The user manual for SPARX
(Simulation Package for Astrophysical Radiative Transfer)
The version alpha 2
2014.10.03

Introduction

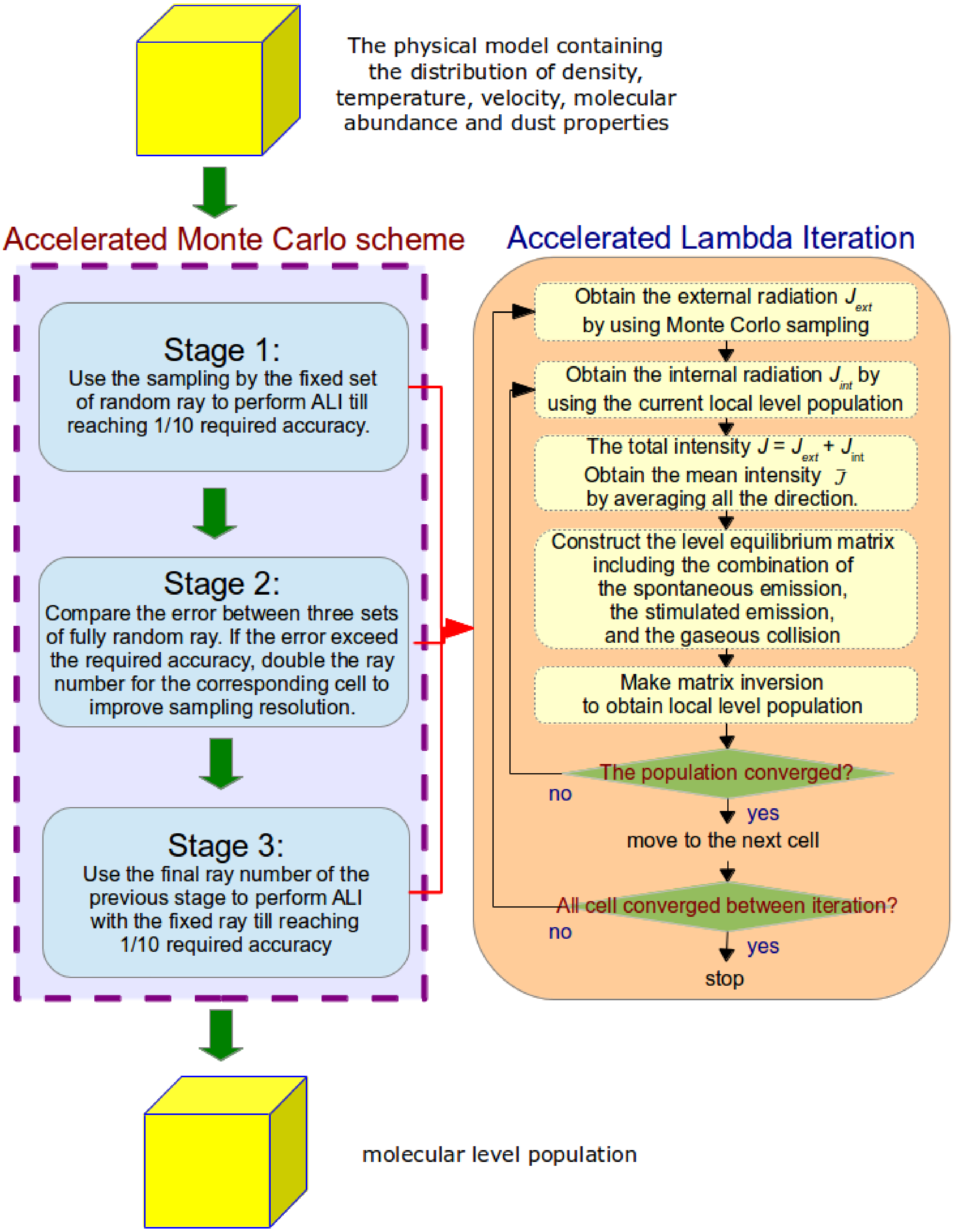
SPARX is a 3-dimensional parallelized radiative transfer simulator. The core of the package is used for solving non-LTE molecular level excitation by adopting the accelerated Monte Carlo method and accelerated Lambda iteration. The image producer as the post processing can generate the molecular line spectrum, image of dust continuum and polarization, and the line splitting of Zeeman effect. SPARX can handle the coordinate of both spherical 1D/2D/3D and 3D nested-Cartesian mesh. The package basically is implemented by pure C with the parallelization utilized by MPI in the computing, and Python is used as the operating interface.

Numerical Method

The Monte Carlo method has been widely used for the physics where Knudsen number is relatively large ($Kn=\lambda/\Delta x>>1$) such as long-characteristics radiative transfer. M.R. Hogerheijde and F.F.S. Van der Tak (2000) proposed the accelerated Monte Carlo method (AMC) to preserve the accuracy of the Monte Carlo sampling. The accelerated idea separates two stages to utilize accelerated Lambda iteration (ALI), which was proposed by G.B. Rrbicki and D.G. Hummer (1991), for reaching spatial preliminary consistency at first stage then improve the random sampling resolution to the demanding accuracy at the next stage.

ALI differentiates the intensity inside a cell into the contribution from the internal intensity of the cell J_{int} and the contribution from the external cell J_{ext} in order to reduce the computational consumption of ray-tracing. Between the iteration, ALI samples J_{ext} once then performs detailed balance calculation to make J_{int} and population self-consistent.

The AMC scheme was further modified and adjusted for MPI parallelization. A higher signal-to-noise-ratio, compared to the converged criterion (4/3 SNR is adopted), is used for the criterion to diminish the MC error by doubling the ray number. Purely MC error is considered in Stage 2 of AMC without being mixed with iteration error, due to the reason for the parallelization, an additional Stage 3 is to perform fixed ray sampling by the ray number at final Stage 2 and eliminate the error of Lambda iteration. This 3-stages scheme has been validated that the solution can acquire the demanding Monte Carlo error.



The physical model containing the distribution of density, temperature, velocity, molecular abundance and dust properties

Stage 1:
Use the sampling by the fixed set of random ray to perform ALI till reaching 1/10 required accuracy.

Stage 2:
Compare the error between three sets of fully random ray. If the error exceed the required accuracy, double the ray number for the corresponding cell to improve sampling resolution.

Stage 3:
Use the final ray number of the previous stage to perform ALI with the fixed ray till reaching 1/10 required accuracy

Obtain the external radiation J_{ext} by using Monte Corlo sampling

Obtain the internal radiation J_{int} by using the current local level population

The total intensity $J = J_{ext} + J_{int}$
Obtain the mean intensity $\bar{J}$ by averaging all the direction.

Construct the level equilibrium matrix including the combination of the spontaneous emission, the stimulated emission, and the gaseous collision

Make matrix inversion to obtain local level population

The population converged?

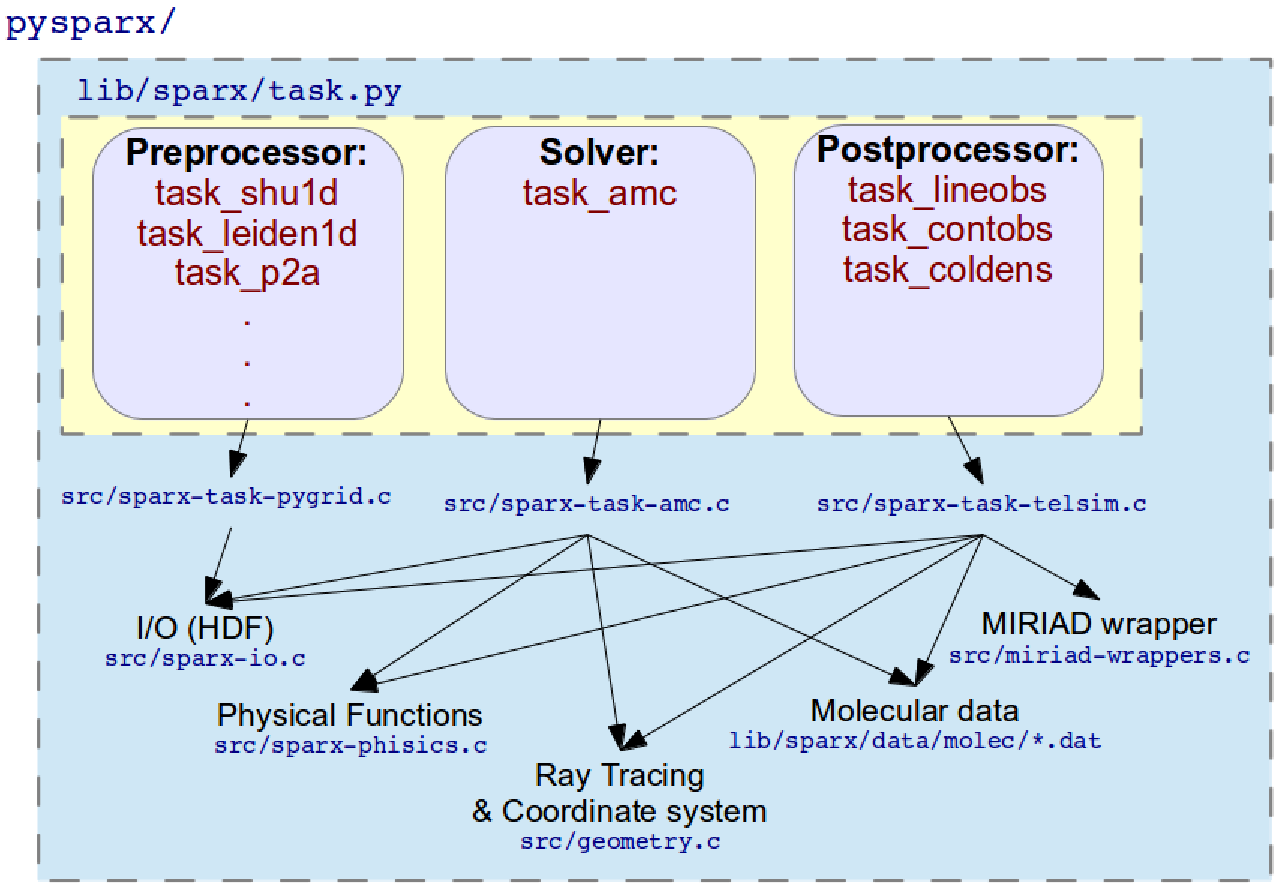
move to the next cell

All cell converged between iteration?

molecular level population

The Code Structure

All the tasks in SPARX are defined in pysparx/lib/sparx/tasks.py .



Installation

- **Prerequisites**

SPARX depends on the following packages with the related versions

- Hierarchical Data Format (HDF5) ≥ 1.8.4
- GNU Scientific Library (GSL) ≥ 1.13
- Fastest Fourier Transform in the West (FFTW) ≥ 3.2.2
- Miriad ≥ 4.1.3
- Python ≥ 2.5
  - Numpy ≥ 1.4.0
  - Matplotlib ≥ 0.99.1
  - Pytables ≥ 2.1.2

When presuming to install on TIARA's machines, just put the command in *.bashrc* to add these modules.

```
module add autoconf/2.68 automake/1.11.1 HDF/5-1.8.2_ic11.0 miriad/2011.7_gfortran
gsl/1.5_ic11.0 HDFView/2.1 libtool/2.4 icc/11.0 lam/7.1.4_ic11.0
```

Setup

Then, download SPARX source code and extract

```
wget https://www.asiaa.sinica.edu.tw/~ithsieh/SPARX/sparx-alpha-2.tar.gz -O- | tar -xz
```

Install the package

```
cd pysparx
./install
```

The installed package is under $HOME/opt/sparx . We may add the executable binary path in *~/.bashrc*

```
PATH=$PATH:$HOME/opt/sparx/bin
export PYTHONPATH=$HOME/opt/sparx/lib/python2.5/site-packages:$PYTHONPATH
```

Test

After reloading the .bashrc script, type the command “sparx” to test it. The screen output shall come.

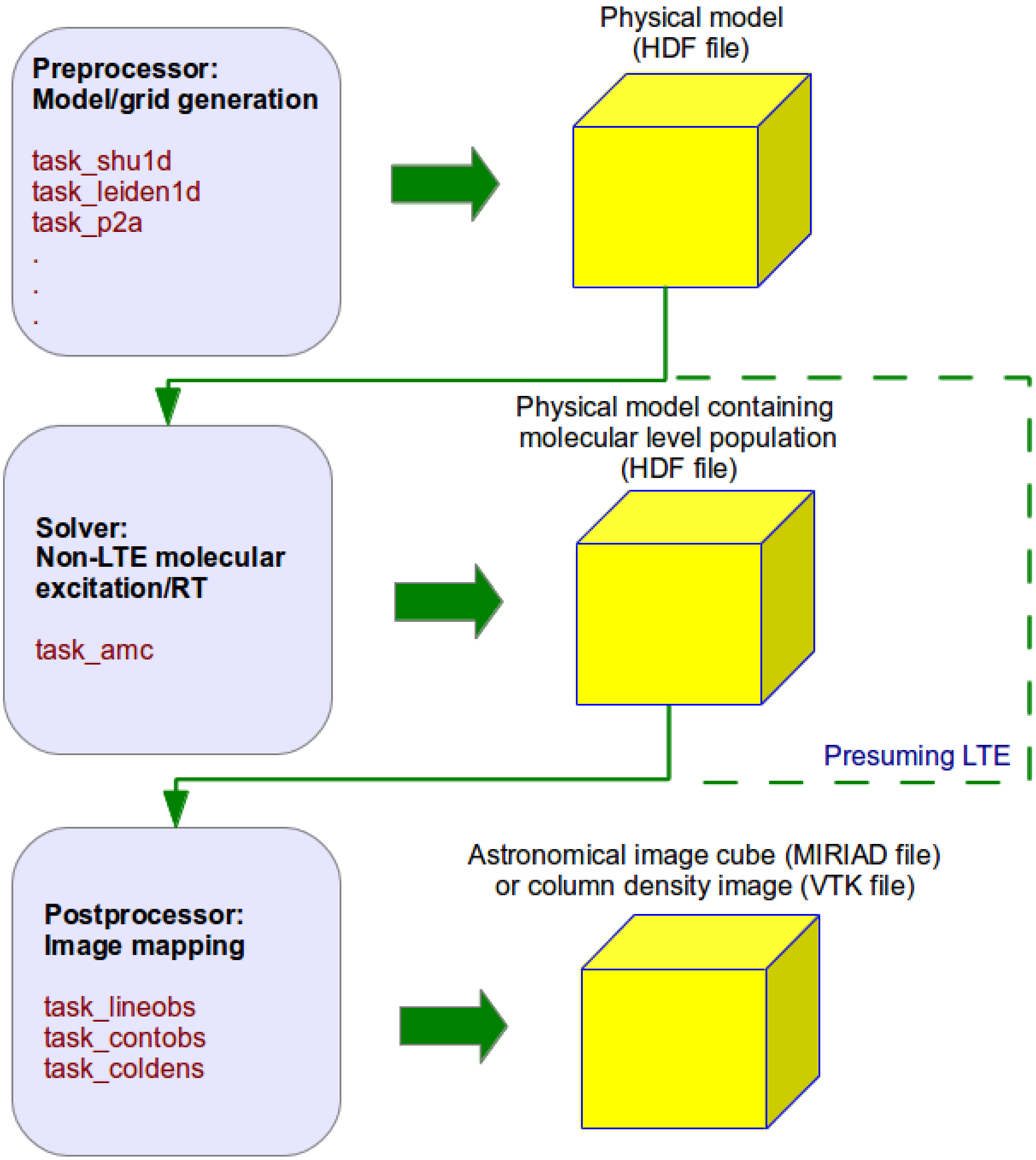
```
Usage: sparx [OPTIONS] COMMAND [TASK] [TASK OPTIONS] [KEY1=VAL1 KEY2=VAL2 ...]
Type 'sparx help COMMAND' for help on COMMAND
Type 'sparx help TASK' for help on TASK

COMMAND can be one of the following:
 help
 run

TASK can be one of the following:
 task_amc
 task_coldens
 .
 .
 .
```

Tutorial / Usage

■ Overview of the package

SPARX provide preprocessor, molecular/excitation solver, and postprocessor to produce emission image or column density image. Type “sparx” would print out all the task name and the syntax.

To know the options of each task, type “sparx help TASK_NAME” ; and “sparx run TASK_NAME options” for running the task.



Build a model for 1D Shu's collapsing cloud

To build a 1D Shu's collapsing cloud (F.H. Shu,1977), we have built the task task_shu1d

```
sparx run task_shu1d out=model n_max='1e12m^-3' ndiv=64 abundance=1e-9 Vin='0.1kms^-1'
```

Here we put the characteristic density $10^{12}m^{-3}$ and the infalling speed 0.1km/s on the radius 0.01pc, the model and the corresponding griding shall be described in the HDF file model.

Run the molecular excitation solver AMC

```
sparx run task_amc source=model out=pops molec='hco+' lte='True' snr=20 tolerance=5e-3
```

Here the molecule HCO^+ is used, initial condition based on LTE (local thermal equilibrium) assumption, the requested SNR (signal-to-noise ratio) is 20, and the stopping criteria of the Lambda iteration is $5x10^{-3}$ which is one order less than the demanding Monte Carlo error corresponded to SNR=20 . In task_amc, "lte" flag determines the initial condition for molecular level populations at the beginning of the Lambda iteraton. "lte='True'" initiates the populations to LTE assumption, otherwise "lte='False'" means the excitation calculated all from the ground state.

Mapping the image

```
sparx run task_lineobs source=pops out=map line=0 dist='1kpc' cell="['1asec','1asec']"
npix="[128,128]"
```

To synthesize the image of the molecular line emission, we project the radiation on the certain orientation to the plane of sky. The input file should be HDF contained griding, the physical condition, the level population, and the specific molecule in the former AMC output file pops. If we would like to see the emission under LTE assumption, we could turn on the "lte='True'" flag here and just use the "source=model" to quickly produce LTE line image.

Plot the physical properties and level populations

"sparx-plot" command could plot the 1-D properties directly on the X-window. The usage is sparx-plot [OPTIONS] PARM FILE1 FILE2, for example,

```
sparx-plot lev0 pops
```

This demand would plot the fractional population of level 0 in the dataset pops. PARM should be "n_H2" for the gas density, "T_k" for the kinetic temperature, "X_mol" for the molecular abundance, "V_cen" for the radial velocity in 1-D, and "lev0/lev1/lev2..." for the molecular level populations.

The command "sparx-plot" should require "Tkinter" module support in PYTHON to activate the graphical interface.

Create 1D model

To build a 1-D preprocessing PYTHON module in SPARX, we could insert a new task in pysparx/lib/sparx/tasks.py by just modifying the template inside the scripts.

For example, in the preprocessing task called "task_shu1d" defines the outer and inner boundaries of the 1-D geometry as radius_out and radius_in, number of the spherical cell for n, mass ratio of gas to dust for grid.gas_to_dust, number density of molecular hydrogen in MKS unit for grid.n_H2, kinetic temperature of the gas for grid.T_k, molecular abundance for grid.X_mol, turbulent speed for grid.V_t, the kinematic radial velocity for grid.V_i .

Create 2D/3D spherical model

To create a spherical 2-D (R, θ coordinate) or 3-D (R, θ, Φ coordinate) case, there is a preprocessing PYTHON template available on-line
https://www.asiaa.sinica.edu.tw/~ithsieh/SPARX/preprocess2D.py
This template is built to generate a spherical 2-D early-type planetary disk and envelope model. The Φ-directional resolution *np* determines the number of the cell on Φ-axis. While

np=1 means it's 2-D Φ-symmetric ring-like cells.

Create 3D Cartesian model

Here is the template for generating a model of the Shu's collapsing cloud in Cartesian nested grid.
https://www.asiaa.sinica.edu.tw/~ithsieh/SPARX/nested3D.py

Parallelize AMC computing (on TIARA machines)

A template of the submitting script could be referred from
/lustre/home/ydhsieh/test_hyosun/run
After modifying the demanding number of nodes and the processors per node by defining the line

```
#PBS -l nodes=8:ppn=12
```

and change the working directory

```
PBS_O_WORKDIR=YOUR_WORKING_DIRECTORY
```

than use the command qsub to submit the parallel job

```
qsub THE_SCRIPT
```

resume AMC by replacing the final result of level population as the initial condition

Enable the "`trace='TRUE'`" flag in task_amc job to record the result of all the level population in the every step of the iteration, than, after the job being terminated, put the last record into the new task_amc run, by assigning `pops=THE_LAST_POPULATION` ,to consider the data of the population as the initial condition for this run.

Dust continuum map

The task "task_contobs" can be used to produce the map of the continuum emission

```
sparx run task_contobs source=POPULATION_FILE out=IMAGE_FILE
wavelen=WAVE_LENGTH ....
```

"wavelen" denotes the wave length of the radiation. For example, using '850um' represents 850μm emission.

Zeeman effect

Zeeman effect takes place when the magnetic field is so strong that the molecular spin is quantumized. Molecular hyperfine structure shall split the line emission into two counter-polarized and counter-frequency-shifted spectra.

"zeeman" option would give the polarized quantity stokes Q for estimating the B-field along the line-of-sight.

```
sparx run task_lineobs source=.... out=..... zeeman='TRUE'
```

The molecular line fitting

The molecular line fitting module utilizes simulated annealing method to approach the optimized solution by gradually narrowing down the corresponding temperature

```
sparx run task_linefitting target=SPECTRUM_DATA kmax=100 X0=1e-9 molecule='hco+'
stopNR=1e-2
```

The five flag are the target spectrum using to approach, the SA algorithm's maximal iteration number, the initial guess of the molecular abundance, the molecule to approach, and the final tolerant error.
Detailed implementation needs to look into pysparx/lib/sparx/tasks.py and should revise how

to determine the error between observational spectrum.