PaperPosition',[0 0 30 21]);
set(gcf,'PaperOrientation','landscape');
set(gcf,'Name','raw all chan.');
```

```

subplot(2,1,1)
t = sec(miin)./60; % time in min for internal calibration values
h = plot(t,Imin,'-');
for ii=1:6
    set(h(ii),'Color',num2str(colors(ii)))
end
title([files(1,:)':' int. path'])
ylabel('I / counts'),xlabel('time / min ')
legend('205','220','235','250','265','280')
grid
```

```

subplot(2,1,2)
t = sec(mout)./60; % time in min for measurement values
h = plot(t,Imout,'-');
for ii=1:6
    set(h(ii),'Color',num2str(colors(ii)))
end
ylabel('I / counts'),xlabel('time / min ')
title([files(1,:)':' ext. path'])
legend('205','220','235','250','265','280')
grid
eval(['print -dpsc ',files(1,1:length(files(1,:))-4),'_raw.ps'])
```

```

% Let's find the mean values and standard deviations for all periods (in and out)
% First for int. cal. position
t = sec(miin); % time in secs again for int. cal. position
ind = find( diff(t) > 10); % this find indices 28 56 84 etc. (for nrem = 3) which
are the last indices of an interval
ind = [1; ind+1 ;size(t,1)+1]; % Now the indices are starts of intervals, only the
last one is last index of series +1
Iminmean(length(ind)-1,6)=0; % create matrix of means
Iminstd(length(ind)-1,6)=0; % and standard deviations
for j=1:length(ind)-1
    Iminmean(j,1) = mean( Imin(ind(j):ind(j+1)-1,1) );
    Iminstd(j,1) = std(Imin(ind(j):ind(j+1)-1,1) );
    Iminmean(j,2) = mean( Imin(ind(j):ind(j+1)-1,2) );
    Iminstd(j,2) = std(Imin(ind(j):ind(j+1)-1,2) );
    Iminmean(j,3) = mean( Imin(ind(j):ind(j+1)-1,3) );
    Iminstd(j,3) = std(Imin(ind(j):ind(j+1)-1,3) );
    Iminmean(j,4) = mean( Imin(ind(j):ind(j+1)-1,4) );
    Iminstd(j,4) = std(Imin(ind(j):ind(j+1)-1,4) );
    Iminmean(j,5) = mean( Imin(ind(j):ind(j+1)-1,5) );
    Iminstd(j,5) = std(Imin(ind(j):ind(j+1)-1,5) );
    Iminmean(j,6) = mean( Imin(ind(j):ind(j+1)-1,6) );
    Iminstd(j,6) = std(Imin(ind(j):ind(j+1)-1,6) );
    tminmean(j) = mean( t(ind(j):ind(j+1)-1) );
end
```

```

% Second for the measurement position:
t = sec(mout); % time in secs again
ind = find( diff(t) > 8); % this finds the last indices of an interval
ind = [1; ind+1 ;size(t,1)+1]; % Now the indices are starts of intervals, only
last one is last index of series
```

```

Imoutmean(length(ind)-1,6)=0; % create matrix of means
Imoutstd(length(ind)-1,6)=0; % and standard deviations
Imoutrelstd(length(ind)-1,6)=0;
Imoutrel(size(t,1),6) = 0; % create matrix of ext. path values normalized with
following int. path interval
Imoutrelmean(size(t,1),6) = 0;
for j=1:length(ind)-1
    Imoutmean(j,1) = mean( Imout(ind(j):ind(j+1)-1,1) );
```

```

Imoutrel(ind(j):ind(j+1)-1,1) = Imout(ind(j):ind(j+1)-1,1)./Iminmean(j,1);
Imoutrelmean(j,1) = mean( Imoutrel(ind(j):ind(j+1)-1,1));
Imoutstd(j,1) = std( Imout(ind(j):ind(j+1)-1,1));
Imoutrelstd(j,1) = std( Imoutrel(ind(j):ind(j+1)-1,1));
Imoutmean(j,2) = mean( Imout(ind(j):ind(j+1)-1,2) );
Imoutrel(ind(j):ind(j+1)-1,2) = Imout(ind(j):ind(j+1)-1,2)./Iminmean(j,2);
Imoutrelmean(j,2) = mean( Imoutrel(ind(j):ind(j+1)-1,2));
Imoutstd(j,2) = std( Imout(ind(j):ind(j+1)-1,2));
Imoutrelstd(j,2) = std( Imoutrel(ind(j):ind(j+1)-1,2));
Imoutmean(j,3) = mean( Imout(ind(j):ind(j+1)-1,3) );
Imoutrel(ind(j):ind(j+1)-1,3) = Imout(ind(j):ind(j+1)-1,3)./Iminmean(j,3);
Imoutrelmean(j,3) = mean( Imoutrel(ind(j):ind(j+1)-1,3));
Imoutstd(j,3) = std( Imout(ind(j):ind(j+1)-1,3));
Imoutrelstd(j,3) = std( Imoutrel(ind(j):ind(j+1)-1,3));
Imoutmean(j,4) = mean( Imout(ind(j):ind(j+1)-1,4) );
Imoutrel(ind(j):ind(j+1)-1,4) = Imout(ind(j):ind(j+1)-1,4)./Iminmean(j,4);
Imoutrelmean(j,4) = mean( Imoutrel(ind(j):ind(j+1)-1,4));
Imoutstd(j,4) = std( Imout(ind(j):ind(j+1)-1,4));
Imoutrelstd(j,4) = std( Imoutrel(ind(j):ind(j+1)-1,4));
Imoutmean(j,5) = mean( Imout(ind(j):ind(j+1)-1,5) );
Imoutrel(ind(j):ind(j+1)-1,5) = Imout(ind(j):ind(j+1)-1,5)./Iminmean(j,5);
Imoutrelmean(j,5) = mean( Imoutrel(ind(j):ind(j+1)-1,5));
Imoutstd(j,5) = std( Imout(ind(j):ind(j+1)-1,5));
Imoutrelstd(j,5) = std( Imoutrel(ind(j):ind(j+1)-1,5));
Imoutmean(j,6) = mean( Imout(ind(j):ind(j+1)-1,6) );
Imoutrel(ind(j):ind(j+1)-1,6) = Imout(ind(j):ind(j+1)-1,6)./Iminmean(j,6);
Imoutrelmean(j,6) = mean( Imoutrel(ind(j):ind(j+1)-1,6));
Imoutstd(j,6) = std( Imout(ind(j):ind(j+1)-1,6));
Imoutrelstd(j,6) = std( Imoutrel(ind(j):ind(j+1)-1,6));
tmoutmean(j) = mean( t(ind(j):ind(j+1)-1) );
end

```

```

%%
%% Fig.4: ext. path values normalized with following int. path interval
%%

```

```

figure(4)
set(gcf,'PaperUnits','centimeters');
set(gcf,'PaperPosition',[0 0 30 21]);
set(gcf,'PaperOrientation','landscape');
set(gcf,'Name','normalized ext. path');

t = sec(mout)./60; % time in min for mirror out values
h = plot(t,Imoutrel,'-');
for ii=1:6
    set(h(ii),'Color',num2str(colors(ii)))
end
ylabel('I_{ext} / I_{int}'),xlabel('time / min ')
title([files(1,:)': ext. path, calibration in LNS'])
legend('205','220','235','250','265','280')
grid
eval(['print -dpsc ',files(1,1:length(files(1,:))-4),'_extnorm.ps'])

```

```

%%
%% Fig.5: means and standard deviation as error bars for each interval
%%

```

```

figure(5)
set(gcf,'PaperUnits','centimeters');
set(gcf,'PaperPosition',[0 0 30 21]);
set(gcf,'PaperOrientation','landscape');
set(gcf,'Name','means + std dev');

% first for int. path (upper panel)
subplot(2,1,1)
hold on
[dum tminmat] = meshgrid(1:6,tminmean);
h = errorbar(tminmat./60,Iminmean,Iminstd);
for ii=1:6
    set(h(ii),'Color',num2str(colors(ii)))
end

```

```

legend('205','220','235','250','265','280')
grid
ylabel('mean {I(\lambda)} / counts'),xlabel('time / min')
title([files(1,:)':' int. path'])
ax = axis;
ax(1)=0;
axis(ax);

% then for ext. path (lower panel)
subplot(2,1,2)
[dum tmoutmat] = meshgrid(1:6,tmoutmean);
hold on
h = errorbar(tmoutmat./60,Imoutmean,Imoutstd);
for ii=1:6
    set(h(ii),'Color',num2str(colors(ii)))
end
legend('205','220','235','250','265','280')
ax = axis;
ax(1)=0;
axis(ax);

grid
ylabel('mean {I(\lambda)} / counts'),xlabel('time / min')
title([files(1,:)':' ext. path'])
eval(['print -dpsc ',files(1,1:length(files(1,:))-4),'_means.ps'])

% if(length(tmoutmat)>length(tminmat))
%     tmoutmat(length(tmoutmat),:) = [];
%     Imoutmean(length(Imoutmean),:) = [];
%     Imoutstd(length(Imoutstd),:) = [];
% end
% if(length(tminmat)>length(tmoutmat))
%     tminmat(length(tminmat),:) = [];
%     Iminmean(length(Iminmean),:) = [];
%     Iminstd(length(Iminstd),:) = [];
% end

% from the lab calibration:
Ioff1 = 1000 .* [0.73723096989466      0.93689932661418      2.47702213544662
2.32764682034082      2.74370449575342];
Ioff2 = [1.04467450289282      0.33928188910878      0.82393639046140      0.78263023349615
0.92605517522418];
p2 = [ 38.20596460696951 -13.78382751978668];

ab2 = -log((Imoutrel(:,2)./Imoutrel(:,6))-Ioff2(2))); %absorbance of 220 nm channel

figure(9)
set(gcf,'PaperUnits','centimeters');
set(gcf,'PaperPosition',[0 2 21 25]);
set(gcf,'PaperOrientation','portrait');
set(gcf,'Name','conc. appl. cal.');
```

```

%save time as jday
timereal=(sec(mout)/(60*60*24))+jdaytime(1);

h = plot(sec(mout)./60,polyval(p2,ab2),'b');
title(['Concentration calculated from calibration (220 nm channel)'],'FontSize',14)
xlabel('time / min','FontSize',14),ylabel('concentration / \mu mol l^{-1}','FontSize',14)
set(gca,'FontSize',14)
grid

figure(10)
h = plot(timereal,polyval(p2,ab2),'b');
title(['Concentration calculated from calibration (220 nm channel)'],'FontSize',14)
xlabel('time / Jday','FontSize',14),ylabel('concentration / \mu mol l^{-1}','FontSize',14)
set(gca,'FontSize',14)
grid

```

```
eval(['print -dpsc ',files(1,1:length(files(1,:))-4),'_linregbc.ps'])

%save 6 channel data
eval(['channels_',files(1,1:length(files(1,:))-4),'=[timereal Imoutrel];'])
eval(['save channels_',files(1,1:length(files(1,:))-4)]);

% save nitrate values
eval(['Nitr_',files(1,1:length(files(1,:))-4),'= [timereal polyval(p2,ab2)];'])
eval(['save Nitr_',files(1,1:length(files(1,:))-4)]);
```