



## User Guide to ARCiS

Michiel Min

June 14, 2023

# 1 Introduction

## 1.1 Terms of use

By using ARCiS you agree to the following:

- If in doubt on any of the results, you consult with me. Email: [M.Min@sron.nl](mailto:M.Min@sron.nl)
- You cite the appropriate papers listed below.

The most important reason for this is to make sure that ARCiS is used in a correct way and the result are scientifically useful. ARCiS is a complex code which can do a lot of things, this also means things can go wrong. Please refer to [Min et al. \[2020\]](#) for the first full description of the fundamental properties of the code.

Note that there are several parts of the code from different developers:

- Cloud formation framework: [Ormel and Min \[2019\]](#)
- Optical properties of cloud particles computed using DHS: [Min et al. \[2005\]](#), [Toon and Ackerman \[1981\]](#)
- Refractive indices for the cloud species, see references in [Min et al. \[2020\]](#)
- Molecular opacities: [Chubb et al. \[2021\]](#) and references therein
- Multinest Retrieval tools: [Feroz and Hobson \[2008\]](#), [Feroz et al. \[2009, 2019\]](#)
- GGchem when including chemistry: [Woitke et al. \[2018\]](#)
- Disequilibrium chemistry implementation: [Kawashima and Min \[2021\]](#)
- Diffusion implementation for 3D structures: [Chubb and Min \[2022\]](#)
- Coupling with planet formation parameters: [Khorshid et al. \[2022\]](#)

## 2 Installing ARCiS

Before installing ARCiS you need:

A Fortran compiler: This can be either `gfortran` or `ifort` (any other might work but is not tested).

`cfitsio` library: This is needed to allow fortran to read and write binary fits files.

`MultiNest`: This allows `ARCiS` to perform Bayesian retrievals.

On a Mac the easiest is to install `cfitsio` through HomeBrew (google 'homebrew' for installation instructions. After that:

```
# brew install cfitsio
```

Next fetch `MultiNest`:

```
# git clone https://github.com/JohannesBuchner/MultiNest.git
# cd MultiNest/build
# cmake ..
# make
# sudo make install
```

Next fetch the git source code from:

```
# mkdir ARCiS ; cd ARCiS
# git clone https://github.com/michielmin/ARCiS.git ./src
# cd src
# make gfort=true multi=true
```

This creates the `ARCiS` binary, which you can put in any path accessible (`/usr/bin` or something like that).

You also need data files to be stored under: `$(HOME)/ARCiS/Data/`  
These data files can be downloaded from <http://www.exoclouds.com> using the password: `ARCiSData`

### 3 Using `ARCiS`

To run `ARCiS` you further only need an input file. On the prompt type:

```
# ARCiS inputfile.dat -o outputdir
```

which creates the output directory `outputdir` containing the output files.

There are several options included in ARCiS. These are given as keywords in the `inputfile.dat` file (or whatever you call it). Keywords are always given as `key=value` and can be anywhere in the file (order does not matter). Also, you can overwrite keywords set in the input file from the command line in the following way

```
# ARCiS inputfile.dat -o outputdir -s key1=value1
```

Any number of keys can be set on the command line. Just make sure the first argument of the command line is the name of your input file. Note that ARCiS always takes the last keyword value it encounters, first reading the input file, next the command line keywords one by one.

## 4 Keywords

There are many keywords for a variety of different setups. Chemistry or not, cloud formation or not, various temperature structures, perform retrievals, 3D or 1D structure, and many more options. Below are some very basic setups discussed. The runtime of a single forward model can vary from 0.01 second to 15 minutes depending on your choices. For more information it is best to get in contact.

### 4.1 Base properties

**Rp** Radius of the planet in Jupiter radii.

**Mp** Mass of the planet in Jupiter masses.

**Pp** Atmospheric pressure corresponding to radius Rp. Default is 10 bar.

**Tstar** Temperature of the host star in K.

**Rstar** Radius of the host star in Solar radii.

**distance** distance to the system in parsec.

**Dplanet** Distance of the planet to the star in AU.

**planetname** Name of the planet to read from the database. Radius, mass and distance of the planet and the star are read from the database.

## 4.2 Grid setup

**pmin, pmax** Minimum, maximum pressure considered in the atmosphere

**nr** Number of pressure points

**lmin, lmax** Minimum, maximum wavelength considered in micron. Note that for temperature computations these must be set wide enough to ensure energy balance is properly computed.

**specres** Spectral resolution  $R$  in  $\lambda/d\lambda$

**specresdust** Spectral resolution for computation of the solid state species in the clouds.

## 4.3 Abundances of the molecules

Homogeneous abundances can be set using keywords like e.g.  $H_2O=1d-4$ . Only molecules that are defined through this somewhere in the input file are taken into account. These abundances are overwritten when chemistry is used.

**chemistry** Logical determining if chemistry is computed or not (either .true. or .false.)

**condensates** Logical determining if condensates should be taken into account in the chemistry computations (default is .false. and most stable is to leave it like that).

**COratio** C/O ratio of the atmosphere

**metallicity** Metallicity of the atmosphere

## 4.4 Opacities and raytracing

**cia** Logical determining if CIA is taken into account

**maxtau** Maximum optical depth considered for the raytracing

**compute** Logical determining if the opacities need to be recomputed from the linelists

**scattering** Logical determining if scattering of the thermal radiation is included

**scattstar** Logical determining if scattering from the star is included

## 4.5 Temperature structure

**computeT** Logical determining if the temperature structure is computed self-consistently

**maxiter** Maximum number of iterations for the temperature structure

**betaT** Cosine of the angle of incoming radiation.

**TeffP** Effective temperature of the radiation from inside the planet

**Tp** Temperature of the planet at 1 bar when `computeT=.false.`

**dTp** Temperature gradient when `computeT=.false.`

$$\log_{10}(T[\text{K}]) = \log_{10}(T_p[\text{K}]) + dT_p \log_{10}(P[\text{bar}]) \quad (1)$$

## 4.6 Retrieval

### 4.6.1 Observations

**obs1:type** Can be "trans", "emis" or "emisR".

**obs1:file** Filename with the observation. Should be in format:

column1: wavelength in micron

column2: trans or emis spectrum (same units as the output file to compare with)

column3: error

column4: spectral resolution of this wavelength bin (so  $\lambda/\Delta\lambda$ )

**obs1:beta** Weight of this observation. Only relevant if more than one obs is defined.

## 4.6.2 Parameters

**fitpar:keyword** Keyword to be retrieved. This key switches automatically to the next retrieval parameter.

**fitpar:min** Minimum value considered

**fitpar:max** Maximum value considered

**fitpar:log** Logical determining if the parameter is sampled logarithmically

## 4.7 Very rough instrument simulation (use at own risk!)

You can have ARCiS create simulated observations including estimate of the noise as function of wavelength.

*Note that this is absolutely not intended as a replacement of a proper instrument simulation! It only includes photon noise and can be used to get a very rough estimate of the expected performance.*

**instrument1:name** Can be "MIRI", "NIRSPEC", "WFC3", "JWST" or "ARIEL". When it is something else it is assumed to be a filename containing a proper instrument simulation (file format currently the ExoSim format).

**instrument1:ntrans** Number of transits to average over. When put to 0, the number of transits is computed to assure that at each wavelength 7 scale-heights can be observed with  $5\sigma$  accuracy.

## 5 Output files

There are many output files, most are for checking the model. The most important ones are discussed here.

**log.dat** This is the log file containing the runtime output.

**input.dat** Copy of the input file used appended with the command line keywords. This allows rerunning exactly this model again without command line keywords.

**mixingratios.dat** This file contains the temperature structure and the abundances of the molecules as a function of height in the atmosphere.

- trans** Transmission spectrum. Header explains units.
- emis** Emission spectrum. Header explains units.
- emisR** Emission spectrum relative to the stellar emission.
- .txt** File containing the MultiNest output when retrieval was done.
- bestfit.dat** File containing the input file for the best fitting model when retrieval was done.

## 6 Examples

There are a few examples in the Example directory downloaded with the git repository for retrieval and forward modelling.

## References

- K. L. Chubb and M. Min. Exoplanet Atmosphere Retrievals in 3D Using Phase Curve Data with ARCiS: Application to WASP-43b. *A&A*, 665:A2, Sept. 2022. doi: 10.1051/0004-6361/202142800.
- K. L. Chubb, M. Rocchetto, S. N. Yurchenko, M. Min, I. Waldmann, J. K. Barstow, P. Mollière, A. F. Al-Refaie, M. W. Phillips, and J. Tennyson. The ExoMolOP database: Cross sections and k-tables for molecules of interest in high-temperature exoplanet atmospheres. *A&A*, 646:A21, Feb. 2021. doi: 10.1051/0004-6361/202038350.
- F. Feroz and M. P. Hobson. Multimodal nested sampling: an efficient and robust alternative to Markov Chain Monte Carlo methods for astronomical data analyses. *MNRAS*, 384(2):449–463, Feb. 2008. doi: 10.1111/j.1365-2966.2007.12353.x.
- F. Feroz, M. P. Hobson, and M. Bridges. MULTINEST: an efficient and robust Bayesian inference tool for cosmology and particle physics. *MNRAS*, 398(4): 1601–1614, Oct. 2009. doi: 10.1111/j.1365-2966.2009.14548.x.
- F. Feroz, M. P. Hobson, E. Cameron, and A. N. Pettitt. Importance Nested Sampling and the MultiNest Algorithm. *The Open Journal of Astrophysics*, 2(1):10, Nov. 2019. doi: 10.21105/astro.1306.2144.



- Y. Kawashima and M. Min. Implementation of disequilibrium chemistry to spectral retrieval code ARCiS and application to 16 exoplanet transmission spectra. Indication of disequilibrium chemistry for HD 209458b and WASP-39b. *A&A*, 656:A90, Dec. 2021. doi: 10.1051/0004-6361/202141548.
- N. Khorshid, M. Min, J. M. Désert, P. Woitke, and C. Dominik. SimAb: A simple, fast, and flexible model to assess the effects of planet formation on the atmospheric composition of gas giants. *A&A*, 667:A147, Nov. 2022. doi: 10.1051/0004-6361/202141455.
- M. Min, J. W. Hovenier, and A. de Koter. Modeling optical properties of cosmic dust grains using a distribution of hollow spheres. *A&A*, 432(3):909–920, Mar. 2005. doi: 10.1051/0004-6361:20041920.
- M. Min, C. W. Ormel, K. Chubb, C. Helling, and Y. Kawashima. The ARCiS framework for exoplanet atmospheres. *Modelling philosophy and retrieval. A&A*, 642:A28, Oct. 2020. doi: 10.1051/0004-6361/201937377.
- C. W. Ormel and M. Min. ARCiS framework for exoplanet atmospheres. The cloud transport model. *A&A*, 622:A121, Feb. 2019. doi: 10.1051/0004-6361/201833678.
- O. B. Toon and T. P. Ackerman. Algorithms for the calculation of scattering by stratified spheres. *Appl. Opt.*, 20(20):3657–3660, Oct. 1981. doi: 10.1364/AO.20.003657.
- P. Woitke, C. Helling, G. H. Hunter, J. D. Millard, G. E. Turner, M. Worters, J. Blecic, and J. W. Stock. Equilibrium chemistry down to 100 K. Impact of silicates and phyllosilicates on the carbon to oxygen ratio. *A&A*, 614:A1, June 2018. doi: 10.1051/0004-6361/201732193.