

DOCUMENTATION

CHIMES (0.6)

JACQUES LE BOURLOT

AND THE ISM TEAM OF PARIS OBSERVATORY

December 27, 2020



CHIMES

CHemical Interstellar Medium Evolution Solver

1	Introduction	3
2	Installation and Source Files	5
2.1	Input Files	5
2.2	Input parameters	6
2.2.1	Base_Param.dat	6
2.2.2	Base_Speci.dat	7
2.2.3	Base_Chemi.dat	8
2.3	Output files	9
2.3.1	Screen output	9
2.3.2	Standard files	9
2.3.3	Optional files	10
2.4	Source files	11
3	Chemistry	13
3.1	H ₂ Formation rate	13
3.2	Special reaction type “101”	13
4	Steady state	15
4.1	Conservation equations	15
4.2	Stability	15
5	Maxima and Minima	17
6	Fourier Transform	19
7	Chemical analysis	21
8	Non-dimensional evolution equations	23
9	Depletions	25
9.1	Initial conditions	25
9.2	C and O depletions	25
A	Maximum interpolation	27

Introduction

This is version 1.0 of the documentation of the chemical evolution code CHIMES. Its aim is to solve the evolution equations of the abundances of chemical species within a dark molecular cloud, starting from user chosen initial conditions. This code has been used for all computations done in [Roueff & Le Bourlot \(2020\)](#). The evolution equations are described in [Section 8](#).

The code derives from a rather ancient version of the MHD shock code by G. Pineau des Forêts and D. Flower, in its mono fluid version, but it has been heavily updated and modified. It is a “0D” code, where no structure is assumed. We follow the evolution of a single cell of gas, “deep” within a molecular cloud. This depth is controlled by a given A_V (see parameters below, [Section 2.2](#)).

It is available from <https://ism.obspm.fr>.

This documentation describes how to install and use the code ([Section 2](#)), with a short description of the input files ([Section 2.1](#)) and the source files ([Section 2.4](#)). All input parameters of the main input files are detailed in [Section 2.2](#).

The following sections are more specific, and detail some specificities of the physics used in the code.

Installation and Source Files

Unpack the archive in a fresh directory (Hereafter, the *main* directory). There are four sub-directories:

1. `Data`: Contains all the input files.
2. `Doc`: Contains this file and the original paper (Roueff & Le Bourlot, 2020).
3. `Out`: Initially void. All outputs are written here.
4. `Src`: Contains the source files and a Makefile.

Go to directory `Src`. Check the file `Makefile`, and adapt it to your preferred Fortran compiler. Currently, `gfortran`¹ is used, but the code has also been tested with `ifort`². The mandatory libraries `BLAS` and `LAPACK` are usually provided with the compiler.

Build the executable by running:

```
$> make
```

The main code `Chem_Evol` is created locally and copied to the main directory (above `Src`). An auxiliary tool `Chem_Analys` is also created (see Section 7 for its usage).

Goto directory `Data`. It contains four input files whose names are `Pap_xxx.dat`. `xxx` is a mandatory part that can be `Param`, `Speci`, `Deple`, `Chemi`, as described in Section 2.1. `Pap` is here the *Base* name, used to create a subdirectory `Out/Base` containing the output files (Thus, here it would be `Out/Pap`), and used to run the code from the main directory:

```
$> Chem_Evol Base
```

where `Base` is the base name (Thus, here it would be `Chem_Evol Pap`).

Edit the input files to fit your needs, then come back to the main directory and run the code. Some indications on the integration are written on a single line after each time step.

For standard usage, one does not need to edit the sources files. See Section 2.4 if more informations are needed.

2.1 Input Files

Input data are gathered in the sub-directory `Data`. There are four different files. The naming scheme is as follow:

- A `Base` name is used for all files. It must be provided as an argument to the executable code on the command line. If none is given, then the generic name `Base` is used. Current input files use the name `Pap`.
- `Base_Param.dat`: Most input parameters and Flags. See description in Section 2.2.

¹<https://gcc.gnu.org/wiki/GFortran>

²<https://software.intel.com/content/www/us/en/develop/tools/oneapi/components/fortran-compiler.html>

- `Base_Speci.dat`: Chemical species list, with detailed composition, and tentative initial abundances (these **will** be modified after reading, see below).
- `Base_Deple.dat`: Total depletion for each atom. Used to set, e.g., the C/O ratio or the metallicity. After reading both `Base_Speci` and `Base_Deple` files, initial abundances are adjusted so that total depletions are enforced to within machine precision. The correction is applied to specific species that are believed to be among the most abundant in a standard dark cloud. This is done in routine `ajust_abin`. If a satisfying solution is not found, the program stops. It is then the responsibility of the user to modify the initial abundances in `Base_Speci.dat` so that a solution may be found. This procedure may be replaced “one day” by something more robust.
- `Base_Chemi.dat`: Chemistry file. It uses a format similar to the PDR code.

Using this “base name” scheme is a flexible way of having different cases simultaneously. Experience shows that initial conditions must be adapted multiple times to illustrate various physical behaviors. This helps to keep track of these modifications together with a specific location for the output files.

2.2 Input parameters

2.2.1 `Base_Param.dat`

`Base_Param.dat` gathers most control parameters. There is a single parameter per line to allow a short comment to be set in this file. Currently the parameters are, from top to bottom:

- `F_St_Eq`: Equation of state of the gas. There are currently 3 possibilities:
 - 1: Constant temperature T and density n_{H} . Hence no thermal balance is computed, and routine `source` is not used.
 - 2: Isochoric (constant density) case, with thermal balance computation. The heating and cooling sources have not been checked since the initial version of Flower et al. in the '90s. Use with caution.
 - 3: Isobaric (constant pressure) case, with thermal balance computation. Same caution as above.
- `densh`: H atoms number density $n_{\text{H}} = n(\text{H}) + 2n(\text{H}_2) + n(\text{H}^+) + \dots$, in cm^{-3} . This is **not** the gas number density ($\sum n(\text{X})$ over all species), which is variable `xdens` in the code.
- `tempr`: (Initial) gas temperature. Kept fixed if `F_St_Eq == 1`.
- `zeta`: Cosmic rays ionization rate (s^{-1}).
- `rad`: External radiation field intensity. No test has been done using a strong field.
- `av`: Position within the cloud of the gas cell under study. Currently this is expected to be 10 or more, so that the influence of ultraviolet photons remains negligible. This limitation could be removed one day.

- `rop`: Ortho/Para ratio of H₂ (O/P). Although H₂ rotational excitation is computed, it is currently not used to compute O/P. There are two possibilities to set O/P: Any value of `rop` between 0 and 1 is used “as is”, and never changed during the run. If `rop` < 0 (typically -1), then it is computed as a function of the gas temperature at steady state. Note that this does **not** account for formation on grain effects.
- `tgr`: Dust temperature.
- `ratiog`: Dust to gas mass ratio.
- `radgr`: Grain harmonic mean radius (i.e.: $\sqrt{a_{min} a_{max}}$).
- `rhogr`: grain density (gcm⁻³).
- `h0`: Maximum integration step size (Myr). It must be adjusted by the user depending on the kind of information sought. E.g.:
 - For standard integration towards a stable steady state, use about 1/100th of the total integration time.
 - For FFT computation, use a bit less than 1/10 of the oscillations period.
 - For a precise determination of the abundance maxima of various species, use a very short `h0`, but increase `ks` (see below) to limit output file sizes.
- `tend`: Final integration time (Myr). It should be adjusted to ensure that the transitory part of the evolution, associated with initial conditions, has been fully dissipated. Beware of critical slow down if the system evolves close to a bifurcation point.

A comment line is inserted here in the file. Standard usage seldom requires any modification below.

- `tper`: Estimated oscillations period (if any). Used internally to tune the computation to some specific cases (e.g., FFT computation). Naturally, this requires already a fairly good understanding of the physical situation. Not used in normal usage.
- `ks`: Computed state is outputted every `ks` integration step. Useful to reduce output files size.
- `FFFT`: Flag controlling the computation of the Fourier transform of some variables. (1: compute!, 0: don't). See Section 6.
- `FCHA`: Flag controlling the output of files used by the chemistry analyzer. See Section 7.
- `frnH`, `poc1`, `poc2`: Specific tuning, to be adjusted together with modifications of the source files. Do not use for normal runs. Leave values “as is” (normally -1, meaning “do nothing”). See also Section 9.

2.2.2 Base_Speci.dat

List of species whose abundance is computed in the code, with one species per line. The format is identical to the PDR code (up to column 64, after which only comments are found). In fixed form, the FORMAT is:

```
(i3,2x,a8,1x,2i2,3i1,1x,4i1,1x,5i1,1x,6i1,1x,i2,1x,1pe10.3,0pf9.3,1x,a23)
```

This is used to read respectively:

- `l`: an index used to mark the end of the file (`l=888`).
- `speci`: species name.
- `ael`: stoichiometric composition of `speci`.
- `abin`: Initial abundance.
- `enth`: Reference enthalpy.
- `observ`: A comment.

`ael(j,i)` gives the number of atom `j` in species `i`. The list of atoms with their indexing is given in the following Table.

<code>j</code>	atom	<code>j</code>	atom	<code>j</code>	atom
1	H	2	C	3	N
4	O	5	He	6	D
7	¹³ C	8	¹⁵ N	9	¹⁸ O
10	Ar	11	F	12	Na
13	Mg	14	gr	15	Si
16	Ne	17	S	18	Cl
19	Ca	20	Fe	21	<i>ch</i>

Note that N° 14 is not an atom, but may be used for a very limited interaction with grains. N° 21 is the charge (not the number of electrons). So $ch = 1$ for positive ions (C^+ , ...) and $ch = -1$ for negative ions.

2.2.3 Base_Chemi.dat

List of chemical reactions between species. One reaction per line. The format is identical to the PDR code. In fixed form, the format is:

```
(a5,1x,a8,4a8,a7,e8.2,1x,f5.2,1x,f8.1,i4)
```

This is used to read respectively:

- `reff`: A reference code (the list is available in the PDR code).
- `r1`, `r2`: reactant species (three body reactions are not yet included).
- `p1`, `p2`, `p3`, `p4`: Up to four products.
- `gamm`, `alpha`, `bet`: Coefficient of the Arrhenius like reaction rate.
- `itype`: Reaction type.

See routine `chimie` in `Chem_Sub.f90` for the way rates are computed for each reaction type. See also the PDR code documentation, as the usage is mostly the same.

2.3 Output files

2.3.1 Screen output

Some initializations are fast written on screen and disappear. They can be found in the `Base.out` file.

Then, during execution, the code writes a single line on screen every `ks` return from the integration routine `VODE`, with some basic informations. Namely:

- `k`: index of iteration
- `hs`: time step used. This should converge fast towards `h0` (as read in `Base_Param.dat`).
- `ts`: time reached (in Myr)
- `temp`: Gas temperature
- `ne`: $n(e^-)$
- `nH`: $n(H)$
- `nh2`: $n(H_2)$
- `o/p`: ortho to para ratio of H_2 .

At the end of the integration, a steady-state is computed (see Section 4), and the errors on abundances and their evolution written on screen. For advanced users, the stability of this steady-state can be evaluated. This requires editing the source code.

If the flag `F_EIGEN` is set to 1 in `Chem_Sub.f90`, then some corresponding eigenvalues are computed and given on screen. Namely:

- Eigenvalues with significantly positive real parts.
- Eigenvalues with non 0 imaginary parts.

By default, this flag is set to 0 as the information given are tricky to analyse, and probably not useful for most users.

Finally, two estimates of the period computed with the last three maxima of H are given if the uncertainty is low enough (currently to within 10^{-3}). If not, a warning message is printed. This last line may be filtered in a script shell to explore parameter space.

2.3.2 Standard files

Four standard output files are written. Their names use the “Base” name with various extensions:

- `Base.graph`: Contains the full time evolution of the run. Used to plot variables.
 - First line is a header line, with the name of the variables in each column, beginning with the time t .
 - * Columns 2 to 4: ζ , T and n_H .
 - * Then all abundances (neutral, ions, electrons)
 - * Then a handful of variables linked to thermal balance. See comments in the code for their meaning.

- All following lines: values at t .
- `Pap.out`: Various data and statistics on the run, such as all parameters, the chemistry, ... This file is subject to changes depending on the users requirements.
- `Pap.max`: values of t_i and $n(\text{H})$ at t_i for all maxima of atomic hydrogen. Various tries have shown that H is well suited to tracking the evolutions of possible oscillations. The choice of variable can be changed in the source. This is useful to determine the relaxation time towards a permanent regime and/or the period.
- `Pap.fin`: For each species, its main characteristic after dissipation of the transitory regime. Each line gives:
 - An index
 - The species name
 - The steady state abundance³ computed at the end of the run using a Newton-Raphson scheme (see Section 4)
 - The time t_{min} and abundance n_{min} of the last minimum
 - The time t_{max} and abundance n_{max} of the last maximum
 - The ratio $I_{ep} = \frac{n_{\text{H}}}{\zeta}$
 - The temperature T
 - The ratio $\frac{n_{max}}{n_{min}}$
 - The time interval $t_{max} - t_{min}$

Note that if $t_{max} - t_{min} < 0$ a full period should be added to it. This is not checked in the output since an accurate value of the period cannot be computed automatically in every cases.

2.3.3 Optional files

Two optional files are written if required:

- If `F_CH_A = 1`, then data files fit for chemistry analysis are written. These files are not meant to be read by the user, but should be used with `Chem_Analys` tool (see Section 7).
- If `F_FFT = 1`, then a Fourier transform of the time evolution of some abundances is taken (See Section 6).

The Fourier transform is written in file `Base.fft`, also in the `Out/Base` directory. In the current version, 6 different variables are computed, which are arbitrarily set to C, CO, CH₄, CD₅⁺, NH₃ and SH. This can be set in the code main program. Experience shows that the informations gathered are essentially the same. Maybe we should simplify to a single variable (?).

³This is **not** the value at the end of the run!

2.4 Source files

The code itself is built from three specific files, and two well known publicly available routines. A further auxiliary tool is provided to help in analysis of the chemistry. All source are compiled using a `Makefile`.

- `Chem_Evol.f90`: Main source file. Calls initialization routines, then loops over the ODE integrator to get the time dependent solution, and eventually calls some post-processing routines.
- `Chem_Aux.f90`: Sets a few `PARAMETERS` (e.g. `dp`), and provides auxiliary routines. Up to now: FFT computation & integer to character converter `ii2c`. Module `Chem_Aux` **must** be USED in **all** routines.
- `Chem_Sub.f90`: Provides all computation routines, and contains most of the physics. Namely:
 - `init`: Reads parameters input files and initializes the run.
 - `lectur`: Reads species and chemistry input files.
 - `ajust_abin`: Adjusts species depletions to specific values (see Section 9).
 - `diffun`: Main routine to compute the differential equations. Called by the integrator. Hence its interface is fixed and cannot be changed.
 - `chimie`: Computes chemical rates (and optionally the Jacobian).
 - `source`: Computes most heating and cooling rates.
 - `coolH2`: Computes H₂ rotational excitation and cooling.
 - `steady_st`: Computes one system fixed point by a Newton-Raphson scheme starting from the final integration step. In the case of oscillations, this steady state is expected to be the closest to any limit cycle and to be unstable. This is checked by computing the Jacobian matrix eigenvalues. See also Section 4.
 - `max_ht`: Post-processing utility used to output only times and values of maxima of $n(\text{H})$. This is very useful to compute the transitory regime characteristic time scale, and to evaluate the period of oscillations if any.
 - `extrema`: Post-processing utility giving the times and values of the last maximum of all species. Useful to analyze the chemistry.
- `dvode_f90_m.f90`: VODE integrator module, adapted to Fortran 90. Used “as is”.
- `qeispack_Mod.f90`: Routine `RG`, which computes eigenvalues and eigenvectors of a real general matrix.
- `Chem_Analys.f90`: A chemistry analyzer tool, in the spirit of, but independent of, the analogous PDR code tool. Used for post-processing.

Compilation (call to `make`) is done within `Src`, and the executables `Chem_Evol` and `Chem_Analys` are then automatically copied in the parent directory. Note that `BLAS` and `LAPACK` libraries are required, but they are generally automatically available with the standard compiler `gfortran`.

Note that compilation of `dvode_f90_m.f90` and `qeispack_Mod.f90` usually generates many warning, since these routines have not been adapted to recent Fortran usage.

Very few tuning needs to be done in the code itself. Most options are controlled through parameters read from the input files (See Section 2.2). The few exceptions are explained below. The

user should not modify the code unless he/she knows exactly what he/she is doing. This is still “Work in Progress”!

The code should be run from the main directory (above `src`) by:

```
$> Chem_Evol Base
```

where `Base` is the base name of all input files. A subdirectory `Out/Base` is then created where all output files are gathered.

The current chemistry file does not use grain chemistry at all. Very limited neutralization of ions on grains is possible. This is currently “work in progress”.

Photo-destruction reactions are included. However, the influence of the radiation field intensity has not been tested.

The current chemistry file includes a very large and exhaustive deuteration scheme, going all the way to full deuteration (e.g. the formation of CD₄ from CH₄). See [Roueff & Le Bourlot \(2020\)](#) for a description.

3.1 H₂ Formation rate

H₂ formation on grains does not use adsorbed species on grains as is done in the PDR code, but only “old style” rates. This is a fair approximation in that kind of computations, but should be revised one day. The same formalism is used to form H₂, HD and D₂ on the surface of grains.

These rates read:

$$k = \gamma \frac{n_H}{n(X)} \left(\frac{T}{300} \right)^\alpha$$

where $\gamma = 5 \cdot 10^{-17}$ for H₂ and HD and $5 \cdot 10^{-22}$ for D₂, and X is H for H₂ and HD and D for D₂. This is an approximation that makes the following 2 hypothesis:

- The ratio of adsorbed H and D on grains is the same as in the gas (hence the factor 10^{-5} in the rate prefactor) \Rightarrow This should be changed to the real D/H ratio read from the input files.
- H impinging on a grain reacts only with adsorbed H, and never with adsorbed D.

This second hypothesis is necessary to get a consistent rate. The creation term is:

$$\frac{d[AB]}{dt} = k [A] [B]$$

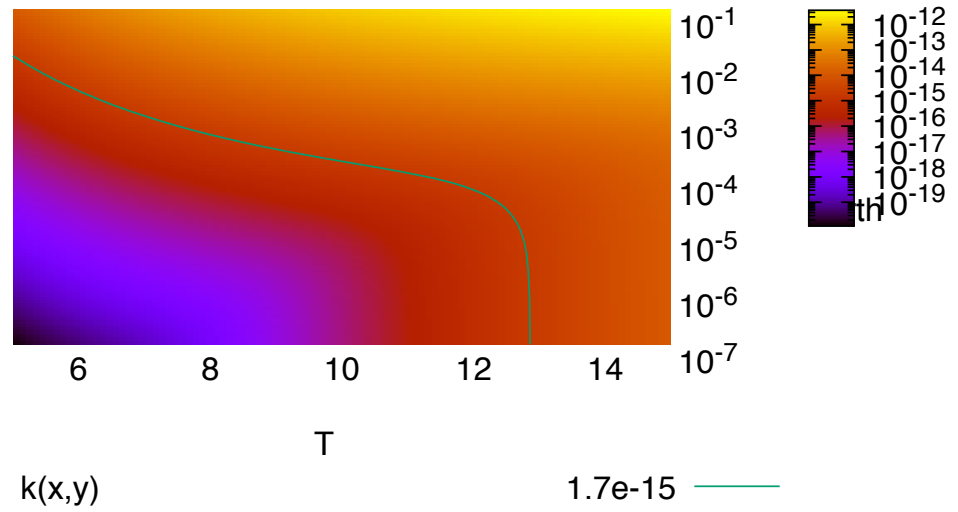
If A = B, then one multiplication by [H] (resp. [D]) cancels in k , as it should. But in the case of HD, only adsorbed H can be accounted for this way.

Full account of adsorbed species, as in the PDR code, is necessary to go beyond these approximations.

3.2 Special reaction type “101”

This is $N^+ + H_2 \rightarrow NH^+ + H$. Its rate comes from [Dislaire et al. \(2012\)](#). It is one of only two reactions that depend on H₂ excitation, through its O/P ratio. The other one is $H_2D^+ + H_2 \rightarrow H_3^+ + HD$, which – up to now – seems to play a very minor role in the oscillatory behavior.

The first reaction is critical. At $n_H = 1.75 \cdot 10^4 \text{ cm}^{-3}$, $T = 12 \text{ K}$ and $\zeta = 4.9 \cdot 10^{-17} \text{ s}^{-1}$, we find that oscillation are possible if its rate is below $1.7 \cdot 10^{-15} \text{ cm}^3 \text{ s}^{-1}$. Using the rate given by [Dislaire et al. \(2012\)](#), this rate is possible only for simultaneously low values of O/P and T , as seen below.



Note that the y axis here is the fraction of Ortho H_2 and not the O/P ratio itself. The difference is negligible in the region of interest.

Steady state

Computation of the nearest fixed point proceeds from the state X reached at the end of the final integration step. If X_0 is the required fixed point, at steady state we have:

$$F(X_0) = 0$$

Starting from $X = X_0 + \delta X$, we compute the correction δX from:

$$F(X) = F(X_0) + J(X_0) \delta X + o(\delta X)$$

So, we need to solve:

$$J(X_0) \delta X = F(X)$$

4.1 Conservation equations

For most species, the function $F(X)$ is the evolution term of our differential equation. However, for each atom present, we need to choose one species to ensure conservation of mass. Otherwise, the Jacobian matrix is singular. The relevant equation is:

$$\sum_i \alpha_{ji} x_i - \delta_j n_H = 0$$

where α_{ji} is the stoichiometric coefficient (number of atoms j in species i), and δ_j is depletion relative to H. The corresponding line in the Jacobian matrix has element :

$$\frac{\partial}{\partial x_k} \left(\sum_i \alpha_{ji} x_i - \delta_j n_H \right) = \alpha_{jk}$$

Note: within the code, the flag “F_Jac” must be set to “1” to compute the Jacobian in routine “chimie”. This is done automatically.

Since we are looking for dark cloud conditions, the species used for conservation are set in the code, using variables `iconv` and `lconv` (set in `ajust_abin`). In case of difficulties, the code needs to be recompiled. Currently, we use: H₂, C, N, O, He, HD, S, Fe⁺.

4.2 Stability

The character of the fixed point should be determined from an analysis of the eigenvalues of the Jacobian matrix. However, close to a bifurcation point, this matrix is extremely ill conditioned. Hence, it is not possible to determine exactly the position of a bifurcation point by finding the value of a parameter for which the highest eigenvalue crosses the 0 axis.

To test the accuracy of eigenvalues, we use two completely independent routines to diagonalize the Jacobian matrix, one from the LAPACK library, and an older but different one from the EISPACK library. Various tests have shown that they agree completely far from a bifurcation point, and also always for eigenvalues significantly different from 0. Close to 0, the order and exact values are unreliable. Therefore, we set the values to exactly 0 and do not try to use them.

For pairs of eigenvalues with positive real part, the imaginary part gives an indication on the oscillation period. These values are printed on screen at the end of the run if $F_EIGEN = 1$.

Maxima and Minima

During integration, the code records the time of occurrence of the last maximum and minimum for each species. This is done by keeping in memory the last 2 integration steps for each call to `VODE`. After each step, and for each species, we test whether:

$$n_2 < n_1 \text{ .AND. } n_2 < n_3$$

$$n_2 > n_1 \text{ .AND. } n_2 > n_3$$

where n_1 , n_2 and n_3 are abundances at 3 consecutive time steps t_1 , t_2 and t_3 . If one of the conditions is satisfied, the corresponding time and abundance (t_2, n_2) is recorded for this species. See Annex A for the equations used to interpolate these 3 points to an extremum.

The final values are saved in the file `Base.fin`, together with the steady state computed at the end of the run.

This file is used to analysis the phase shift in the evolution of all species.

Furthermore, all maxima (time and values) for one specific species are recorded in a specific file `Base.max`. Currently we use H (this can be changed in the code). This is used to evaluate:

- The duration of the transitory phase (for the first steps, where the abundance varies from one maximum to the next)
- The period (difference between two recorded times of maxima)
- The accuracy (fluctuations in the maxima values).

Fourier Transform

This is an useful tool to check for various independent oscillation processes in the chemical evolution. Up to now, it is computed from the evolution of H. This can be changed by editing the main source file.

A good spectral resolution (i.e. a small $\delta\nu$) requires integration during a long time. Recall that:

$$\delta\nu = \frac{1}{T_{rec}}$$

where T_{rec} is the total recording time used to compute a FFT. Conversely, a very high frequency is not necessarily required. Hence a rather crude time step h_0 can be used. As a rule of thumb, an h_0 a bit smaller than $1/10$ of the period is usually adequate (at least as a first try). It also means that an FFT can not be computed while the specific case has not been already studied.

To ensure a long enough integration, the default number of points in the input signal is set to 4096. This can be changed in the code. Furthermore, it is mandatory to wait beyond dissipation of the transitory phase (duration T_{trans}) before recording the first points. Thus, the necessary total integration time T_{tot} is:

$$T_{tot} = T_{trans} + T_{rec} = T_{trans} + 4096 h_0$$

T_{trans} is usually not known in advance. The default is set arbitrarily to $\frac{T_{tot}}{4}$. This should be changed if inspection of previous results show that it is not appropriate.

As an example, an oscillatory state far from a bifurcation has an transitory phase with a typical time scale of 1 Myr. Since this is an exponential damping, we can approximate $t_{trans} \sim 5$ Myr. If the period is about 0.7 Myr, then, we need to record the evolution during about 286 Myr. So, we can set T_{tot} (parameter `tend` in the code) to 400 Myr. This value accounts for the Sup of 5 Myr and $\frac{T_{tot}}{4} = 100$ Myr, with the remaining $\frac{3 T_{tot}}{4} > 286$ Myr.

Writing code to make all this completely automatic may be possible, but is quite cumbersome. So, some flexibility from the user is required at this stage.

Chemical analysis

To understand the chemical mechanism at work, it is necessary to identify the main reactions leading to formation and/or destruction of any species at a given time. To help in this analysis, a dedicated tool is provided: `Chem_Analys`. It uses specific files created by `Chem_Evol` if `F_CH_A=1` in `Base_Param.dat`.

During a run, the code saves (in the same directory as other output files) 5 files containing the full state of the evolution at equally spaced time steps set to $\frac{1}{4}$ of a typical period time T_{per} , as read from the input `Base_Param.dat` file. Since the exact period is not known in advance, it has to be provided by the user. The value read in the input file needs not be precise, and it can be used for zooming on a specific region. The current times used for each snapshot are:

$$t_i = T_{tot} - 1.2 T_{per} + (i - 1) \frac{T_{per}}{4}$$

Each snapshot is self contained. The redundant informations do not use so much space. We record.

- The main parameters of the run (n_H , T , ζ , O/P).
- The time stamp.
- All abundances.
- All individual chemical rates.
- The total abundance source term as used by the integrator (used for accuracy check).

The analyzer asks for:

- The name of the file containing the snapshot. The user must provide the full path name starting from the main `CHIMES` directory. E.g.: `Out/Pap/Pap_01.cha`.
- A threshold parameters used to output only the main reactions.
- Which species is examined.

The last question is asked recursively until the user enters "0".

For each species, the output provides:

- The abundance.
- The total source term, as computed by the analyzer from the chemistry, and the one used in the code. Ideally, they should match exactly. Any discrepancy may indicate a problem in the input files or the chemistry routines.
- The ordered list of creation reactions then of destruction reactions up to the given threshold.

Non-dimensional evolution equations

Evolution equations are of the form:

$$\frac{dn_i}{dt} = \sum_{\alpha,\beta} k_{\alpha,\beta} n_\alpha n_\beta + \sum_{\alpha,\beta} k_\alpha n_\alpha$$

where n_i is a species abundance in cm^{-3} , t is the time in s, $k_{\alpha,\beta}$ a chemical reaction rate in $\text{cm}^3 \text{s}^{-1}$ for binary collisions, and k_α an unary rate in s^{-1} .

The only one component reactions are destruction by cosmic rays or secondary photons (the external radiation field is negligible here). If x_i is the relative abundance of species i , so that $n_i = x_i n_{\text{H}}$, then:

$$\frac{dx_i}{dt} = n_{\text{H}} \sum_{\alpha,\beta} k_{\alpha,\beta} x_\alpha x_\beta + \sum_{\alpha,\beta} k_\alpha x_\alpha$$

Now write the reactions rates as:

$$k_{\alpha,\beta} = \frac{\zeta}{n_{\text{H}}} r_{\alpha,\beta}; \quad k_\alpha = \zeta q_\alpha$$

where $r_{\alpha,\beta}$ and q_α are dimensionless rate coefficients. The equation is now:

$$\frac{1}{\zeta} \frac{dx_i}{dt} = \sum_{\alpha,\beta} r_{\alpha,\beta} x_\alpha x_\beta + \sum_{\alpha,\beta} q_\alpha x_\alpha$$

This suggests to introduce a dimensionless time $\tau = \zeta t$. This allows to remove completely the density n_{H} and cosmic rays ionization rate ζ from the equations and to use a single parameter $I_{ep} = \frac{\zeta}{n_{\text{H}}}$ to measure the relative weight of one body and two bodies reactions. The final equation reads, without any parameter beyond the rate coefficients:

$$\frac{dx_i}{d\tau} = \sum_{\alpha,\beta} r_{\alpha,\beta} x_\alpha x_\beta + \sum_{\alpha,\beta} q_\alpha x_\alpha$$

$r_{\alpha,\beta}$ and q_α may still depend on the gas temperature T . If $F_{\text{St_Eq}} = 1$, T is constant and these rates are computed only once and saved for the whole run. If thermal balance is computed, they are upgraded at each step.

Parameter `frnH` in `Base_Param.dat` can be used to vary ζ and n_{H} while keeping a constant ratio I_{ep} :

- If `frnH` < 0 : do not use, do nothing.
- Else set:

```
densh = densh * frnH
zeta = zeta * frnH
```

Hence, the ratio is preserved, but n_{H} and ζ are scaled by the same value.

Depletions

Among the control parameters, depletions lead to some specific problems.

- It is not possible to explore all of them separately. It would be too long and costly.
- Keeping them at reference values interferes with the setting of initial conditions.
- Some combination as, e.g. the C/O ratio, may be of special interest, but there are many ways to vary such a ratio.

If only a single depletion value is to be changed, this can be adjusted directly in the file `Base_Deple.dat`.

9.1 Initial conditions

As shown in Roueff & Le Bourlot (2020) the choice of initial values abundances may have a profound impact on the early evolution of the system. Furthermore the “standard” choice in the literature (i.e., all H on H₂ and all other atoms on the atomic species, either neutral or singly ionized) is very unfortunate.

For each atom X, we need to satisfy two different constrains:

- Keep its depletion δ_X at a fixed value.
- Allow the user to vary the initial abundances of all species containing “some” X.

This is rather tricky since one species may contain more than one X atom, and also other atoms Y, Z... The general optimization problem is over-constrained, and there is no straightforward best solution. The following algorithm is used:

- From the file `Base_Speci.dat`, compute the total amount of X in the initial conditions.
- Compute the difference with the required depletion as computed from $\delta_X n_H$, where δ_X comes from `Base_deple.dat`.
- Add or subtract that difference to a specific species containing X and that should be the reservoir of X in standard dark cloud conditions.

It is expected that the impact on the solution will thus be minimal. Careful tuning by the user may be required in some specific cases. If this algorithm fails, then the program stops immediately and a message is written on the screen.

9.2 C and O depletions

There are at least two different ways to study the impact of C and O depletions.

- Changing the total abundance of C and O.
- Changing the ratio C/O while keeping a constant metallicity.

Some modifications to the initialization routines have been made to facilitate this exploration. Keeping the total C and O constant while varying C/O implies a small computation. Let us write:

$$\delta_C + \delta_O = p_1$$

$$\frac{\delta_C}{\delta_O} = p_2$$

So:

$$\delta_O = \frac{p_1}{1 + p_2} \quad (9.1)$$

$$\delta_C = \frac{p_2 p_1}{1 + p_2} \quad (9.2)$$

Parameters `poc1`, `poc2`, in `Base_Param.dat` can be used to change the depletions.

- If `poc2 = -1` : do not use, do nothing.
- Else use Eq (9.1) and (9.2).

Maximum interpolation

Given 3 points (t_i, y_i) $i \in [1:3]$ such that $y_2 > y_1$ and $y_2 > y_3$, find the point (t, y) that maximize the fitted parabola. Let us define:

$$L_1(t) = \frac{(t - t_2)(t - t_3)}{(t_1 - t_2)(t_1 - t_3)}$$

$$L_2(t) = \frac{(t - t_1)(t - t_3)}{(t_2 - t_1)(t_2 - t_3)}$$

$$L_3(t) = \frac{(t - t_1)(t - t_2)}{(t_3 - t_1)(t_3 - t_2)}$$

And:

$$P(t) = y_1 L_1(t) + y_2 L_2(t) + y_3 L_3(t)$$

We see, by inspection, that this is our required quadratic function. Its derivative is:

$$P'(t) = y_1 L_1'(t) + y_2 L_2'(t) + y_3 L_3'(t)$$

with:

$$L_1'(t) = \frac{2t - t_2 - t_3}{(t_1 - t_2)(t_1 - t_3)}$$

$$L_2'(t) = \frac{2t - t_1 - t_3}{(t_2 - t_1)(t_2 - t_3)}$$

$$L_3'(t) = \frac{2t - t_1 - t_2}{(t_3 - t_1)(t_3 - t_2)}$$

So, $P'(t) = 0$ leads to:

$$y_1 (2t - t_2 - t_3)(t_2 - t_3) - y_2 (2t - t_1 - t_3)(t_1 - t_3) + y_3 (2t - t_1 - t_2)(t_1 - t_2) = 0$$

$$t = \frac{y_1 (t_2^2 - t_3^2) - y_2 (t_1^2 - t_3^2) + y_3 (t_1^2 - t_2^2)}{2 (y_1 (t_2 - t_3) - y_2 (t_1 - t_3) + y_3 (t_1 - t_2))}$$

Bibliography

Dislaire, V., Hily-Blant, P., Faure, A., Maret, S., Bacmann, A., Pineau des Forêts, G., 2012, “Nitrogen hydrides and the H₂ ortho-to-para ratio in dark clouds”, A&A, 537, A20

Roueff, E., Le Bourlot, J., 2020, “Sustained oscillations in interstellar chemistry models”, A&A, 643, A121