

SUN/167.6

Starlink Project
Starlink User Note 167.6

V S Dhillon
G J Privett
K P Duffey

12th December 2001

PERIOD
A Time-Series Analysis Package
Version 5.0
User's Manual

Contents

1	Introduction	1
2	Initializing and Running	1
3	Data Format and Storage	1
3.1	Reading ASCII files as input	1
3.2	Reading OGIP FITS files as input	2
4	Using this Document	3
5	Menu Options	4
6	A Simple Recipe	12
7	Future Improvements	13
8	Acknowledgments	13
9	References	14
A	History	15

1 Introduction

This document describes how to use PERIOD (version 5.0), a software package designed to search for periodicities in data. Version 5.0 is the successor to 4.2, of which it is a Double Precision implementation, also re-configured to utilise dynamic array allocation/deallocation for the majority of its input, output and work arrays.

This document is intended for use with the UNIX software (version 5.0). If you have previously used a VMS version, you should be aware that the UNIX *is* case-sensitive with regard to directory structure and file names. For this reason the instructions below are in lower case and may appear odd to those familiar with VMS.

2 Initializing and Running

To run PERIOD under UNIX, you need only type the word `period` at the shell prompt.

3 Data Format and Storage

3.1 Reading ASCII files as input

The simplest form of input for PERIOD is an ASCII file. The files may contain any number of rows and columns. It is possible to specify which column refers to the x -axis, which column refers to the y -axis and which column refers to the y -axis errors. The y -axis errors are optional and if included are used (or handled) by all operations in the main PERIOD menu (see section 5). However, the only periodicity-finding option in the `period_period` sub-menu which actually uses the errors on data points is CHISQ (see section 5). The remaining five techniques (CLEAN, FT, PDM, SCARGLE, STRING) ignore errorbars if they are present. Note also that the input file must contain x -axis values which are in ascending order, otherwise the program will report a warning and either sort the data (if requested to do so) or abort the input.

Data is stored and processed within PERIOD using a *slot* system. A data slot is simply an array holding one dataset. The maximum number of data slots which can be handled at any one time in PERIOD is 40; hence PERIOD has the capability of analysing a large amount of data simultaneously.

The first command that is usually run in PERIOD is INPUT (see section 5), which loads datasets into data slots:

```
Enter first and last slots for input (0,0 to quit) :
```

In order to load the first slot with a single dataset, you should reply `1,1` to the above prompt. Similarly, if you want to load slots 4 through to 9 with 6 datasets, one should reply `4,9` to the above prompt. It is important to note that slots *can* be overwritten. Typing `0,0` will return the you to the menu.

Most PERIOD commands prompt not only for an input slot, but also for an output slot:

```
Enter first and last slots for input (0,0 to quit) :
Enter first and last slots for output (0,0 to quit) :
```

The input should contain the dataset (or datasets) and the output will contain the result of the operation on the dataset (or datasets). For example, given a set of 5 time-series (which have previously been loaded into slots 1 to 5 using INPUT) which need to be fitted with a sine curve (using the command FIT, see section 5), you would type 1,5 in reply to the first prompt and 6,10 in reply to the second. PERIOD will then fit sine curves to the data files loaded in slots 1 to 5 and put the fits in slots 6 to 10. Clearly, the number of output slots must be equal to the number of input slots. In addition, if any selected input slot is empty, PERIOD will abort the operation and return you to the main menu. It should be noted that in order to save on storage space you could have typed 1,5 in reply to the second prompt and the original data-files would have been overwritten by the resulting sine curves.

3.2 Reading OGIP FITS files as input

Because of the increasing use of OGIP standard FITS files within the X-ray community, PERIOD has been extended to allow FITS file table extensions where the FITS keyword HDUCLASS has the value OGIP.

As with ASCII data, data files may contain any number of rows. There is also no restriction on the number of FITS file extensions within the file to be read. Again, you must specify which columns of the table refers to the x -axis, which column refers to the y -axis and which column refers to the y -axis errors. The y -axis errors are optional and if included are used (or handled) by all operations in the main PERIOD menu (see section 5).

After you have entered the name of the file to be read, the file is opened and the keywords relating to telescope type etc are read. These are displayed so that you can make sure you are looking at the right file. The output will look something like this.

```
PERIOD> ogip

Enter name of OGIP FITS file (<CR> to quit) : test.fits

File details are...

Telescope:  ROSAT
Instrument:  PSPCC
Detector:    Unknown
Object:      XRT/PSPC PSF AR LA

Total number of extensions found:  4
Number of OGIP extensions found:  4

Extension  1
A binary table.
Size  3 by  2
EXTNAME is:  STDGTI
HDUCLAS1 is:  GTI
HDUCLAS2 is:  STANDARD
```

```

Extension  2
A binary table.
Size  8 by  29380
EXTNNAME is:  STDEVT
HDUCLAS1 is:  EVENTS
HDUCLAS2 is:  ACCEPTED

```

```

Extension  3
A binary table.
Size  8 by  1246
EXTNNAME is:  REJEVT
HDUCLAS1 is:  EVENTS
HDUCLAS2 is:  REJECTED

```

You are then asked to select the extension to be examined and are shown the header descriptions for each table column. From this information you can select which column should be in x -axis, y -axis or the y -axis errors part of your chosen data slot.

```

Enter the extension number to be considered (0 to quit) : 3
Column          Contents
  1              X
  2              Y
  3             PHA
  4              PI
  5             TIME
  6             DETX
  7             DETY
  8            STATUS
Enter the TTYPE column number to be read as X (0 to quit) : 1
Enter the TTYPE column number to be read as Y (0 to quit) : 2
Enter the TTYPE column number to be read as Y errors (0 to ignore) : 0

Enter the slot number into which the data should be read (0 to quit) : 1

Reading data into slot  1

Read  1246 out of  1246

** OK: Filled Slot=  1

```

After this you are given the option to reread the same document and fill a different slot.

While slightly more complicated to read than normal ASCII the program is still very easy to use.

4 Using this Document

There are two main sections in this document. Section 5 should be consulted by experienced users who require detailed information on individual PERIOD commands while Section 6 is intended for inexperienced users who require an introductory recipe detailing the steps that should follow when searching for periodicities in data.

Please note that *PERIOD* also has an extensive on-line help facility. This provides information at a level of detail similar to that found in section 5. However, it also provides a detailed description of the individual prompts, something which cannot be found in this document.

5 Menu Options

PERIOD is a menu-driven package. On entering PERIOD, you will be confronted with the following menu options, which are described in greater detail below.

```
|*****|
**| PERIOD  :>  A time-series analysis package |**|
**| Version :>  5.0 for UNIX                      |**|
**| Date    :>  12 December 2001                  |**|
|*****|
```

Options.

```
INPUT    --  Input ASCII file data.
OGIP     --  Input OGIP FITS table data
FAKE     --  Create fake data.
NOISE    --  Add noise to data.
DETREND  --  Detrend the data.
WINDOW   --  Set data points to unity.
OPEN     --  Open a log file.
CLOSE    --  Close the log file.
PERIOD   --  Find periodicities.
FIT      --  Fit sine curve to folded data.
FOLD     --  Fold data on given period.
SINE     --  +, -, / or * sine curves.
PLT      --  Call PLT.
STATUS   --  Information on stored data.
OUTPUT   --  Output data.
HELP     --  On-line help.
QUIT     --  Quit PERIOD.
```

PERIOD>

Any one of these commands can be entered by typing anything from the shortest unambiguous string up to the full command name. Therefore, P would be ambiguous, but PE would not.

INPUT

As described in section 3.1, this option allows you to input ASCII data into PERIOD. The routine determines the number of columns in the input files and then prompts the user for which columns refer to the *x*-axis, *y*-axis and *y*-axis errors (if desired, see section 3). For example, if the user is inputting radial velocity data, the *x*-axis would most probably be HJD's, the *y*-axis the heliocentric radial velocities and there would most likely be errors associated with each radial

velocity value. Note that the x -axis values must be in ascending order, otherwise INPUT will report a warning and either sort the data (if requested to do so) or abort. Note also that the y -axis errors are used by all options in the main PERIOD menu, but by only the CHISQ periodicity-finding option in the period_period sub-menu.

OGIP

As described in section 3.2, this option allows you to input data from an OGIP FITS table into PERIOD. The routine displays some information about the file requested and allows you to choose which of its available tables is to be examined. You then select which of the columns in the file refers to the x -axis, y -axis and y -axis errors (if desired, see section 3).

FAKE

Allows you to create fake data with which to test or experiment with PERIOD. Two options are catered for: periodic data or chaotic data. The periodic data are created by summing a user-specified number of sine curves of the form:

$$Y = \text{GAMMA} + (\text{AMPLITUDE} * \text{SIN}((2.0 * \text{PI}) / \text{PERIOD}) * (X - \text{ZEROPT})))$$

The chaotic data are created using a simple logistic equation of the form:

$$X_{n+1} = \text{LAMBDA} * X_n * (1 - X_n)$$

(see, for example, Scargle 1990ab).

NOISE

Using this option, it is possible to add noise to data or randomize data. The latter operation is carried out by specifying the [N]ew dataset option, which will construct an artificial dataset of the same mean value and the same standard deviation as the original. Selecting the [0]ld dataset allows you to apply noise to data, create errorbars on the data points, and/or add noise to the data sampling (so that, for instance, an evenly sampled dataset becomes unevenly sampled). This routine is useful, not only in creating realistic artificial datasets (in conjunction with FAKE), but also in investigating the effects of noise on a period detection.

DETREND

This option removes the D.C. bias from data, which if not removed gives rise to significant power at 0 Hz. There are two options: If the data show no long term trends, it is best to simply subtract the mean and divide by the standard deviation (the [M] option). This gives a dataset with a mean of zero and a standard deviation of one. Otherwise, it is best to subtract a low-order polynomial fit to the data (the [P] option), since if these are not removed, a Fourier transform will inject a significant amount of power at the frequency of the long term variations.

WINDOW

One of the main problems with the *classical* periodogram¹ (see Scargle 1982 for a definition), is *spectral leakage*, of which there are several forms. Leakage to nearby frequencies (sidelobes) is due to the finite total interval over which the data is sampled. Leakage to distant frequencies is due to the finite size of the interval between samples. The WINDOW option sets all the *y*-axis data points to unity. A discrete Fourier transform of the resulting data (using, for example, the FT option, see below) yields the window function (or spectrum), which shows the effects of spectral leakage.

OPEN

It is possible to store the fits calculated by SINE and PEAKS in a log file. This option opens a new log file (if it does not already exist), or else re-opens an old log file and skips over the existing entries.

CLOSE

This option closes the currently open log file.

PERIOD

This is where all the work is done. You will be confronted by the following sub-menu:

```
Options.
-----

SELECT  -- Select data slots.
FREQ    -- Set/show frequency search limits.
CHISQ   -- Chi-squared of sine fit vs frequency.
CLEAN   -- CLEANed power spectrum.
FT      -- Discrete Fourier power spectrum.
PDM     -- Phase dispersion minimization.
SCARGLE -- Lomb-Scargle normalized periodogram.
STRING  -- String-length vs frequency.
PEAKS   -- Calculate period from periodogram.
SIG     -- Enable/disable significance calc.
HELP    -- On-line help.
QUIT    -- Quit PERIOD_PERIOD.

PERIOD_PERIOD>
```

- SELECT – Selects input and output slots for processing, as described in section 3. The input slots should contain the time-series, the output slots will contain, for example, the power spectra. SELECT must be run *every* time a periodicity-finding option is about to be executed; although tedious, this prevents one from accidentally overwriting slots.

¹Throughout the PERIOD package and this document, the terms *power spectrum* and *periodogram* are used interchangeably, although strictly speaking the power spectrum is a theoretical quantity defined as an integral over continuous time, of which the periodogram is merely an estimate based on a finite amount of discrete data (Scargle 1982).

- **FREQ** – Sets the frequency search parameters. The minimum frequency, maximum frequency and frequency interval can be selected by you. Generally, there is no restriction on the number of frequencies to be stepped-through in the processing. Alternatively, by entering 0's, default values can be accepted. Note that the default values are set on entering the **PERIOD** package and thus the **FREQ** option need not be run if default frequencies are required. The default values are calculated as follows: minimum frequency = 0 (ie. infinite period), maximum frequency = $1 / (2 \times \text{Smallest Data Interval})$ (ie. Nyquist), frequency interval = $1 / (4 \times \text{Total Time Interval})$.
- **CHISQ** – This is a straight-forward technique where the input data is folded on a series of trial periods. At each trial period, the data is fitted with a sine curve. The resulting reduced- χ^2 values are plotted as a function of trial frequency and the minima in the plot suggest the most likely periods. See Horne, Wade and Szkody (1986) for an example of the use of this method, which is ideally suited to the study of radial velocity data or any other sinusoidal variations. Note that windowed data cannot be processed by this option since no sine fit is possible.
- **CLEAN** – The **CLEAN** algorithm was originally developed for use in aperture synthesis and was later applied to one-dimensional data by Roberts, Lehár and Dreher (1987). An adapted version of Lehár's code is used here, and is particularly useful for unequally spaced data. The algorithm basically deconvolves the spectral window from the discrete Fourier power spectrum (or dirty spectrum). This produces a **CLEAN** spectrum, which is largely free of the many effects of spectral leakage. In order to prevent small errors from destabilizing the **CLEAN** procedure, the user is prompted for two parameters – the loop gain and the number of iterations. Briefly, with each iteration, some fraction (governed by the loop gain) of the window function is removed from the dirty spectrum. For convergence, the loop gain must lie between 0 and 2, typical values being between 0.1 and 1. Values at the bottom of this range require more iterations, but should provide more stability. Hence, the number of iterations should be large if the loop gain is small, typical values lying between 1 and 100. Note that an increase in the number of cleans produces a less noisy spectrum but, in general, the amplitude of the peaks is decreased, sometimes by a substantial amount. See Roberts, Lehár and Dreher (1987) for further details on choosing these parameters.
- **FT** – This option performs a classical discrete Fourier transform on the data and sums the mean-square-amplitudes of the result to form a power spectrum (see, for example, Deeming 1975). This discrete Fourier transform is defined for arbitrary data spacing and is equal to the convolution of the true Fourier transform with a spectral window. Hence, the effects of data spacing, such as aliasing, are all contained in the spectral window, which can be generated using the **WINDOW** option (see above). This spectral window should be analysed in conjunction with the discrete Fourier transform generated here in order to estimate the effects of aliasing.
- **PDM** – The phase dispersion minimization (PDM) technique is simply an automated version of the classical method of distinguishing between possible periods, in which the period producing the least observational scatter about the mean light curve (or, for example, radial velocity curve) is chosen. This technique (which is described in detail by Stellingwerf 1978) is well suited to cases in which only a few observations are available over a limited period of time, especially if the light curve is highly non-sinusoidal. The data is first folded on a

series of trial frequencies. For each trial frequency, the full phase interval (0,1) is divided into a user-specified number of bins. The width of each bin is specified by the user, such that a point need not be picked (if a bin width narrower than the bin spacing is selected) or a point can belong to more than one bin (if a bin width wider than the bin spacing is selected). The variance of each of these bins (or samples) is then calculated. This gives a measure of the scatter around the mean light curve defined by the means of the data in each sample. The PDM statistic can then be calculated by dividing the overall variance of all the samples by the variance of the original (unbinned) dataset. This process is then repeated for the next trial frequency. Note that windowed data cannot be passed to this option since its variance is zero. If the trial period is not a true period, then the overall sample variance will be approximately equal to the variance of the original dataset (ie. the PDM statistic will be approximately equal to 1). If the trial period is a correct period, the PDM statistic will reach a local minimum compared with neighbouring periods, hopefully near zero.

- **SCARGLE** – By redefining the classical periodogram (ie. the discrete Fourier periodogram) in such a manner as to make it invariant to a shift of the origin of time, Lomb (1976) and Scargle (1982) developed a novel type of periodogram analysis, quite powerful for finding, and testing the significance of, weak periodic signals in otherwise random, *unevenly sampled* data. Horne and Baliunas (1986) have elaborated on the method and Press and Rybicki (1989) present a fast implementation of the algorithm, a modified version of which is used here. This implementation uses FFTs to increase the speed of computation (although it is in no way equivalent to conventional FFT periodogram analysis). Note that windowed data cannot be passed to this option since it needs to calculate the variance (which is zero) to normalize the power of the periodogram.
- **STRING** – The string-length method is an intuitively simple method, described in detail by Dworetsky (1983) and Friend *et al.* (1990). The data is folded on a series of trial periods and at each period the sum of the lengths of line segments joining successive points (the string-length) is calculated. Minima in a plot of string-length versus trial frequency indicate possible periods. The string-length method is especially useful in the limit of a very small number (about 20 or more) of randomly spaced observations of periodic phenomena. Note that windowed data cannot be passed to this option due to the *y*-data scaling process (see Dworetsky 1983).
- **PEAKS** – This option should be run once a periodogram has been obtained. It finds the highest peak in the periodogram (or lowest trough if it is a string-length, PDM or reduced- χ^2 plot) between user-specified frequencies. The resulting period is calculated, along with an error. Errors on period detections are notoriously difficult to estimate. The estimate used in the previous version of PERIOD (v3.0) employed a formula derived by Kovacs (1981). The derivation assumed a single signal, Gaussian noise and even data spacing. This is clearly not the case with most astronomical datasets and the formula is hence of little use (see Horne and Baliunas 1986). Schwarzenberg-Czerny (1991) presents a detailed account of the accuracy of period determinations and advises a *post-mortem* analysis by measuring the width and heights of peaks in a periodogram. Although virtually impossible to automate, it is possible to do this manually from within PERIOD using the fitting routines of QDP/PLT (see above). Therefore, for the sake of generality and to avoid uncertainties, version 4.0 of PERIOD now only outputs an error derived by calculating the half-size of a single frequency bin, centred on the peak (or trough) in a periodogram, and

then converting to period units. This error gives an indication of the accuracy to which a peak can be located in a periodogram (due to the frequency sampling). Clearly, with a larger frequency search interval it is more difficult to locate a peak precisely and this is reflected in the error estimate. However, this error estimate does not take into account the fact that the peak (or trough) may not represent the true period (which can be shifted due to a number of effects) and it should therefore be regarded as a *minimum* error and not a formal error.

If the significance calculation is enabled (with the SIG command, see below), two false alarm probabilities are quoted alongside the period. The first (FAP1) is the probability that, given the frequency search parameters, there is no periodic component present in the data with this period. The second (FAP2) is the probability that the period is not actually equal to the quoted value but is equal to some other value. Note that FAP1 is only output if the whole frequency range is specified to be analysed in PEAKS (see below). One sigma errors on both significance values are also given. If the significance values are zero, these errors are displayed as -1, implying that the false alarm probabilities lie between 0.00 and 0.01 with 95% confidence. Clearly, the lower a significance value and its error, the more likely the quoted period is a correct one. If both the significances and errors are displayed as -1, this means that the input periodogram has not been subjected to a significance calculation (ie. the significance calculation has been disabled). Note that the results can be written to a log file if one is open. For more information on the SIG option, see below. For useful discussions on errors and significances of period determinations, see Schwarzenberg-Czerny (1991) and Nemec and Nemec (1985).

- SIG – This option works as a switch, either turning on or turning off the significance calculation. The default on entering PERIOD is for the significance calculation to be disabled. This means that no significance values are calculated or attached to period determinations. By typing SIG, the significance calculation is enabled. You are first prompted for the number of permutations in the sample. To ensure reliable significance values, the minimum number of permutations is set to 100. You are then prompted for a seed for the random number generator – this number determines the starting point in a number series of infinite period. Therefore, entering the same seed on two calls to SIG will result in the same sequence of random numbers. If SIG is already enabled, one can disable the significance calculation by typing SIG again.

With the significance calculation enabled, every time a period-finding option is run (CHISQ, FT, SCARGLE, CLEAN, STRING, PDM) a Fisher randomization test is performed (see, for example, Nemec and Nemec 1985). This consists of calculating the periodogram as usual and loading the specified output slot. The y -axis data is then shuffled to form a new, randomized time-series. The periodogram of this dataset is then calculated (but not stored in the output slot, which will always contain the periodogram of the real time-series). This randomization and periodogram calculation loop is then performed for the number of permutations specified by the user. This can take a considerable amount of time, depending on the number of data points in the time-series, the frequency search parameters and the number of permutations.

Once the loop is complete, you should enter the PEAKS option to view the resulting significances. Two significance estimates are given in PEAKS. The first, denoted FAP1, represents the proportion of permutations (ie. shuffled time-series) that contained a trough lower than (in the case of the CHISQ, STRING and PDM options) or a peak higher than (in the

case of the FT, SCARGLE and CLEAN options) that of the periodogram of the unrandomized dataset *at any frequency*. This therefore represents the probability that, given the frequency search parameters, no periodic component is present in the data with this period and it is only output in PEAKS if the whole frequency range is specified to be analysed. The second significance, denoted FAP2, represents the proportion of permutations that, *at the frequency given by the period output by PEAKS*, contained troughs lower than (or peaks higher than) the peak or trough in the periodogram of the real dataset. This therefore represents the probability that the period is not actually equal to the quoted value but is equal to some other value, and is quoted for any frequency range specified in PEAKS. Standard errors on both of these false alarm probabilities are also given (see Nemec and Nemec 1985).

It is perhaps worth mentioning here that significance estimates of period detections are notoriously unreliable. The methods used in the previous version of PERIOD (v3.0) suffered from a number of problems. For example, the F-test used with the PDM method (Stellingwerf 1978) has been proved to be incorrect (see, for example, Heck, Manfroid and Mersch 1985). Similarly, the theoretical minimum string-lengths quoted by Dworetsky (1983) are misleading, since they are based on evenly-spaced functions and it is possible to obtain values below this even for pure noise data with certain data spacings. The well-known SCARGLE false alarm probabilities are also incorrect, since the Horne and Baliunas (1986) equation for the number of independent frequencies has been shown to be incorrect (Christian Knigge (Oxford), private communication). Even if correct, the Horne and Baliunas formula would be incorrect to apply in a general way since it is a poor approximation to small datasets. The only reliable method of estimating significances from such non-parametric tests is by some sort of Monte Carlo or randomization method. As described above, one such method (Fisher randomization) has been implemented in this version of PERIOD (v4.0) following the prescription described by Nemec and Nemec (1985).

- **HELP** – This command provides on-line help for PERIOD. Detailed information about individual commands can be obtained by typing `HELP 'COMMAND'` (eg. `HELP PEAKS`).
- **QUIT (or EXIT)** – This quits the PERIOD_PERIOD sub-menu and returns the user to the main PERIOD menu.

Returning to the main PERIOD menu:

FIT

Folds the data on a given period and zero point and then fits the data with a sine curve. The sine curve has the form: $Y = \text{GAMMA} + (\text{AMPLITUDE} * \text{SIN}((2.0 * \text{PI}) / \text{PERIOD}) * (X - \text{ZEROPT}))$. Outputs the fit parameters (which can be written to a log file) and the resulting sine curve.

FOLD

Folds the data on a given period and zero point. Hence, this option transforms the data onto a phase scale, where one phase unit is equal to one period and phase zero is defined by the zero point. If the zero point is not known, the data can be folded by taking the first data point as the zero point. This option is useful for checking whether derived periods actually give sensible results when applied to the data. In addition to normal folding, it is also possible to phase bin the data, which folds the data and then averages all the data points falling into each bin.

SINE

Adds, subtracts, multiplies or divides a sine curve from data. The sine curve has the form: $Y = \text{GAMMA} + (\text{AMPLITUDE} * \text{SIN}((2.0 * \text{PI}) / \text{PERIOD}) * (X - \text{ZEROPT}))$. This option is useful for removing or adding known periods from/to data, thus enabling or testing the detection of other periods.

PLT

This routine calls PGPLOT routines to display the graphs of the slots requested. The layout of the displays is fixed but output file types such as landscape postscript files can be created. This represents slightly less functionality than the original XANADU based QDP PLT routine, but no QDP PLT routine is currently available for LINUX.

In order to receive on-line help, simply type HELP at the PERIOD-PLT prompt. To exit PERIOD-PLT and return to the PERIOD menu, type EXIT.

STATUS

Returns information on the data slots or on the stored fits in the log file. This command is useful in order to check which slots contain which datasets and also as a means of obtaining some elementary statistics on the stored data. You can also use this option to check the fits from the SINE and PEAKS options stored in the log file without having to exit the package and read the log file.

OUTPUT

Writes any selected slot to an ASCII file on disk. This is the only way of saving data created by PERIOD (it does not write to FITS files), and should therefore be run before QUITting in order to store, say, a power spectrum.

HELP

This command provides on-line help for PERIOD. Detailed information about individual commands can be obtained by typing `HELP 'COMMAND'` (eg. `HELP PERIOD`).

QUIT (or EXIT)

This option quits a PERIOD session. However, it does provide a last chance to stay in the package. This is essential to prevent accidental exit, since any data files created using PERIOD will be lost on exit from the package *unless* one `OUTPUTS` the data first.

6 A Simple Recipe

A simple guide designed to introduce inexperienced users to the steps involved in detecting periodicities is outlined below. Detailed descriptions of the individual PERIOD commands can be found in section 5 and in the on-line help facility (which also gives help on the individual prompts one is confronted with).

- (1) Create an ASCII data file containing the time-series.
- (2) Read ASCII data into PERIOD using `INPUT`.
- (3) or read OGIP FITS files into PERIOD using `OGIP`.
- (4) Detrend the data using the `[M]` option in `DETREND` (if the data show long term variations, use the `[P]` option instead).
- (5) Open a log file for the fits using `OPEN`.
- (6) Enter the `PERIOD_PERIOD` menu by typing `PERIOD`.
- (7) Select the slots which contain the time-series data and specify the output slots for the periodograms using `SELECT`.
- (8) Set the frequency search limits using `FREQ`. If you have no idea what the period is, accept the default values by typing 0 (or alternatively, by not typing `FREQ` in the first place).
- (9) Enable the significance calculation by typing `SIG` and specifying, say, 200 permutations.
- (10) Calculate the Lomb-Scargle periodogram by typing `SCARGLE`.
- (11) Now run `PEAKS` on the resulting periodogram, specifying the frequency range which contains the peak you wish to measure (you may enter 0,0 if you wish to process the entire range). Write the results to the log file.
- (12) Now reselect the time-series slots and different output slots for a new periodogram using `SELECT`.

- (13) Type CLEAN with 5 iterations and a loop gain of 0.2, for example. (Before doing this, you may wish to disable the significance calculation by typing SIG again, since the CLEAN algorithm can take a considerable amount of processing time).
- (14) Run PEAKS on the resulting periodogram and store the results.
- (15) Now quit the PERIOD_PERIOD sub-menu by typing QUIT.
- (16) Plot the periodograms using PLT. Check to see the validity of the highest peak selected.
- (17) Check the results in the log file using STATUS. In particular, look closely at the false alarm probabilities.
- (18) Fold the *original* data on the most likely period using FOLD.
- (19) Plot the folded data using PLT. If this looks sensible, the period may well be correct. Make a postscript file by typing epsf_1. To see the other options type '?' instead.
- (20) Output the periodograms and folded data to an ASCII file on disk using OUTPUT.
- (21) Exit PERIOD by typing QUIT.

The above description is intended only to be a very brief guide. Clearly, a great deal more experimentation is required before it can definitely be said that a period has been detected. For example, you should investigate other large peaks in the periodogram, try smaller or larger frequency ranges, or try one of the other periodicity-finding options (a useful comparison of a number of different techniques is given by Carbonell, Oliver and Ballester 1992 and Heck, Manfroid and Mersch 1985). Other analysis techniques might also be attempted, such as subtracting a sine curve from the data in order to investigate its effects on the harmonics and enable the detection of less-dominant periods.

7 Future Improvements

Include more periodicity-finding options, such as AoV (Schwarzenberg-Czerny 1989), maximum entropy method (Fahlman and Ulrych 1982), an FFT, and chaos analysis (Canizzo and Goodings 1988, Scargle 1990ab and Lehto, Czerny and McHardy 1993).

8 Acknowledgments

The author gratefully acknowledges the use of subroutines written by Joseph Lehár (Cambridge), Keith Horne (STScI) and Tom Marsh (Oxford), as well as those written by Press and Rybicki (1989). I also thank the bug-finding guinea-pigs Martin Still (Sussex), Mark O'Dell (Sussex), Erik Kuulkers (Amsterdam) and Thomas Augusteijn (Amsterdam) for their many useful comments and suggestions. Finally, I am indebted to Christian Knigge (Oxford) for his help with bug fixes and his many suggestions regarding significance estimates.

9 References

- Canizzo, J. & Goodings, D., 1986. *Astrophys. J. Lett.*, **334**, L31.
- Carbonell, M., Oliver, R. & Ballester, J. L., 1992. *Astron. Astrophys.*, **264**, 350.
- Deeming, T. J., 1975. *Astrophys. Space Sci.*, **36**, 137.
- Dworetsky, M. M., 1983. *Mon. Not. R. astr. Soc.*, **203**, 917.
- Fahlman, G. G. & Ulrych, T. J., 1982. *Mon. Not. R. astr. Soc.*, **199**, 53.
- Friend, M. T., Martin, J. S., Smith, R. C. & Jones, D. H. P., 1990. *Mon. Not. R. astr. Soc.*, **246**, 637.
- Heck, A., Manfroid, J. & Mersch, G., 1985. *Astron. Astrophys. Suppl. Ser.*, **59**, 63.
- Horne, J. H. & Baliunas, S. L., 1986. *Astrophys. J.*, **302**, 757.
- Horne, K., Wade, R. A. & Szkody, P., 1986. *Mon. Not. R. astr. Soc.*, **219**, 791.
- Kovacs, G., 1981. *Astrophys. Space Sci.*, **78**, 175.
- Lehto, H. J., Czerny, B. & McHardy, I. M., 1993. *Mon. Not. R. astr. Soc.*, **261**, 125.
- Lomb, N. R., 1976. *Astrophys. Space Sci.*, **39**, 447.
- Nemec, A. F. L. & Nemec, J. M., 1985. *Astron. J.*, **90**, 2317.
- Press, W. H. & Rybicki, G. B., 1989. *Astrophys. J.*, **338**, 277.
- Roberts, D. H., Lehár, J. & Dreher, J. W., 1987. *Astron. J.*, **93**, 968.
- Scargle, J. D., 1982. *Astrophys. J.*, **263**, 835.
- Scargle, J. D., 1990a. *Astrophys. J.*, **359**, 469.
- Scargle, J. D., 1990b. In *Errors, Bias & Uncertainties in Astronomy*, Eds. Jaschek, C. & Murtagh, F., Cambridge University Press, Cambridge.
- Schwarzenberg-Czerny, A., 1989. *Mon. Not. R. astr. Soc.*, **241**, 153.
- Schwarzenberg-Czerny, A., 1991. *Mon. Not. R. astr. Soc.*, **253**, 198.
- Stellingwerf, R. F., 1978. *Astrophys. J.*, **224**, 953.

A History

This appendix lists all of the changes made to the PERIOD package and the release dates of the various versions.

DATE	Author	Comments
1992 July 10	VSD	VERSION 1.0 released to Robert Smith (Sussex), Mark O'Dell (Sussex) and Erik Kuulkers (Amsterdam). *** VERSION 1.0 ***
1992 July 21	VSD	This history file added to the package.
1992 July 21	VSD	PERIOD_SCARGLE modified to output EFFM instead of PROB and JMAX. This enables PEAKS to calculate the false alarm probability of any peak in the periodogram.
1992 July 21	VSD	PERIOD_PERIOD modified to cope with changes in PERIOD_SCARGLE and calculate the false alarm probability of any peak selected in PEAKS. Also included a trap for invalid values of the loop gain in CLEAN (greater than 0 and less than 2).
1992 July 21	VSD	PERIOD_INPUT modified to prevent crashes if the input file contains two contiguous x-axis points with the same or decreasing values.
1992 July 21	VSD	PERIOD_DOC extended by adding a recipe.

DATE	Author	Comments
1992 July 21	VSD	PERIOD_MAIN, _CLEAN, _DETREND, _FIT, _FOLD, _FT, _INPUT, _NOISE, _PERIOD, _PHASE, _POLYFIT and _STATUS modified in order to handle 10000 data or frequency points and 40 data files containing 20 columns. This ensures a page file quota of no more than 34000 is required to run PERIOD.
1992 July 21	VSD	PERIOD_LOG, _READFREE, _STATUS modified by setting large loop variables in data or parameter declarations.
1992 July 21	VSD	Included a MAXCOL parameter in PERIOD_MAIN and PERIOD_INPUT in order to cut down array sizes. Also set size of MAXANT to MXROW in order to ensure that only the maximum number of permitted rows are loaded.
1992 July 21	VSD	PERIOD_COM:PERIOD_MAIL.COM added in order to make e-mailing new versions of PERIOD easier.
1992 July 21	VSD	VERSION 1.1 of PERIOD released to incorporate the above changes. Copy given to Mark O'Dell (Sussex).
		*** VERSION 1.1 ***
1992 July 29	VSD	PERIOD_SCARGLE modified to remove calculation of false alarm probability from subroutine.
1992 July 29	VSD	PEAKS in PERIOD_PERIOD modified to include a more rigorous calculation of the SCARGLE false alarm probability using an analytical expression for the number of independent frequencies.
1992 July 30	VSD	NWK parameter, specifying sizes of the workspace arrays in PERIOD_PERIOD and PERIOD_SCARGLE, reduced from MXROW*100 to MXROW*16 in order to cut down on memory usage.
1992 July 30	VSD	PERIOD_MOMENT modified to remove redundant skew and kurtosis calculations.
1992 Aug 6	VSD	PERIOD_DOC.TEX split into two documents -- a user document (PERIOD_USER.TEX -- LUN 2.1) and a system document (PERIOD_SYSTEM.TEX -- LSN 1.1).
1992 Aug 7	VSD	PERIOD_INTRO modified to give my address as RGO La Palma and my e-mail account as CAVAD::VSD.
1992 Aug 17	VSD	PERIOD_USER.TEX largely rewritten in order to make document clearer and more comprehensive.
1992 Aug 17	VSD	Sampling interval in MEM option of PERIOD_PERIOD now output in PEAKS instead of in MEM.

DATE	Author	Comments
1992 Aug 17	VSD	Dimension of NWK in PERIOD_PERIOD changed from 16 to 27. This is because the maximum possible workspace usage (assuming 10000 data points and 10000 frequency steps) in PERIOD_SCARGLE is 262144.
1992 Aug 17	VSD	VERSION 1.2 of PERIOD released to incorporate the above changes. Installed on SUSTAR as local STARLINK software. Copy sent to Erik Kuulkers (Amsterdam).
*** VERSION 1.2 ***		
1992 Sept 24	VSD	Annoying information written to screen when running PLT removed from PERIOD_PLT.
1993 Mar 3	VSD	Now at ING, not Sussex! PERIOD_PDM and PERIOD_FTEST subroutines added for the new Phase Dispersion Minimization (PDM) option.
1993 Mar 3	VSD	PERIOD_PERIOD extensively modified to cope with the new PDM option. The two major changes are the inclusion of the code to call PERIOD_PDM (which calculates the PDM statistic) and PEAKS, which now does an F-test to determine the significance of a PDM minimum by calling PERIOD_FTEST.
1993 Mar 3	VSD	Error check on zero period added to PERIOD_FAKE, PERIOD_FIT, PERIOD_PHASE and PERIOD_SINE.
1993 Mar 4	VSD	Reduced the size of the INFILEARRAY string in PERIOD_PERIOD by shortening 'Frequency' to 'Freq'.
1993 Mar 4	VSD	Adapted PERIOD_SYSTEM.TEX (LSN 1.2) and PERIOD_USER.TEX (LUN 2.2) to incorporate new PDM option and other changes to bring the documentation into line with the STARLINK release.
1993 Mar 4	VSD	New e-mail address, version number and date added to PERIOD_INTRO.
1993 Mar 4	VSD	PERIOD_COMPILE, PERIOD_LIBRARY, PERIOD_MAIL all modified to include the new PERIOD_PDM and PERIOD_FTEST subroutines.
1993 Mar 4	VSD	VERSION 2.0 of PERIOD released to incorporate the above changes. Installed on ING Vax and copies sent to STARLINK (RLVAD::STAR), Sussex (SUSTAR::SYSTEM) and Amsterdam (PSI\%IBERPAC.DWINGELOO::UVAA01::ERIK)
*** VERSION 2.0 ***		

DATE	Author	Comments
1993 Mar 16	VSD	Bug in FOLD option found by Martin Still (Sussex). Traced to PROMPT_ZEROPT parameter not setting ZEROPT parameter correctly. Modified PERIOD_FIT and PERIOD_PHASE subroutines accordingly (by removing the PROMPT_ZEROPT parameter completely).
1993 Mar 17	VSD	Included an on-line help option. A new subroutine, PERIOD_HELP, reads a set of help files contained in the new PERIOD subdirectory PERIOD_HLP. The help files are named according to the convention: PERIOD_'COMMAND'.HLP. PERIOD_MAIN and PERIOD_PERIOD have been modified accordingly, as have the various command procedures and documents (LSN 1.3 and LUN 2.3).
1993 Mar 17	VSD	Minor changes made to PERIOD_INPUT to reduce references made to RUBY.
1993 Mar 17	VSD	TYPE replaced by WRITE in PERIOD_INPUT and PERIOD_STATUS to conform to standard F77.
1993 Mar 17	VSD	Bug in PERIOD_LOG which prevented the opening of an old log file fixed.
1993 Mar 17	VSD	VERSION 3.0 of PERIOD released to incorporate the above changes. Installed on ING Vax and copies sent to STARLINK (RLVAD::STAR), Sussex (SUSTAR::SYSTEM) and Amsterdam (PSI\%IBERPAC.DWINGELOO::UVAA01::ERIK) This was the first version to be released on STARLINK.
*** VERSION 3.0 ***		
1993 Jul 5	VSD	Included line number as diagnostic of corrupted x-axis data in PERIOD_INPUT (in response to request from Margaret Penston, RGO).
1993 Aug 21	VSD	Bug in PERIOD_PHASE found by Martin Still (Sussex). When folding data using binning on a number of files the folded data is all placed into a single (incorrect) slot. Problem traced to the use of the COUNTER variable for two different tasks (slot number and data array elements). Changed the data array variable to BINCOUNT. Checked all PERIOD subroutines for a similar error and nothing found.
1993 Aug 2	VSD	Bug in PERIOD_CLEAN found (and corrected) by Christian Knigge (Oxford). When processing a number of files using CLEAN the first periodogram is calculated correctly, but subsequent files are

DATE	Author	Comments
		wrongly processed, as are files processed if CLEAN is run twice without exiting PERIOD. Problem traced to the C array which is now initialised on each call to PERIOD_CLEAN. Checked all PERIOD_PERIOD subroutines for a similar error and nothing found.
1993 Aug 22	VSD	PERIOD_CLEAN made more robust by checking the HWHM calculated by PERIOD_HWHM is not zero before calling PERIOD_FILLB -- this used to cause a crash.
1993 Aug 22	VSD	PEAKS in PERIOD_PERIOD now checks that different numbers are specified for the periodogram and time-series. Also now quits if a 0 is entered for the slot number (this did not work properly before).
1993 Aug 22	VSD	Bug in CHAOS option of PERIOD_FAKE did not reset the initial value parameter if processing more than one slot. Problem remedied by introducing a new parameter -- INITVAL.
1993 Aug 22	VSD	PERIOD_DETREND made more robust by removing the PERIOD_POLYFIT subroutine and calling PERIOD_LSQUAR directly from PERIOD_DETREND. Program now outputs the chi-squared value of the polynomial fit. Also rewrote PERIOD_POLY to handle polynomials of order up to 20 (the previous limit was 12) and removed the redundant subroutine PERIOD_DPOLY. Updated the command procedures in PERIOD_COM by removing any reference to PERIOD_POLYFIT, which has now been deleted from the PERIOD_FOR subdirectory.
1993 Aug 22	VSD	Bug found in DETREND -- the polynomial fit was divided and not subtracted from the time-series. Have modified PERIOD_DETREND accordingly.
1993 Aug 22	VSD	Added my INTERNET e-mail address to PERIOD_INTRO and incremented the date and version number (3.1).
1993 Aug 27	VSD	Removed string length significance limits from PEAKS in PERIOD_PERIOD since misleading (the theoretical minimum only applied to an evenly-spaced sinusoid). Followed comment by Christian Knigge (Oxford).
1993 Aug 27	VSD	Corrected reduced chi-squared values in the CHISQ option of PERIOD_PERIOD and PERIOD_FIT by dividing by the number of points minus the number of free parameters (=3 in PERIOD_SINFIT). Previously, the chi-squared value was only divided by the number of points. Bug spotted by Christian Knigge (Oxford).

DATE	Author	Comments
1993 Aug 27	VSD	Removed chi-squared value of polynomial fit from PERIOD_DETRENDED and replaced with simple RMS calculation. VERSION 3.1 never released.
		*** VERSION 3.1 ***
1993 Nov 13	VSD	Removed CHAOS option from PERIOD_PERIOD since it is of no use in its present form. Modified PERIOD_HELP and various .COM files and deleted the .HLP file to reflect this change.
1993 Nov 13	VSD	Removed MEM option from PERIOD_PERIOD since it is of no use in its present form. Modified PERIOD_HELP and various .COM files and deleted the .HLP file to reflect this change. Removed the now redundant subroutines PERIOD_EVLMEM, PERIOD_MEMCOF and PERIOD_MDIAN1 from the PERIOD_FOR directory and placed them in the PERIOD_DEV directory.
1993 Nov 13	VSD	Minor bug in PERIOD_NOISE discovered. Removed the option which allows one to add Poissonian noise to the data since it incorrectly just added the SQRT of the data value.
1993 Nov 13	VSD	Removed PERIOD_FTEST from the PERIOD_FOR directory and placed in PERIOD_DEV directory. This routine is now redundant since significances are calculated in PERIOD_PERIOD using a Fisher test (see below).
1993 Nov 14	VSD	Major changes to PERIOD_PERIOD in order to include significance estimate for each periodogram. Removed the estimates from PEAKS (which were all incorrect), added an extra option SIG and modified each of the 6 remaining PERIOD finding options in order to perform a Fisher randomisation test. Also included new INFO parameter in PERIOD_FT, PERIOD_SCARGLE and PERIOD_CLEAN.
1993 Nov 14	VSD	Modified PERIOD_INTRO to reflect new version, date and removed request for acknowledgments.
1993 Nov 14	VSD	Modified PERIOD_DETRENDED and PERIOD_POLY by setting maximum polynomial order from 20 to 10 (since this was causing underflow errors).
1993 Nov 14	VSD	Changed variable type of SEED in PERIOD_NOISE from REAL to INTEGER, since this caused a crash.
1993 Nov 14	VSD	Changed check of minimum number of points in PERIOD_SINFIT from .LT. 3 to .LE. 3 since this caused a crash.
1993 Nov 18	VSD	Updated all the .HLP files to reflect the above changes and included a new file for the SIG option.

DATE	Author	Comments
1993 Nov 18	VSD	Restructured PERIOD subdirectories. Now only 4 subdirectories -- .FOR, .HLP, .DOC and .SYS. The first three directories contain the fortran source code, help files and user documents (as before). The .SYS directory contains the executable image and the command procedures for initialising and making the PERIOD package.
1993 Nov 18	VSD	Restructured the command procedures. There are now only two: PERIOD_START.COM and PERIOD_MAKE.COM (which calls PERIOD_MAKE.OPT). This is to bring it more into line with STARLINK standard. PERIOD_START.COM now defines the new directories and defines a new logical PERIOD_DISK. PERIOD_MAKE.COM compiles, creates and fills the object library, links and then deletes the object files and object library.
1993 Nov 18	VSD	To bring into line with STARLINK, the PERIOD initialisation command is now PERIODSTART (and not PERIOD_START).
1994 Jan 15	VSD	Included a new subroutine PERIOD_CASE which converts strings from lower to upper case. Replaces calls to the VAX/VMS specific STR\UPCASE.
1994 Jan 24	VSD	Modified PERIOD_STRING and PERIOD_SINFIT so as not to crash when presented with windowed data. All PERIOD_PERIOD options can now either cope with windowed data or else report an error.
1994 Jan 25	VSD	Changes made to PERIOD_DOC files. This file converted to \LaTeX\ and renamed PERIOD_HISTORY.TEX. PERIOD_SYSTEM.TEX renamed to SSN167.TEX and rewritten to reflect above changes. SUN167.TEX also rewritten to reflect above changes. PERIOD_USER.TEX (the old Sussex LUN) deleted.
1994 Jan 26	VSD	Changes made to PERIOD_INPUT so that if it finds x-axis data which is not in ascending order, it prompts users for whether they would like to sort it or not. If they do, it sorts the data using PERIOD_SHELLSORT. If not, it aborts as before. Follows suggestions by Ed Zuiderwijk (RG0).
1994 Feb 1	VSD	Following extensive testing, VERSION 4.0 of PERIOD released to incorporate the above changes. Installed on ING Vax and copies sent to STARLINK (RLVAD::STAR) and Amsterdam (ERIK@astro.uva.nl).

*** VERSION 4.0 ***

DATE	Author	Comments
1995 June 15	GJP	PERIOD Version 4.0 ported to UNIX. Functionality remains the same as version 4.0 Now requires an environmental variable to find the help files. SUN167 and SSN25 re-written dropping references to VMS.
1995 Sept 26	GJP	Added the OGIP reading option and modified the documents accordingly. SSN167 becomes SSN25.
1995 Oct 12	GJP	Modified SUN167, SSN25 and PERIOD_HISTORY to be compatible with star2html so that WWW versions can be created by latex2html. Also, removed conversion of input files names to upper case. Submitted to STARLINK.
1995 Dec 15	BLY	Removed dependencies on XANADU versions on PLT and FITSIO by a) providing xanlib.a and plt.dhf for supported systems, b) using Starlink version of FITSIO. Makefile and documents modified accordingly. Hypertext versions of documents created. Starlink Release.
*** VERSION 4.1 ***		
1997 Mar 10	GJP	PERIOD 4.1 ported to LINUX. QDP PLT routine replaced by PGPLOT with some small loss of interactive plotting capability. Slightly clearer error messages in some routines, modified help files and a minor VAXism in period_output was fixed. A number of routines have been changed to ensure that the variables used are preinitialised to zero.
1997 May 07	BLY	Modified period_poly.f to use integer powers not double precision powers when raising X to higher powers since X can be negative. Old method caused core dump on Digital Unix.
*** VERSION 4.2 ***		
*** VERSION 5.0 ***		
2001 Dec 12	KPD	PERIOD implemented in Double Precision. Dynamic array allocation/deallocation introduced for the majority of the input, output and work arrays.