Guide to KISIP v6

Introduction

KISIP is the Kiepenheuer-Institute Speckle Interferometry Program.

The completely rewritten v6 introduces:

- ANSI C Code – runs on most POSIX systems
- parallelization
- highly improved Fried parameter estimation based on a weighted least squares fit
- STF's for AO corrected data
- recursive & iterative weighted least squares speckle masking phase reconstruction

System Requirements

In order for KISIP v6 to run, the following C libraries need to be installed on the system:

- MPI – Message Passing Interface
  Because of the MPI standard, one can choose from different libraries. This version has been tested LAM-MPI >7.0.6 (recommended) available at http://www.lam-mpi.org
- GSL – GNU Scientific Library

  available at http://www.gnu.org/software/gsl/
- FFTW – fastest fourier transform of the west

  available at http://www.fftw.org
- GraphApp – to get the graphical interface

  available at http://enchantia.com/software/graphapp/

These libraries run on most UNIX systems.

Starting the program

Since LAM-MPI is recommended, here is a short overview of what needs to be done to start the program using the LAM-MPI environment.

1. Modify the .cshrc/.bashrc to include the correct paths
   - no output may be written to stderr by the .cshrc/.bashrc
   - if the libraries are installed in e.g. ${HOME}/lib/{mpi,gsl,fftw,gapp}, then make sure that

     ```
     setenv LD_LIBRARY_PATH ${HOME}/lib/mpi/lib:${HOME}/lib/gsl/lib:${HOME}/lib/fftw/lib:${HOME}/lib/gapp/src
     set path = ( ${HOME}/lib/mpi/bin )
     ```

     in the .cshrc. Change your .bashrc accordingly if applicable.
   - my experience is that it's best to insert these lines at the top of the 'rc' file
2. Modify the bhost file to your needs
   Within the bhost file the nodes (computers) to be used are defined. It is a list of either the name or the IP of the nodes followed by a statement of how many CPUs are present on that machine.

   Example:

node1 cpu=1
198.68.0.1 cpu=2

This will boot the LAM environment with two nodes, node1 utilizing 1 CPU, 198.68.0.1 using 2 CPUs. **Please note:** the computer that KISIP v6 is executed from needs to be present in this list, preferentially as the first node in the list, in this case node1, which will be the 'master'. The rest of the nodes are the 'slaves'. This means that you should issue an "ssh -X node1" in this example and continue with 3.

3. Boot the LAM environment
   The LAM environment using the command:
   lamboot -v bhost
   This gives some output on what node is currently booted and whether the command was successful or not and some hints on why it was/wasn't successful.
4. Execute KISIP v6
   KISIP itself is executed in the following way:
   mpirun C entry
5. Shutting down the LAM environment
   This is done only, if KISIP is definitely not used anymore. Even if the program is stopped using Ctrl-C you do not need to shut down the LAM environment. Once you are finished, just execute:
   lamhalt

Interface

Introduction

When started, the interface of KISIP v6 shows 2 tabs, one called 'Extended Knox-Thompson' (greenish) and the other 'Triple Correlation' (blueish). Whatever tab is selected when the 'Start' button on the bottom is hit, that method for phase averaging/reconstruction is used for image reconstruction.

Menu bar

File

In the file menu window, the location of the files to be reconstructed as well as some other file properties are set.
The input files (speckle bursts) are expected to be

- calibrated in the sense of dark- and flatfield correction
- in the format NAME.SUF.
  While NAME, the base filename, can be chosen freely, SUF has to be a 3 digit number.

Selected filename (...) for opening:
Please enter the whole path as well as the base filename NAME. It should look like:
/path/to/file/NAME
(Beware: if you select the file with the 'open' button, you need to remove the suffix after selection)

The format of these files (speckle bursts) is required to be platform native binary float, without any separator between the single images!

Start and end suffix (...):
This is needed for the batch processing of a file series.
Please enter the start and end SUF numbers of the files that are to be reconstructed. It doesn't matter, if

files with numbers within the range are missing.

Selected filename (...) for saving:
Please enter the path and base filename for the result. Several files will be created, their names will be unique with the SUF number of the corresponding input file. It should look like:
/path/to/output/OUTNAME

Noise File:
Path and filename to a burst (with the same number of images and header like that of a speckle burst) of flatfielded flatfield images.

The format of this file is **required** to be **platform native binary float**.

Data

In the data menu, the basic properties of the data files are set.

xsize, ysize:
Set the image size in pixels here

number of frames:
Set the number of frames in one burst

Remarks:

- remember: speckle burst file format
  see explanation: Menu - Selected filename (...) for opening
- remember: noise file
  this number is also used for the noise file
  see explanation: Menu - Noise File

header:
Set the header offset in bytes of one burst. This could be used if the burst is in FITS format (which normally has a header offset of 2880 bytes). I have never tried it...

Remarks:

- remember: noise file
  this number is also used for the noise file
  see explanation: Menu - Noise File

arcsec per px in x/y:

Enter the pixelscale. Try to get this as accurate as possible

Warning:

- I have NOT really tested what happens, if the scale in x and y doesn't match. Probably the program crashes (really ... I have a lot of code cleanup to do), but different scales shouldn't happen too often, right?

Telescope diameter:
Enter the telescope diameter in millimeter

Wavelength:
Enter the observed wavelength in nanometer

AO used?:
Check 'yes' here, if you have used an AO system to acquire the data, otherwise check 'no'. If the STFs have been modeled specifically for your AO system, then check the checkbox.

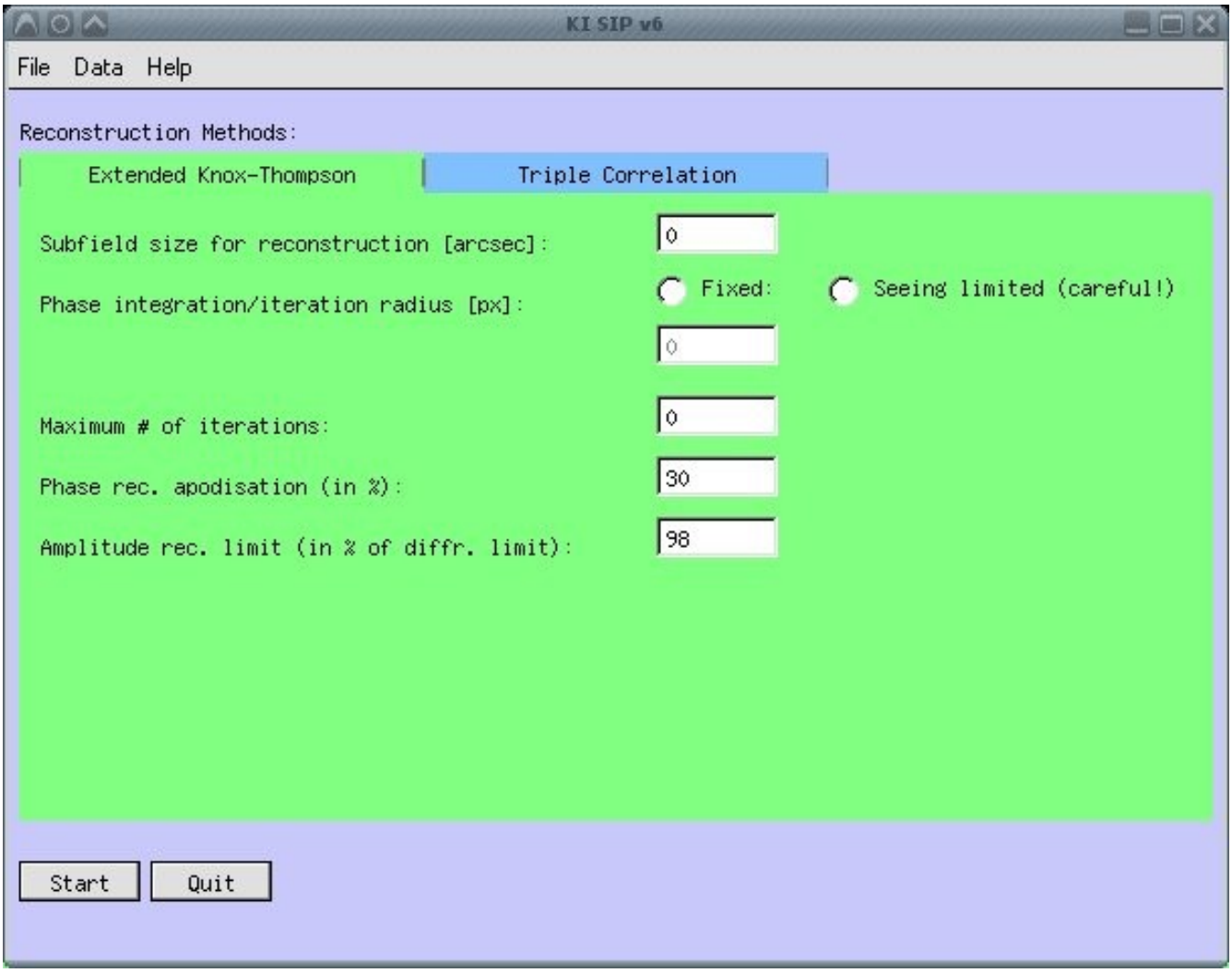
AO lock structure location:

If you choose 'auto', then KISIP will try to detect the lock structure by itself which works in most cases, but unfortunately not all. This depends on the quality of the data. In case of a failure, check 'manual' and enter the position of the lock structure in pixel.

Reconstruction method tabs

Within the above mentioned tabs the main settings of the two implemented methods are set. To increase the usability, I tried to make the fields look as similar as possible.

Extended Knox-Thompson



Subfield size:
This is the parameter with which you control your subfield size. The size should normally not be larger than the size of the isoplanatic patch. Especially in bad seeing though, breaking this rule and increasing the subfield size might help get a more appealing reconstruction, in particular using the extended Knox-Thompson technique.
This parameter is entered in arcseconds. The number will be converted into pixels and then scaled to the nearest power of two to speed up the reconstruction process.
Remark: The bigger the subfields, the longer the computational time!

Phase integration/iteration radius:
Both fixed and seeing limited are options. If seeing limited is chosen, the maximum radius allowed by the seeing is chosen. Be careful: this can be well more than 100 shifts for the cross-spectrum! Not only does that use a lot of memory, it also increases the computational time (most of the times unnecessarily). YOU HAVE BEEN WARNED! In most cases, a fixed shift radius of 6 pixels

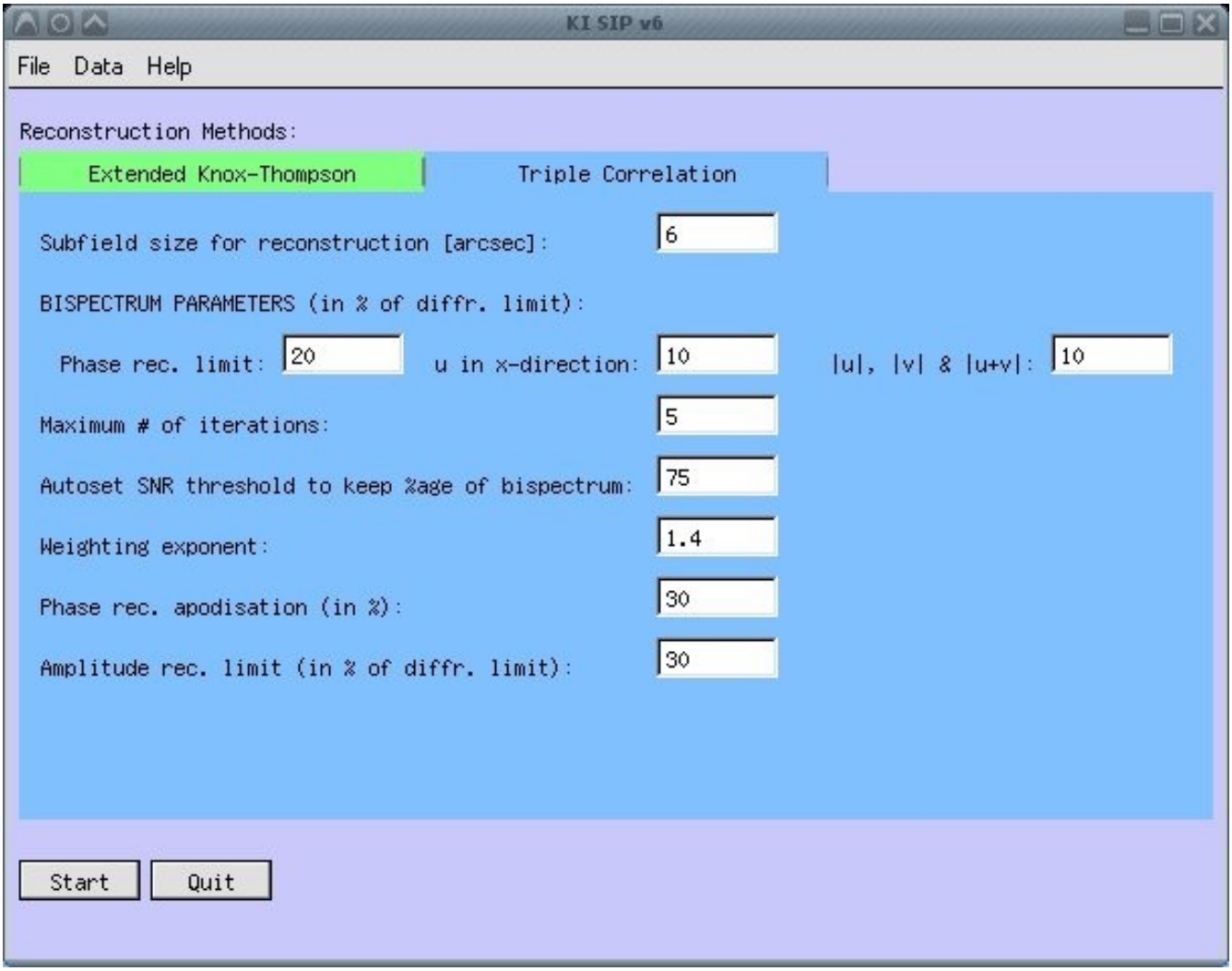
(corresponding to ~ 20-30 shifts for the cross-spectrum) is good enough.

Maximum number of iterations:
Enter here the maximum number of iterations you want to perform for the Successive-Over Relaxation algorithm.

Phase reconstruction apodisation:
This is a parameter that allows the setting of the percentage of the apodisation used for the windowing function used to reduce the artifacts introduced by the fourier transform. Entering 30% will effectively apodise 15% on top, bottom, left and right edge of the image.
This can affect the result! Windowing in the spatial domain means convolution in the fourier domain, so phases get mixed up – usually, this shouldn't be a problem, but for some reason it is. Keep it at 30%; for sunspots, slightly smaller values (25% or so) are better, to avoid artefacts in the umbra.

Amplitude rec. limit:
Although there is a sophisticated noise filtering function present in the program, this value might give you some power over where to cut (its a plain lowpass). Especially useful, when there are some camera readout noise peaks present in the data. If you dare, go to 100% ;)

Triple correlation



Subfield size:
This is the parameter with which you control your subfield size. The size should normally not be larger than the size of the isoplanatic patch. Especially in bad seeing though, breaking this rule and increasing the subfield size might help get a more appealing reconstruction.
This parameter is entered in arcseconds. The number will be converted into pixels and then scaled to the nearest power of two to speed up the reconstruction process.
Remark: The bigger the subfields, the longer the computational time! Especially for the Speckle Masking method. This has to do with the restriction for the bispectrum (see below).

BISPECTRUM PARAMETERS:
This is the essential part where you don't want to go wrong, as this can increase the computational indeed. Three parameters can be changed.

1. Phase reconstruction limit: In this example it is set to 20% of the diffraction limit, which is a very small value. At least 75% or higher is recommended as this is the only way to get near diffraction limited images. Reducing this of course decreases the computational time greatly.
2. u in x-direction: limit the x component of the first bispectrum vector to a certain percentage of the diffraction limit. This parameter is important.

3. |u|, |v| & |u+v|: limit the lengths of the first and second bispectrum vector as well as their sum to a certain percentage of the diffraction limit.
   Get the best result, when the values in 2. and 3. are equal.

Maximum number of iterations:
Enter here the maximum number of iterations you want to perform for the Iterative Weighted Least Squares Algorithm.

Autoset SNR threshold to keep %age of bispectrum:
Set here the percentage of the bispectrum values to be used for the phase reconstruction. The input value sets a threshold such that only the values with the best signal to noise ratio are kept.

Weighting exponent:
Set here the exponent for the SNR , which is used as weight in the algorithm. This is used to converge faster. A value of 1.3 is usually fine.

Phase reconstruction apodisation:
This is a parameter that allows the setting of the percentage of the apodisation used for the windowing function used to reduce the artifacts introduced by the fourier transform. Entering 30% will effectively apodise 15% on top, bottom, left and right edge of the image.
This can affect the result! Windowing in the spatial domain means convolution in the fourier domain, so phases get mixed up – usually, this shouldn't be a problem, but for some reason it is. Keep it at 30%; for sunspots, slightly smaller values (25% or so) are better, to avoid artifacts in the umbra.

Amplitude rec. limit:
Although there is a sophisticated noise filtering function present in the program, this value might give you some power over where to cut (its a plain lowpass). Especially useful, when there are some camera readout noise peaks present in the data. If you dare, go to 100% ;)

If you have any questions (and I am sure you will have a couple), don't hesitate to contact me at:

woeger@kis.uni-freiburg.de