# Manual for c3.4 Head File "physics_coeff.h"

In short:

| | Related option |
|---|---|
| Number of levels | `LEVEL_N` |
| LVG Model | `OneD, TwoD, Isotropic, Mix` |
| C/A ratio, density of the collisional partner | `NC` |
| Viewing Angle | `OBS_ANG` |
| Temperature | `TEMP_SELE` |
| Molecule type | `MOLE_DATA` |
| Number of calculation points | `TAU_N` |

Detail:

1. Output File Name Options

```
#define OUTPUT_VER "c3.4"
#define OUTPUT_FILE_TAG "co"
#define OUTPUT_TYPE ".csv"
```

The output file will be in the form of

```
OUTPUT_VER[LEVEL_N][Dim][NC NC][OBS_ANG PI][T K]OUTPUT_FILE_TAG+OUTPUT_TYPE
```

For example, if `OUTPUT_VER = "c3.4", LEVEL_N = 5, TwoD = 1, NC = 2.1827273E+03, OBS_ANG = 0.5*M_PI, T = 30, OUTPUT_FILE_TAG = "co"` and `OUTPUT_TYPE = ".csv"`, then the file name will be:

```
c3.4[5][2D][NC 2.1827E3][0.5PI][30K]co.csv
```

2. GNUPlot file Output Option

```
#define GNUPLOT_OUTPUT 1
```

It decides whether to generate GNUPlot file and data or not. (0: No, 1: Yes)

The GNUPlot data file name will be in the form of

```
OUTPUT_VER[LEVEL_N][Dim][NC NC][OBS_ANG PI][T K]OUTPUT_FILE_TAG.dat
```

The GNUPlot command file name will be in the form of

```
OUTPUT_VER[LEVEL_N][Dim][NC NC][OBS_ANG PI][T K]OUTPUT_FILE_TAG.dem
```

3. Molecule data file from LAMDA

```
#define MOLE_DATA "co.win.dat"
```

This is the file name of the molecule data file which contains the Einstein coefficients, collisional coefficients and energy level information.

For SiO, use `"sio.win.dat"`.

The molecule data file should be located at the same directory as the program when you execute it.

4. Density of Molecule

```
#define Nt ((LEVEL_N+1)*LEVEL_N)/2
```

The density or the number of molecules in this cell. This option will not change the results of the program. You can also use `1.0` instead.

```
#define NC 2.1827273E+03
```

This is the density of the collisional partner (in unit of cm^-3). It decides the collisional coefficients and also the C to A ratio.

5. Velocity Gradient Model Options

```
#define OneD 0
#define TwoD 1
#define Isotropic 0
#define Mix 0                    //Mix of OneD and TwoD:
```

```
#define MixRatio 0.1                //MixRatio*sin^2 + cos^2
```

It decides the LVG Model. 0 for disable, 1 for enable. Choose only one model at a time.

`OneD` stands for one-dimentional velocity gradient.

`TwoD` stands for two-dimentional velocity gradient.

`Mix` stands for the mix model of one and two-dimentional velocity gradient, and the `MixRatio` decides the ratio.

`Isotropic` stands for isortopic case.

6. Viewing Angle Options

```
#if OneD     ← Do not change this.
    #define OBS_ANG 0.49167*M_PI
    #define TAU_ANG OBS_ANG
    #define TAU_START 0.001
#else
    #define OBS_ANG 0.5*M_PI
    #define TAU_ANG OBS_ANG
    #define TAU_START 0.01
#endif
```

`OBS_ANG` decides the angle between the line of sight and the z axis.

`TAU_ANG` decides the angle between the z axis and the line which be used as the unit of TAU. It usually is set to `OBS_ANG`.

`TAU_START` decides the starting TAU of the calculation.

The options between `#if OneD` and `#else` are for one-dimentional case. The options between `#else` and `#endif` are for other cases.

7. TAU Region Options

```
#define TAU_N 151                   //number of points in the curve
#define TAU_INC_RATIO 1.09647819614318 //optical depth increase ratio
```

They decide the points of optical depth which the calculation will be carried out at.

8. Number of levels

```
#define LEVEL_N 5
#define TOTAL_N ((LEVEL_N+1)*LEVEL_N)/2
#define TRANS_N ((LEVEL_N-1)*LEVEL_N)/2
```

`LEVEL_N` decides the number of levels. For example, if `LEVEL_N = 5`, then levels J = 0 ~ 4 will be included in calculation.

Do not change `TOTAL_N` and `TRANS_N`.

9. Temperature Options

```
#define TEMP_SELE 4
#define TEMP_B 2.725
```

`TEMP_SELE` decides which temperature option should be chose as the temperature of the cloud from the molecule data file.

`TEMP_B` is the cosmic blackbody radiation temperature (K).

10. Constant

```
#define h_CONST 6.62606896E-34      //Plank constant
#define LIGHT_SPEED 299792458.0     //Speed of light
#define k_CONST 1.3806504E-23       //Boltzmann constant
```

It takes the value from Wikipedia.

11. Convergent Options

```
#define REL_PREC 1E-6
#define EpsRel 1e-4
```

```
#define EpsAbs 0
```

`REL_PREC` is the cirterion for iteration subroutine to decide whether the value of n[] converges or not in each main loop.

`EpsRel` and `EpsAbs` decide the precision of integration for GSL integration subroutine.

12. Other Options

```
#define SLOW_MODE 0
#define Gsl_Integ_Space 3000
```

When `SLOW_MODE` is set to 1, the program will use thermal equilibrium population(n[]) as the initial value in each main loop. Usually, it will not affect the results significantly.

`Gsl_Integ_Space` decides the money space for GSL integration subroutine.

13. Debug Use Options

```
#define SHOW_A_V 0    //Show the A[] and v[] that read from MOLE_DATA
#define SHOW_C 0                  //Show the C[] that read from MOLE_DATA
#define SHOW_NI 0                 //Show the initial n[] that calculated by n_initial_cal()
#define SHOW_NF 0                 //Show the finial n[] that calculated by n_initial_cal()
#define OUTPUT_A_MATRIX_I 0         //Output the a_matrix_i[] that calculated by
a_matrix_initialize()
                              //and write it to the file A_MATRIX_I_FILE
#define A_MATRIX_I_FILE "a_matrix_i[].csv" //the output file of a_matrix_i[]
#define OUTPUT_E 0
#define E_FILE "E[].csv"
#define OUTPUT_BR_N 0
#define SHOW_BR_N 0
#define BR_N_FILE "Br_n[].csv"
```

These will not affect the results of calculation.

0: disable, 1: enable