n2XX
CoRot n2 data eXplorer/eXtractor
Manual V1.2a

A program for preliminiary analysis & extraction of CoRoT n2 data

---

Michael Gruberbauer

Insitute for Astronomy
University of Vienna

Contact
gruberbauer@astro.univie.ac.at

February 28, 2008

Contents

1 Version History

Version	Notes
1.0	First release including all basic features
1.0a	Bug fixing
1.0b	Bug fixing - stable version
1.1	Added zooming feature, added "Show Header(s)"-functionality
1.2	Added ability to open and work with multiple files at once, added DFT calculation, exchanged magnitude transformation with relative intensity transformation, minor bug fixing, made command line accessible while using the program
1.2a	Reading performance greatly improved (about a factor of 5-10), added coordinate display for DFT window, fixed minor compatibility issues

2 Introduction

2.1 Motivation

The CoRot mission as a new cornerstone mission in asteroseismology and planet-finding from space will (and does already) deliver unprecedented amounts of high quality data. Even better, the data is reduced/processed (e.g. correction for instrumental effects) and evaluated even before it reaches the analyst. This culminates in the so called ”n2 data” product, which is supposed to deliver data ready for analysis. However, high quality measurements do not remove the scientist from the equation, and the better an automated data reduction pipeline is claimed to perform, the more caution scientists have to exhibit when getting their hands on the data.

As such, although some users of the ”n2 data” product might feel an urge to simply extract the final light curve from the provided files, most astronomers working with CoRot data still want to examine the light curves at various stages of reduction. Since these are of course also provided with the ”n2 data” product, there is a need for a simple tool that allows one to explore and compare data at these different stages, to explore the fits headers and the telemetry and, finally, to extract exactly those kind of data one is interested in - all within a single program.

To my knowledge there are currently a number of useful scripts available that can be used to perform most of the steps I described above. However, there has not been any publicly accessible, complete package with a graphical user interface, which makes these steps more intuitive and less time-consuming. This is the goal I had in mind for the development of **n2XX**.

2.2 Development

The development of **n2XX** is still an ongoing process. While features, like those mentioned above, are already present, it would be even more useful to perform tasks like detrending and other basic steps of time series analysis, phase dispersion minimisation, etc. right within the program. Still, it only makes sense to supply if there is demand. Therefore, if you find this program useful, have other comments and suggestions, or have found a bug, please consider contacting me by email.

3 Program design

3.1 Requirements

n2XX is an IDL application and therefore needs IDL to run. This confines the usage of the program to machines that are powerful enough to run IDL, have an appropriate operating system, and, in the end, have access to a valid IDL user's licence. IDL does not necessarily need a user's licence - if such is not available, it enters a "demo mode". Within this mode, however, the file I/O is restricted. Thus, I do not recommend using **n2XX** without an IDL license, since major features of the application will not work. Additionally, **n2XX** uses some routines from the **"IDL Astronomy User's Library" (http://idlastro.gsfc.nasa.gov)**. The program has been tested on various Linux distributions and runs comparably well on such computers. On Microsoft Windows machines, however, a few problems may come up (see "Troubleshooting"). The program has not been tested with **GDL (http://gnudatalanguage.sourceforge.net)**, but it makes use of graphical user interface elements (widgets), which are not yet implemented in this open source project.

3.2 Handling CoRoT data

One of the main features of **n2XX** is that it can read any FITS file conforming to the current CoRoT FITS standard. It makes use of some pre-defined keywords and other definitions, which can be found in the CoRoT FITS documentation. For instance, the clipping feature, which provides the possibility to exclude data points that meet specific criteria, relies on the definition of the bitmask in the "Status" fields.

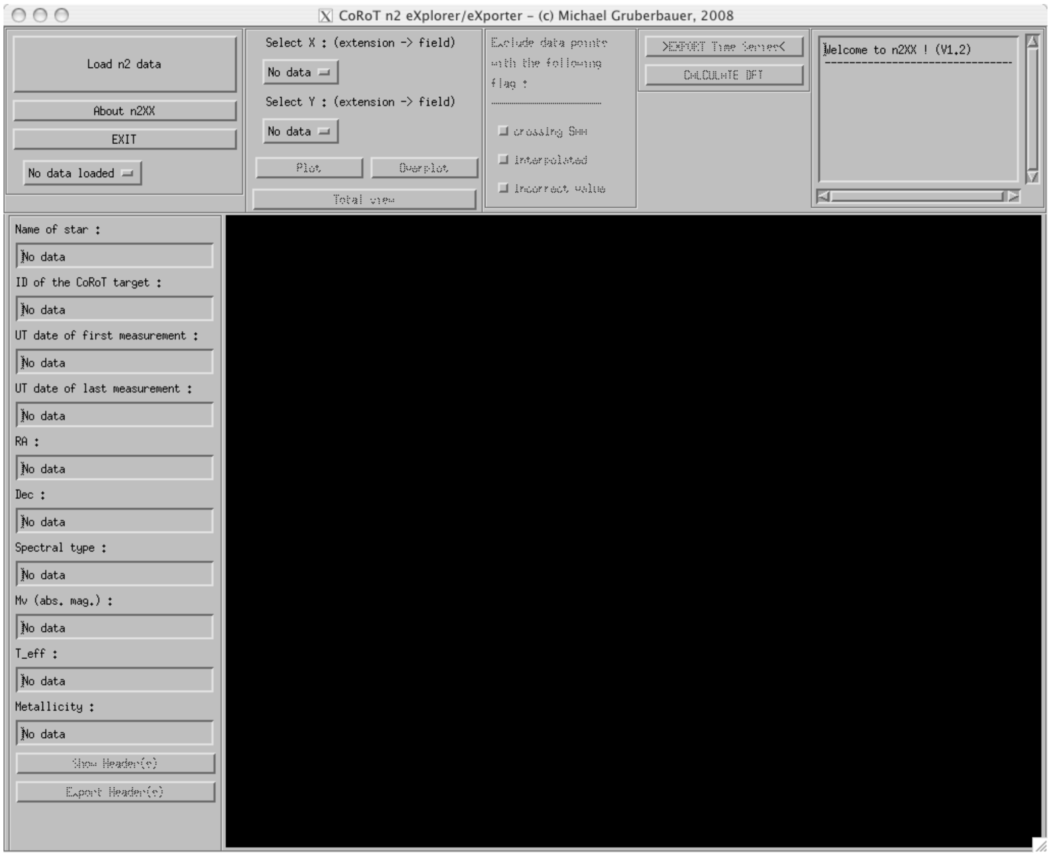
In general, there is no restriction to data from either the exoplanet field or the asteroseismology field, although these files have a different structure. In order to cope with these different structures (e.g. multiple binary table extensions), **n2XX** relies on some features of object-oriented programming, which provide the possibility of highly dynamical data handling. Therefore, there are no hard-coded limitations on how many extensions (and data columns per extension) a CoRoT FITS file should contain.

Nonetheless, compatibility with future revisions of the CoRoT FITS standard cannot be guaranteed. If the overall format changed radically (e.g. no binary tables but two-dimensional images), adjustments to the program will have to be made. In such a situation (and when it has been brought to my attention) the program will be updated as soon as possible.

3.3 User interface

Command line interfaces can be much faster than graphical user interfaces. But as soon as one exceeds a threshold of complexity in the various input parameters, or a specific number of command line tools to call in sequence, the command line is clearly the wrong

Figure 1: The main window



```
Welcome to n2XX ! (V1.2)
--------------------------------
```

Name of star : No data
ID of the CoRoT target : No data
UT date of first measurement : No data
UT date of last measurement : No data
RA : No data
Dec : No data
Spectral type : No data
Mv (abs. mag.) : No data
T_eff : No data
Metallicity : No data

choice. Without visual feedback, processes often have to be repeated until a desired result is achieved. Thus, **n2XX** has been designed to have a simple graphical user interface, entirely realised through the widgets of IDL.

Fig. 1 shows the initial window that **n2XX** users will see. It already contains most of the control elements that will receive user input. The window is divided into three sections:

1. **The top panel**
   It provides the user with the basic options (like loading data and quitting the program) and handles most of the data plotting-specific input. The top panel also contains a checklist which is used to exclude specific data points, and an operations panel that currently shows the 'Export time series' option, as well as the 'Calculate DFT' option. Lastly, a text field (the "info window") has been implemented to provide additional information.

2. **The side panel**
   This presents some details about the observations and the target, taken from the FITS header of the CoRoT data file. It also hosts two buttons, which can be used to either show the header(s) in a separate window, or to export it (or them) to an ASCII file.
3. **The canvas**
   This black box (see Fig. 1) is used to plot the data, as well as to interact with the plot using the mouse (e.g. clicking an dragging a box in order to zoom in).

4 Using n2XX

4.1 Starting the program

To call **n2XX**, the user has to start IDL. Within the IDL interpreter (`IDL>`), make sure that IDL can access the source code of the program, and then just type

```
IDL> .compile n2xx.pro
```

or

```
IDL> .compile n2xx
```

All routines within the source are then compiled. If one encounters an error at this stage, it is most likely due to missing routines from the Astronomy User's Library (see Section 3.1). The application can be started by typing

```
IDL> n2xx
```

4.2 Using the program

4.2.1 Loading data - UPDATED

Using **n2XX** is very simple. Initially, CoRoT data has to be loaded by pressing *Load n2 data*. A new window will pop up, asking the user to select the n2 file (in FITS format) that should be read. **UPDATE : Since V1.2, the user can also specify multiple files to load by using SHIFT or CTRL and selecting multiple file names.** After selecting the wanted file(s) and clicking ok, the info window will display a message that loading has started. If the data contains flux measurements, the program asks, whether these measurements should be converted to relative flux. The program uses the simple formula

$$f_{rel} = f/\overline{f} \tag{1}$$

where f are the flux values read from the CoRoT data and $\overline{f}$ is their mean value. Should the user accept the conversion to relative flux, all other operations (including exporting the data) will use the values from the transformation in equation 1. In order to get back to the original flux measurements, the data have to be loaded again.

As soon as the loading procedure is completed, the info window will contain a corresponding message.

4.2.2 V1.2 : Exploring data: Working with multiple data sets

Since V1.2, directly below the 'EXIT' button there is a droplist (I will be calling it 'starlist' in this manual) giving access to all individual data sets which are currently loaded. The stars are referenced by either 'CoRoT ID' followed by their CoRoT ID, or, if the information about this value is missing in the data, simply by 'Data set' followed by the number of the data set. By selecting a single star from the list, this star becomes the active object, and all information on screen is replaced by information about the current star. Any of the possible features (producing new plots, exporting data/headers, DFT) will only use the data from the active object.

4.2.3 Exploring data: Header information

n2XX gives a lot of information about a specific data set at one glance. First off, the side panel shows a few of the more important keywords from the CoRoT FITS headers (e.g., name of the star, assumed effective temperature, date of observation, etc.). Additionally, the whole header can be inspected by pressing *Show Header(s)* at the bottom of the side panel. In this case a new window opens, and all ASCII headers of all FITS extensions in the CoRoT FITS file will be displayed, one after another, in a scrollable text box. If present, various headers of different extensions will be separated from each other by a line of dashes.

4.2.4 Exploring data: Plotting - UPDATED

The program makes individual data sets within one CoRoT n2 FITS source accessible for plotting. Individual plots can be created by choosing data sets (corresponding to the x-axis and y-axis) via the drop lists in the top panel, an pressing the *Plot*-button (e.g. x = raw stellar flux and y = background - try to find a correlation). Once a plot has been produced, the user is able to zoom to any region. This is done by clicking and dragging a rectangle around the desired region in the plot. Holding the CTRL key and clicking the mouse button undoes the most recent zoom step. In addition, a button in the top panel named *Total view* resets the axis so that the whole data set is visible again.

In case the user wants to compare different data sets in the same plot, an overplotting function has been implemented. If a regular plot has already been created, the user can select any additional x/y-axis combination (e.g. x = reduced stellar flux and y = background) and press *Overplot* to have both data sets in the same plot. The new data set will automatically get a different color in order to distinguish between the graphs. This can be done up to 9 times until the color sequence will be repeated. Zooming works just like described above, when multiple data sets are shown. Overplots, however, will produce a plot that contains all the data points of all data sets - therefore, any zooming prior to overplotting will be undone. **UPDATE : As of V1.2, the light curves of different**

stars can also be plotted simultaneously. The user first has to select a specific star A via the starlist in the top panel and then produce a plot. In order to include a different star B in the plot, selecting the data set for star B via the starlist and subsequently clicking 'Overplot' should do the job.

If the user does not want to compare different data sets anymore, pressing *Plot* will again produce a plot only containing the currently selected x/y-axis data sets of the currently active object.

4.2.5 Exploring data: Excluding data points

In the top panel a checklist ("Exclude data points with the following flag:") can be used to set restrictions for plotting and exporting data points, as well as for the calculation of a DFT. Three cases are currently implemented :

1. crossing SAA - exclude data points that were produced while the satellite was influenced by the South Atlantic Anomaly
2. interpolated - exclude data points that were produced by linear interpolation to fill up unwanted gaps in the light curve
3. incorrect values - exclude data points that are marked as "incorrect"

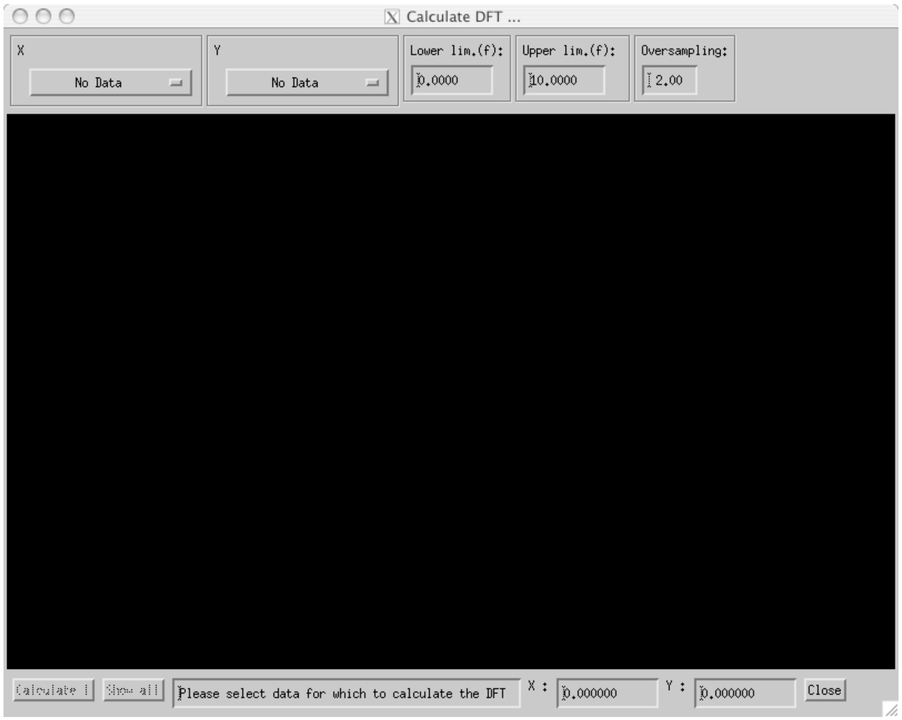
If one ore more of the checklist items are selected, the "info box" will give a summary about how many data points are affected by the selection. Additionally, each following plot (or overplot) AND all data sets exported afterwards will only contain data points that do not meet the selected criteria. As an example : take a light curve and plot the raw flux. Then check the "crossing SAA" box and press overplot. A plot, showing both, the raw flux and the raw flux without SAA-affected data, will be on the canvas.

Attention: Internally, the program only keeps two versions of each data set : the original data set, and a copy to which the currently selected criteria have been applied. Therefore, plotting of data sets with different criteria does not work. As an example : suppose that the user selects the "crossing SAA" criterion and then produces a plot. Then he or she activates an additional criterion, "interpolated", and clicks the *Overplot* button. The result will be two light curves which are exactly the same, where all interpolated and SAA-data points have been removed.

4.2.6 NEW : Exploring data: Calculating a DFT

Since V1.2, **n2XX** also provides the possibility to calculate a Discrete Fourier Transform (DFT). After clicking >*Calculate DFT*< in the top panel of the main window, a new window opens (see Fig. 2). In this DFT window, the user can specify the data to use for

Figure 2: The DFT window



Calculate DFT ...

X: No Data

Y: No Data

Lower lim.(f): 0.0000

Upper lim.(f): 10.0000

Oversampling: 2.00

Please select data for which to calculate the DFT

X : 0.000000

Y : 0.000000

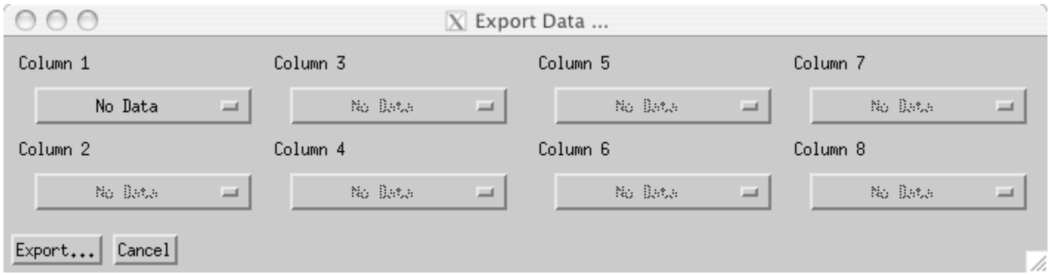
the calculation, set the lower and upper limits in frequency, and set the oversampling of the spectrum. By clicking on *Calculate !* the DFT calculation is started, while the info text at the bottom of the DFT window shows continuously updating progress information. Another window including a button named *Abort calculation* will also pop up to enable the user to abort the calculation. As soon as the calculations are finished, the resulting spectrum is plotted on the canvas in the centre of the DFT window. The plot can then be explored just like in the main program, using the mouse to zoom, CTRL-click to unzoom once, or the 'Show all' button to get back to the total view.

The DFT is calculated in a similar fashion to what is described in [1]. The algorithm has been somewhat optimised for the specifications of IDL. Still, since IDL is much slower than C or FORTRAN in terms of 'brute force' computing performance, the calculation will take a considerable amount of time. This is why, per default, a rather low oversampling factor (which defines the step width in frequency, $\Delta f = 1/(\Delta T \cdot os)$, where ΔT is the length of the data set, usually in days, and os is the oversampling factor) is set, although the value can also be specified by the user.

Alas, I am afraid that I cannot get this routine to run any faster (I'm happily open to suggestions though) under 'native' IDL. I propose that for a quick overview an oversampling factor of 2 is sufficient. For greater detail a higher value can be set, but the frequency range should be decreased as much as possible, in order to retain a bearable computing time.

4.2.7 Exporting data

Figure 3: The export window



To export the header(s) of the CoRoT data, the user just has to click on *Export Header(s)* in the side panel. A new window opens, where the user has to select a file (or specify a filename for a new file) to write the ASCII header(s) to. All headers of all extensions in the CoRoT FITS-source will be written to this file, and the output is formatted as described in Section 4.2.3.

The most useful feature of **n2XX**, eventually, is the ability to create custom multi-column ASCII files from CoRoT n2 data, which also take into account the restrictions described in Section 4.2.5. To export CoRoT data to such files, the user has to press >*EXPORT time series*<. The button label might be a bit misleading - the user can of course also create a multi-column ASCII file that does not contain a date/time column. After pressing the button, a new window opens (see Fig. 3). This window allows the assignment of individual data sets (just like in the drop lists used for setting the x- and y-axis for plotting) to up to 8 columns. The user just has to select the data set for the first column to unlock the second column. Once also a second column has been specified, the third column can be set, and so on. Once all desired data sets have been assigned to columns, pressing *Export ...* opens up a new window. After selecting a filename for the output and clicking *OK* the file is being written and the user is notified in the "info box" as soon as writing is completed.

Attention: Because data sets from different FITS extensions (e.g. raw flux from extension 1 and reduced flux from extension 2) most likely have a different number of data points, exporting such different data samples into the same file only works by using a little trick. **n2XX** searches for the data set with the most data points N_{max} among those that should be exported. It then creates an ASCII file with exactly N_{max} number of rows, in order to be able to write the longest data set completely. All shorter data sets are filled up with the required number of zeros to reach N_{max}.

5 Troubleshooting

Although **n2XX** is a pretty simple program, a few issues can occur. Here I hope to be able to provide some answers for questions that might come up. The following list will be updated if new problems surface.

5.1 Can't get it to work ?

1. Why does **n2XX** not work on my WINDOWS computer ?
   It probably does. Be sure to remove the one line in the code which says : DEVICE, TRUE=24. In future versions I will hopefully be able to implement checking the operating system version, so that such errors will automatically be dealt with. ***UPDATE V1.2*** *: This problem is now fixed, since the program checks which display system is used and adjusts the settings automatically.*

2. Why does **n2XX** not work on my WINDOWS/LINUX/MAC computer ?
   I probably have no idea. If, in the case of Windows, the above trick doesn't do the job, or in case of other operating systems, please send me a little "report" if you have the time. Just include the error message and I'll try to come up with something.

5.2 Troubled data...

1. I am not able to load my files with your program. I only get "Error - check file". Why is that ?
   Please make sure that you are trying to read an original n2 data set which conforms to the official CoRoT data conventions. If you cannot get it to work, most likely other programs will fail too, and you should report this to the people that do the data processing. If other programs can successfully read the file, it would be great if you could give me the version number of the CoRoT file.

2. When selecting a specific data set for plotting nothing happens when I click *Plot*.
   In some CoRoT data sets there seem to be non-numerical entries in the binary tables, which, per definition, cannot be plotted. As of now, ***n2XX*** *cannot distinguish between the various types when it comes to plotting.*

5.3 NEW : Troubled plotting...

1. What is happening with the zooming function under WINDOWS ?
   The WINDOWS display system seems to have some issues with the zooming routines. The event handling under WINDOWS (especially concerning the mouse click/motion/modifier key events) seems to cause problems, and the graphics buffering, which is used to handle zooming, might not work. Therefore, I currently do not recommend using the

zooming functions under WINDOWS since they might produce unforeseeable complications.

References

[1] Kurtz, D.W., 1985, MNRAS, 213, 773