This document gives simple examples about input and output of this idl code.

(1) **Temperature and density history profile**
A sample of temperature and density history profile named 'te_ne_history.dat' is in the folder '.../example' (also see the relative figure 'te_ne_history.eps').

Generally, the data structure in this file is simple:

| Item | Type | Count | Notes |
|---|---|---|---|
| 0 | Integer | 1 | N_lines: define how many streamlines or 'history' to be calculated in this file. |
| 1 | Integer | 1 | N_time: define the number of sampling time points along this streamline or 'history'. |
| | Double, 1D array | 1 | Time array (unit: s): This is a one dimensional array, which has N_time elements to save time nodes along this streamline. |
| | Double, 1D array | 1 | Temperature array (unit: K): This is a one dimensional array, which has N_time elements to save temperature. |
| | Double, 1D array | 1 | Electron number density array (unit: cm^-3): This is a one dimensional array, which has N_time elements to save number density. |
| 2 | Integer | 1 | Define the 2nd streamline as above. |
| | Double, 1D array | 1 | |
| | Double, 1D array | 1 | |
| | Double, 1D array | 1 | |
| 3 | Integer | 1 | Define the 3rd streamline. |
| | Double, 1D array | 1 | |
| | Double, 1D array | 1 | |
| | Double, 1D array | 1 | |
| … | | | |
| N_lines | Integer | 1 | Define the N_lines streamline |
| | Double, 1D array | 1 | |
| | Double, 1D array | 1 | |
| | Double, | 1 | |

| | | | |
|---|---|---|---|
| | 1D array | | |

For example, if we have 3 streamlines needed to be calculated, the N_lines then equals 3. Assuming that these lines include 100, 200, 400 sampling points, the N_time then is equals to 100, 200, and 400 respectively. Therefore, the 'te_ne_history.dat' can be created:

```
! ...
! open file as unit = 11
! ...
write(11,*) n_lines
do iline = 1, nlines
! ...
! Set n_time = 100, 200, & 400 for each iline, and define time_array_1d, te_array_1d,
ne_array_1d
! ...
write(11,*) n_time
write(11,*) time_array_1d(1: n_time)
write(11,*) te_arrar_1d(1: ntime)
write(11,*) ne_array_1d(1: ntime)
enddo
! ...
! close file
! ...
```

In this example 'te_ne_history.dat' file, we have only one streamline. We can read this 'te_ne_history.dat' using a IDL command like this:

```
; initial variables
  np = 1l
  ntime = 1l
; open file
  openr,lun,'te_ne_history.dat',/get_lun
  readf,lun,np
; enter cycle of streamlines
  for ip = 0, np - 1 do begig
; read density and temperature history
    readf,lun,ntime
    time_array = dblarr(ntime)
    te_array   = dblarr(ntime)
    rho_array  = dblarr(ntime)
    readf,lun,te_array
    readf,lun,rho_array
    readf,lun,time_array
    ;
    ; plot profiles
    ;
    plot, time_array, te_array
    ; ...
  endfor
```

(2) **Output structure**

The results of time-dependent ionization calculation is saved as .sav format from this IDL version. It contains all charge states of this elements, for all streamlines at the final time. The structure is named as `conce_strc`, which includes non-equilibrium charge states (nei), equilibrium charge states(eqi), temperature and number density at the final time:

```
conce_strc.nei[0:natom,0:n_lines-1]
conce_strc.eqi[0:natom,0:n_lines-1]
conce_strc.te[0:n_lines-1]
conce_strc.rho[0:n_lines-1]
```

Here, natom is the atomic index of this ion, n_lines is the number of streamlines. For example, for Fe ions, natom = 26. Then *`conce_strc.nei[0, 0]`* is the fraction of neutral Fe for the first streamline at the final time, and *`conce_strc.nei[20, 0]`* is for Fe XXI of the first streamline at the final time.

(3) **A example**

This is a simple example to show how to run the this IDL code.

Step 1. Download all relative files from
https://github.com/ionizationcalc/time_dependent_fortran/archive/master.zip

Step 2. Unzip the downloaded package, for example, named as 'time_dependent_fortran-master.zip'.

Step 3. In the IDL command line window, compile all relative routines:

```
IDL>.compile func_elemt_iv.pro func_index_nearest.pro
func_solveionization_eigen.pro subr_timedepen_eigen.pro
```

Step 4. Define the absolute path of eigenvalue tables and the table size, for example:

```
IDL> path_table='/Users/ccai/time_dependent_fortran-
master/example/chianti_7_te501/'
IDL> ntablesize=501
```

Step 5. Define the temperature and density history profile file 'te_ne_history.dat':

```
IDL> path_tenedat='/Users/ccai/time_dependent_fortran-
master/example/te_ne_history.dat'
```

Step 6. Define the element to be calculated, for example, Fe:

```
IDL> natom=26
```

Step 7. Define the output filename:

```
IDL> output='output_test.sav'
```

Step 6. Perform calculation:

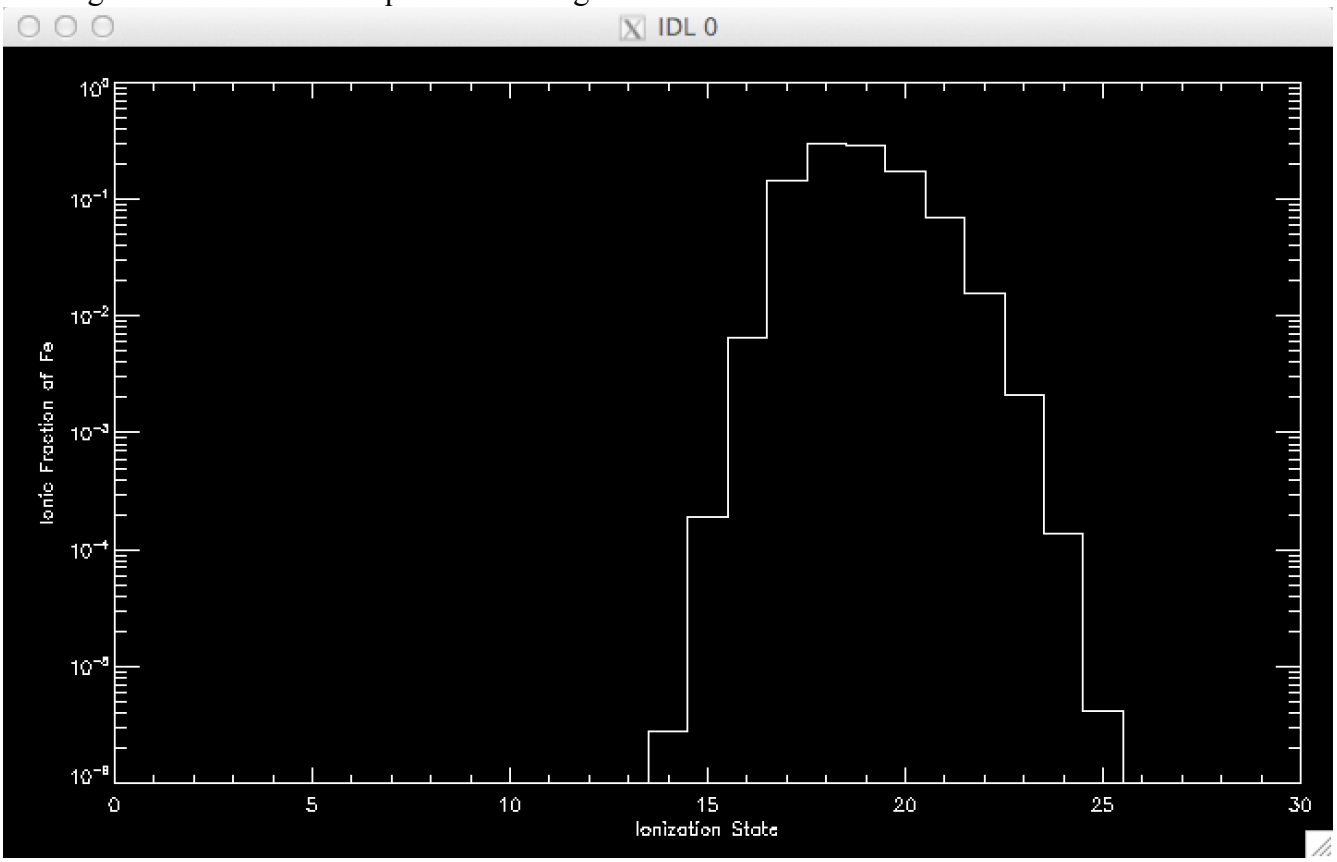
```
IDL>SUBR_TIMEDEPEN_EIGEN,natom,path_table,ntablesize,path_tenedat,output
```

After the running has been completed, a file named 'output_test.sav' is created in the current folder.

Step 7. Check results:

```
IDL>restore,output
IDL>plot,findgen(natom+1)+1,conce_strc.nei[0:natom,0],psym=10,xtitle='Charg
e States',Ytitle='Fraction of iron',/ylog,yyrange=[0.00001,1.0]
```

This figure then shows non-equilibrium charge states of Iron ions at the final time.




```
IDL> print,conce_strc.nei[0:natom,0]
  -9.9339721e-54   9.6601283e-47  -1.6268880e-40   3.3401230e-35   4.9335384e-30
1.3650018e-26  -2.5731748e-24   1.3231986e-20
   9.7935999e-18   3.3166272e-15   7.7899333e-13   1.2242919e-10   2.2856684e-08
2.7830120e-06   0.00019123954    0.0066309200
      0.14358261      0.30114803      0.28579279      0.17572161     0.069132278
0.015549316    0.0021078470   0.00013639478
   4.1630194e-06   8.5242231e-12   4.9847766e-19
```