****

****

SPTOOL for Windows
--- A Software Package for an Analysis of Stellar Spectra ---
************************ USERS MANUAL ***********************
************************ by Y. Takeda (NAOJ)***********************
****

[Still Incomplete Version]

0. Introduction

0.1 What is SPTOOL?

SPTOOL is a software package for analyzing high-dispersion stellar spectra (e.g., line-identification, determination of radial velocity, investigation of atmospheric parameters such as turbulent velocities or elemental abundances) which has been (and is being) developed by Y. Takeda while utilizing the program routines of the well-known ATLAS9/WIDTH9 codes written by Dr. R.L. Kurucz at CfA. It consists of the three component programs, SPSHOW, MPFIT, and WIDTH as briefly described below.

SPSHOW
This is an interactive spectrum browser such as SPLOT in IRAF, equipped with the function of synthetic spectrum simulation as well as that of input-data setup for the companion programs (MPFIT and WIDTH). Since it can visualize how the theoretical spectrum is affected by changing the parameters, a rough first-order analysis is possible while comparing the theory with observation. Most useful for the purpose of identifying lines. Measurement of equivalent widths is also possible.

MPFIT
Used for determining the elemental abundances or atmospheric parameters (e.g., turbulent or rotational velocities) based on the algorithm which automatically and iteratively establishes the solutions such that producing the best-fit theoretical spectrum with the observation. Though this may be a useful tool, it must not be regarded as being a fool-proof black-box. Whether it works efficiently depends heavily on how one formulates the problem adequately (e.g., choice of the parameters to be determined).

WIDTH
A program which determines the elemental abundance from an equivalent width, or inversely calculates the line intensity or profile for a given abundance. It was originally written by Dr. R. L. Kurucz (CfA) as a companion program of to the model atmosphere code known as ATLAS. The present version of WIDTH included in this SPTOOL, however, has been considerably modified by Y. Takeda equipped with various additional special features (e.g., application to blended multiplet features).

0.2.Machine Requirement

This software is developed for the most widely used PC operating system in the world, Microsoft Windows. (Unfortunately, other OS's such as Mac, unix, Linux etc. are totally outside the scope; and I do not have any plan at present to prepare SPTOOL for such other operating systems.) The recent versions of Windows, such as Windows 7 or Windows XP, are highly recommended, but it can run also Windows 95 through 2000. (It does not work on 16-bit OS such as Windows 3.1 or older.)
A free space of several hundred MB for the hard disk would be needed for installation and running of the programs. There may probably be no severe requirement for the speed of CPU clock or for the memory storage, as far as the recent PCs running MS Windows are concerned.

0.3 Developing Environment

All codes are written in Fortran, since this language is exclusively used in Kurucz's ATLAS9/WIDTH9 codes, on which this program package is based. nly one exeption is the special I/O interface program (cpanel.exe) for interactively viewing/editing the parameter file, which was developed with the help of Microsoft Visual Basic. For the purpose of reading and converting FITS-formatted files, I invoked FITSIO libraries ver. 5.03 written in FORTRAN 77 (cf. http://heasarc.gsfc.nasa.gov/docs/software/fitsio/fitsio_old_release.html). All source codes given in the ``srcfiles'' directory.

(1) Version for Windows XP and earlier (since 95)
DIGITAL/COMPAQ Visual Fortran 6.1 (Professional version, with IMSL library
[+Microsoft Visual Basic 5]

(2) Version for Windows Vista and 7
Intel Visual Fortran Composer XE2011 (Professional Version)
[+Microsoft Visual Basic 6]

0.4 How to Use This Manual

First of all, read carefully the instruction for installing SPTOOL described in section 1, and set it up to your PC. Then, it is recommended to jump to section 3 and experience running SPTOOL (SPSHOW/MPFIT/WIDTH).
After you have got a rough feeling about how it works, return to section 2 and learn the fundamental rules and features of SPSHOW in a systematic manner.

1. Installing SPTOOL

1.1 Download

SPTOOL is available from the following web site:

http://optik2.mtk.nao.ac.jp/~takeda/sptool/

Visit this site and download the zip-compressed file relevant for your environment, which includes all necessary programs and data.

[Windows XP , 2000, and earlier (since 95)]
sptool_xp2000.zip

[Windows Vista and Windows 7]
sptool_vista7.zip

Since this file is zip-compressed, some other utility program (e.g., winzip) is needed to uncompress it.

1.2 Setup

You must first determine where to place the SPTOOL directory, in which all subdirectories (containing programs, data, documents, etc.) are located.
I recommend to set it up in the root directory of the standard "C:" drive as I usually do, unless you have any special reason for not doing so;
in this case c:¥sptool will be the SPTOOL directory.
I assume that you would follow this choice in the following explanation.
(Note, however, that the choice is completely arbitrary; for example, you can place the sptool directory in "c:¥Program Files" or "d:¥work", instead of the recommended "c:¥". In these cases, the SPTOOL directory will be c:¥Program Files¥sptool and d:¥work¥sptool, respectively; and the following explanations should be read by replacing the corresponding terms adequately.)

Installing SPTOOL is relatively easy, though some tasks should be done manually. The basic three steps are as follows:

(1) Expand the file sptool.zip at a chosen directory (c:¥), where the SPTOOL directory will be created (c:¥sptool).
(2) Configure the SPTOOL environment variables and make a path to the SPTOOL executable file directory (c:¥sptool¥exefiles) by setting the enviromental variable "PATH". This can be done either by changing the contents of the "autoexec.bat" file or directly configure the variables via the "control panel" tool, depending on the version of Windows you are using. (It may be necessary to reboot PC for making the changes to environment variables effective.)

(3) Customize the configuration file "window.ini" so that the window size may become most effective for your display resolution.

I describe below these three procedure steps more in detail.

1.3 Typical Example

I concentrate here on the most typical case of installing SPTOOL to the root directory of the ``c:'' drive. (The user is recommend to follow the same choice, if possible.)

[1] Expanding the SPTOOL files:
Uncompress (unzip) the file "sptool*.zip" you have downloaded at the root directory of the c drive, "c:¥". You will find that a new directory named "sptool" will be created here.

[Elementary tips for those are unfamiliar with windows command line]
You can do this as follows:
Open the command prompt window as
start -> programs -> accessories -> command prompt
and move to c:¥ by inputting from the command line
``cd c:¥ (Enter)'' [``(Enter)'' means pressing the ``Enter'' key].
Let's confirm that the relevant files surely exist here by
``dir (Enter)''. and then execute the "unzip" program as follows:
unzip sptool.zip (Enter)
Please note that Window's command-line commands are case-insensitive
(unlike unix-like OS); e.g., "DIR", "dir", and "Dir" work exactly the same.

[2] Configuration of environmental variables

It is important to configure the following two environment variables appropriately in order to run SPTOOL.
(i) SPTOOL (indicating the location of the sptool directory)
(ii) PATH (to connect a path to the sptool exectable programs directory so that program files with "*.exe" suffixes or batch files with "*.bat" suffixes may be callable or executable from any directory.

This can be done by using the tool prepared by Windows.
You should login as an administrator-equivalent user.
(1) start -> control panels -> performance and maintenance -> system
(2) Then click the "configuration" tab to show "system property" .
(3) Click the "environment variables" button.
Here, you must newly create the SPTOOL environment variable ("SPTOOL") and assign “sptool directory” to it, and modify the “PATH” variable to make a path to “sptool executable files directory”.
<Setting SPTOOL as c:¥sptool>
Click the "new" button at the "system environment variable" window, and set SPTOOL and

c:¥sptool as a name and value, respectively. (Note that Windows OS is case-insensitive; so lower or upper cases are irrelevant.) And click the "OK" button.

<add c:¥sptool¥exefiles to PATH>

Click the "edit" button at the "system environment variable" window, and edit the PATH variable (maybe already existing) such that appending c:¥sptool¥exefiles to the existing data followed by semicolon.

For example, if PATH variable already has a value of

C:¥%SystemRoot%¥system32

you should modify it as

C:¥%SystemRoot%¥system32;c:¥sptool¥exefiles

and click the "OK" button.

Finally, click the "OK" button of the environment variables window, and that of the system property window, in order to make all the applied changes effective.

I suggest you reboot your PC after you have finished all these procedures.

Note: If the procedure mentioned above does not work, an alternative way may be to edit the "autoexec.bat" file (especially for earlier version of Windows). Namely, you can modify (or newly creating if it does not yet exist) the "autoexec.bat" file which is located at the root directory of the c:drive (i.e., c:¥autoexec.bat), which is fairly simple. Open the c:¥autoexec.bat file (if it does not exist, you can newly create it!) by an appropriate editor, and append the following two lines to the existing contents.

```
set SPTOOL=c:¥sptool
set PATH=c:¥sptool¥exefiles;%PATH%
```

Save the file and reboot your PC. That's all.

[3] Customization of window.ini

Move to the %SPTOOL% directory by typing "cd c:¥sptool (Enter)". There you find the "window.ini" file. Open this file by an editor (e.g., by typing "notepad window.ini (Enter)") and change the values of myscreen.numxpixels and myscreen.numypixels in the first and second line (depending on your display). Some recommended examples are as follows.

```
 800x 600 display:   numxpixels --> 800, numypixels-->480
 (In this coarse resolution case, font sizes may have to be changed:
  fontstd--> 't''Arial''h08w04i'     fontide --> 't''Arial''h06w03' )
1024x 768 display:   numxpixels --> 1024, numypixels--> 630
1152x 864 display:   numxpixels --> 1152, numypixels--> 730
1280x 768 display:   numxpixels --> 1280, numypixels--> 630
1280x1024 display:   numxpixels --> 1280, numypixels--> 900
1366x 768 display:   numxpixels --> 1350, numypixels--> 630
```

1600x1200 display:	numxpixels --> 1600, numypixels-->1080
1920x1080 display:	numxpixels --> 1910, numypixels-->1940

In addition, default IFMOLE value (ifmole_def) may as well be changed to 1
if you are mainly engaged with analyses of low-temperature stars later
than the Sun.

Other parameters can be changed as you wish, though you should have a sufficient understanding of their meaning for doing do.

1.4 Check with the SET command

Now that the preparation has been finished, you should be able to run SPTOOL. Before going further, however, let us check whether everything has been correctly done. Execute ``set (Enter)'' in the command prompt window. If you can confirm that the value (c:¥sptool) of the variable SPTOOL is correctly assigned and that the SPTOOL exectables directory (c:¥sptool¥exefiles) is included in the PATH variable, that's OK.If it does not work, something must have been wrong in your configuration of environmental variables. You should return to section 1.3 again.

[Notes for the Windows 95/98 Users:]
You might possibly encounter with a troublesome message ``insufficient space for environmental variables ...'' when opening the MS-DOS window. In such a case the space allocated for these variables has to be increased. Open the ``config.sys'' file (existing in the root directory of the boot drive; i.e., usually c:¥), insert the following line at the end of the file ``shell=c:¥command.com /e:4096 /p'' and save it. Then, reboot the computer.

1.5 Directry Structure

I summarize here the structures of subdirectories in the sptool directory.
[(...) denotes the parent directory (``c:'' in the standard case)]

(...)¥sptool¥window.ini	(SPSHOW configuration file)
(...)¥sptool¥exefiles¥	(directory containing executable files)
(...)¥sptool¥srcfiles¥	(directory containing program source files)
(...)¥sptool¥k2models¥	(directory containing Kurucz's model atmospheres)
(...)¥sptool¥GF10¥	(directory containing Kurucz and Bell's gf values)
(...)¥sptool¥miscdata	(directory containing miscellaneous data)

2. Fundamentals of SPSHOW

2.1 What Should be Prepared?

Let's begin with the fundamentals of running SPSHOW, an intelligent spectrum browser

equipped with functions of spectrum-simulation as well as preparation of data files for MPFIT and WIDTH. In order to get this program started, the user should consider preparing the following data files in advance
[1] observed spectrum of a target star
[2] model atmosphere relevant to this star
[3] data of spectral lines (wavelength, gf value, etc).

In order to synthesize a theoretical spectrum based on [2] and [3], we should specify various elemental abundances[6,7,...] along with the microturbulent velocity [5]. In addition, we should make up our mind whether molecule formation is included (IFMOLE=1) or not (IFMOLE=0) with the parameter IFMOLE [4]. These three file names [1,2,3] and input parameters [4,5,6,7,...] are passed to SPSHOW via the text file ``cpanel.txt'' (i.e., control panel), which is automatically created in the current directory if not exists. However, the user has no need to be aware of this file explicitly, since a special editor/viewer program ``cpanel.exe'' is responsible for the interface between the user and the content of this file, which is called from SPSHOW if necessary. The use of this interface program is self-explanatory, all what is needed is to assign appropriate file names or to change the value of any parameter as one wishes, and to click the "go" button to proceed further.

Let us then go into more detailed description concerning the three kinds of data files ([1], [2], and [3]) and the parameters to input ([4],[5],[6], ...).

2.2 Data Files

2.2.1 Filename Expression

Any file name is allowed as long as it does not exceed 60 characters and no blank character is included within it. File names without path specification (e.g., tauuma_n.fit) are regarded as being the files in the currect directory (``current directory'' means the directory from which SPSHOW was started). Of course, they can be expressed either with full path (e.g., c:¥work1¥spectrum1.fit) or with relative path to the current directory (e.g., ..¥..¥work2¥spec2.dat).

2.2.2 Observed Spectrum

The stellar spectrum should be already calibrated with wavelength (expressed in unit of $¥rm¥AA$). It is also highly recommended that the spectrum be normalized with respect to an appropriate continuum level (e.g., by using the task ``continuum'' in IRAF). Maximum number of wavelength points to be handled at one time (i.e., total length in the case of 1D spectrum, or the spectrum length of each order in the case of echelle multi-spectra) is 9999, which should not be exceeded.
Two kinds of data format are supported, i.e., text format and FITS format (must be linearly spaced in wavelengths such as by using ``dispcor’’ task of IRAF). FITS files must have names which end with extensions ``*.fits'' or ``*.fit''. All spectrum files without these two extensions are regarded as being in the text format.

In the case of text files, each line corresponds to a data pair at each wavelength point, i.e., wavelength (in Angstrom) in the first field and intensity in the second field in the default configuration, but can be in free format because of being read as ``read(line,*) lambda, intensity''. Only those lines, in which the first appearing non-blank character is numeric ([0-9]), are regarded as being the data lines. Hence, any line where the first non-blank character is non-numeric is discarded, which can be used as a comment line. Consequently, the text-formatted spectrum created by ``wspectext'' task in IRAF can be used unchanged even with FITS headers (cf. the sample spectrum prepared for sample3 in subsection 4.2).

FITS files (``*.fits'' or ``*.fit'') are also supported as far as the wavelength points (in angstrom) are linearly spaced by using the keywords "CRVAL1" (starting wavelength), "CDELT1" (wavelength step). It may be probable, however, that some type of FITS files can not be read by SPSHOW. In such cases, the user is recommended to rewrite the spectrum out to a text file (e.g., by using the task ``wspectext'' in IRAF) and use it.

The specfication of wavelength range (which determines the selected line data in case of blank name for the gf data file) is done by appending [%%%%,&&&&] after the file name, then the spectral portion of %%%%<=wavelength(A)<=&&&&
is regarded as being the observationalspectrum.
For example,
moon.fits[4500,4600]
tauuma_n.fit[6100,6200]
starspec.txt[0,99999]
Of course, you can omit this specification of wavelength range; in this case, the maximum available wavelength range is automatically assigned. (In the case of multi-spectra fits file for echelle spectra, however, a dialog will appear asking to specify the wavelength range, since the covered range is generally large. If you want not show such a dialog while you still want to use the maxumum available range, specify the very small/large wavelength limits such as [0,99999].)

Meanwhile, the Doppler correction can be included by appending {****} at the head of the filename, where "****" is the correction (in km/s) to be applied to the current spectra.
For example,
{-0.5}moon.fits
{-10}tauuma_n.fit[6100,6150]
{+123.23}starspec.txt

[Additional note for multi-spectra file of echelle spectra]
In case that a FITS file has multi-order spectra (e.g., observation with
echelle spectrograph), the user can also specify the order number in such
a way as ``filename[*,order]''. Caution: in this case, the 2-Dimensional data should be configured in the order of [number of wavelength points, number of orders], which can be confirmed, e.g., by the command ``imheader'' in IRAF. If inversed, reverse the order, e.g., by the task ``imtranspose'' in IRAF. For example, suppose that we have a fits file ``echelle.fit'',

which has a configuration of echelle.fit[4096,29] (i.e., containing twenty-nine spectra of 4K length). Then, if one wishes to read the fifteenth spectrum in this file, he should specify ``echelle.fit[*,15]'' as the file name (cf. subsection 4.3).
Alternatively, multi-spectra file may be divided in advance into a number of sub-spectrum corresponding to each order as

```
imcopy echelle.fit[*,1:1] echelle_1.fit
imcopy echelle.fit[*,2:2] echelle_2.fit
imcopy echelle.fit[*,3:3] echelle_3.fit
......
```

Then, each of the ``echelle_n.fits'' files can be used just in the same way as a usual single spectrum.

2.2.3 Model Atmosphere

The data of a model atmosphere (only one model in a file) should follow the ATLAS format (cf. Kurucz 1970) such that ending with the line ``BEGIN''. An easy way to prepare a necessary model is to invoke the data already computed by Kurucz (e.g., Kurucz CD-ROM No.13), from which one can arbitralily select or interpolate a desired model. Of course, one can run ATLAS program to produce the converged model with specified parameters, which is most accurate and recommendable.

Any file name is allowed as far as the condition described in subsubsection 2.1.1 is fulfilled. However, when a 17-character name in a special format ``t?????g&&&x#$$$m%'' is assigned, SPSHOW searches the current directory for it, and if it does not exist, a new model is created by interpolating the model data (taken from Kurucz CD-ROM No.13) in MOD_DIR directory. In this case, parameters are read from the file name as follows:

```
t????? $T_{¥rm eff}$=????? K
g&&&     $¥log g$ = &.&&
x#$$$   [M/H] = +$.$$ (if # is ``+'')
                  = -$.$$ (if # is ``-'')
      ([M/H] should be between -4 and 1)
m%        molecule formation is taken into account (if % is ``1'') or not
          (if % is ``0'') in calculating the number population when models
          are interpolated. (This parameter does not have to be necessarily
          compatible with IFMOLE described in subsubsection 2.2.5.)
```

For example,

```
t24390g479x+035m0 $T_{¥rm eff}$ = 24390 K, $¥log g$ = 4.79, [M/H]=+0.35, no molecule
t06570g329x-253m0 $T_{¥rm eff}$ = ~6570 K, $¥log g$ = 3.29, [M/H]=-2.53, no molecule
t04370g232x-330m1 $T_{¥rm eff}$ = ~4370 K, $¥log g$ = 2.32, [M/H]=-3.30, with molecule
```

The interpolation is done based on the following steps.
(a) Select 8 (= 2^{3}) models from the models in the k2models (MOD_DIR) directory

which adequately encompass the specified three parameters ($T_{¥rm eff}$, $¥log g$, [M/H]).
(b) Each of these 8 models are interpolated with respect to depth to yield a new set of 8 models, so that they may share the common {$¥tau_{5000}$} mesh.
(c) The final model corresponding to the specified parameters is obtained from this new set by 3-dimentional interpolation in the parameter space.

The fixed mesh {$¥tau_{5000}$} referenced in this process is specified in the ``taumesh.500'' file in the k2models (MOD_DIR) directory. The number of mesh points (n) is given in the first line, while the $¥log¥tau_{5000}$ values at each depth point are presented in the following n lines (all in free format). In the default case, this mesh is made of 40 depth-points extending from -6.0 to +1.8
with a step of 0.2. However, this mesh can be arbitralily customized by editing this ``taumesh.500'' file as one may wish.

2.2.4 Line Data

There are 7 kinds of data for a spectral line, which need to be passed to SPSHOW via the line-data file (seven columns in each one line; free format)

CODE: Numerical code represents the species of atom/molecule (see appendix A)
$¥lambda$: Air wavelength expressed in $¥rm¥AA$
$¥chi$: Excitation potential of the lower level expressed in eV
$¥log gf$: logarithm of g(statistical weight of the lower level) times f (oscillator strength)
GAMMAR: logarithm of the radiation(natural) damping constant (log10(gam_rad))
GAMMAS: logarithm of the electron Stark damping constant (per electron density at T=10000 K); log10(gam_e/N_e)
GAMMAW: logarithm of the van der Waals damping constant (per neutral hydrogen density at T=10000 K); log10(gam_H/N_H)
It is requisite for the first four to specify actual values, while for the last three default values are automatically assigned when assuming zero values.
(see also section 10.4).
The name of the file is arbitrary. However, just as in the case of model atmospheres, if a filename following the format of $$$$%%%%.kgf ($$$$ and %%%% are integers corresponding to W1 and W2 Angstroms; for example, 35704020.kgf or 65007500.kgf) is specified, the corresponding file comprising the line data between wavelengths of W1 and W2 is automatically created. In case a wavelength region longer than 10000 Angstrom is concerned, you can specify the filename as $$$$$%%%%%.kgf (e.g., 0950010400.kgf, 1230013800.kgf).

2.2.5 Molecule Control Switch

The user has to tell SPSHOW whether he wants to take into account molecule formation in calculating the number population of relevant species. This is controled by the parameter

IFMOLE, which is allowed to have a value of either 1 (molecules included) or 0 (molecules neglected).

Of course, if you are trying to simulate spectra of molecular lines seen in late-type (G-K) stars, there is no other choice than setting IFMOLE=1. Note also that molecule formation should be taken into consideration, when atomic lines of C or O are concerned, since significant fractions of C or O are consumed to form CO molecules in the atmosphere of late-type stars.

On the other side, in the case of high-temperature eary-type stars which show little or no molecular lines, it is quite safe and natural to set IFMOLE=0.

So, recommended choices are as follows:

A or earlier :	IFMOLE=0
F or later :	IFMOLE=1

2.2.6 Treatment of Blank Names

There are three kinds of data files whose names are to be passed to SPSHOW; i.e., stellar spectrum, atmospheric model, and spectral-line data. However, some of them may be left blank without being specified, which results in three different modes depending on the cases.

star sp.	model atm.	line data	mode
specified	specified	specified	obs vs. theory comparison mode
specified	blank	blank	browsing mode (obs. sp. only)
blank	specified	specified	simulation mode (theor. sp. only)
specified	specified	blank	obs vs. theory comparison mode (line data automatically generated)

Only these four combinations are accepted (otherwise, input dialog box will appear again with the cauton of ``check the file name!'').

Note that a file name beginning with ``!'' is regarded as being equivalent to blank. This is often useful for the purpose of quickly switching a filename into blank or vise versa.

2.3 How to Start SPSHOW

There are two different ways for starting SPSHOW.

2.3.1 Method 1

The easiest way is just to input
spshow (Enter)
without any argument.

In this case, the parameter file ``cpanel.txt'' is initialized (i.e., blank file names for all three, IFMOLE of 0 (or 1) and microturbulence of 2 km/s corresponding to the initial values specified in ``window.ini'', same elemental abundances as those of the model atmospheres) as shown by the viewer ``cpanel.exe''. What you can do next is to specify the necessary file names, to assign appropriate values to the parameters, and to click the ``go'' button. Then a synthesis calculation is started and the resulting spectrum will be displayed.

2.3.2 Method 2

Even if the parameter file ``cpanel.txt'' does not exist, SPSHOW automatically creates it. And this file will be updated when you change the contents on the viewer/editor ``cpanel.exe'' and click the ``go'' button. This file is left even after the execution of the program SPSHOW is terminated, while containing the information of the filenames and the parameters. Consequently, in case you already have this control parameter file as a template, you can run SPSHOW by taking the name of this file ``cpanel.txt'' as the argument:

```
   spshow cpanel.txt (Enter)
```

And the you can set the parameters as you wish on the "cpanel" window.
Of course, you can rename this cpanel.txt file as you wish to some other name (say, cpanel_test1.txt) to keep; then later you can start SPSHOW as

```
   spshow cpanel_test1.txt (Enter)
```

to redo the same simulation.
This has a merit of all information in the previous job is passed to the current session, since the contents of ``cpanel.txt'' are exactly copied already at the starting point.

3. Tutorial

"Practice Makes Perfect." Let's do some trainings of running SPTOOL. The actual sample data are contained in the file "testdata.zip" which can also be downloaded from the SPTOOL web site (http://optik2.mtk.nao.ac.jp/~takeda/sptool/). It is recommended that you unzip this file at the root directory of C: drive [you can do it by executing "unzip testdata.zip (Enter)") by which the directories will be recovered like this:

```
c:¥testdata¥working0
c:¥testdata¥working1
……
```

Each of these directories contain the sample data files corresponding to each of the exercises described below.

3.0 Warming up: How the model/line data are prepared

Let us move into c:¥testdata¥working0. This can be done as follows.
First, open the command-prompt window by proceeding as "start – all programs – accessories – command prompt." Then, type like this

```
cd   c:¥testdata¥working0   (Enter)
```

“(Enter)” means to press the Enter key. In the following desriptions, this (Enter) is often omitted for simplicity. Note also that commands are case-insensitive in Windows OS.

We first experience some commands of data preparation (which are automatically called from SPSHOW program when necessary) in order to familiarize using command lines.

[Preparation of model atmosphere data --- command “modint”]
Usage: modint *Teff* *logg* *[X]* *IFMOL*

Type as follows:

```
modint   5780   4.44   0.00   1   (Enter)
modint   5780   4.44   0.00   0   (Enter)
modint   13450   2.80   0.30   0   (Enter)
modint   4290   1.25   -1.97   1   (Enter)
modint   29500   4.62   -0.55   0   (Enter)
```

You wil find that the following five model-atmosphere files have been created.

```
t05780g444x+000m1
t05780g444x+000m1
t13450g280x+030m0
t04290g125x-197m0
t29500g462x-055m0
```

[Preparation of line data (Kurucz CD-ROM No.23) --- command “makekgf”]

Type as follows:

```
makekgf   14201460.kgf   (Enter)
makekgf   32003300.kgf   (Enter)
makekgf   53007100.kgf   (Enter)
makekgf   0932013300.kgf   (Enter)
makekgf   2320033100.kgf   (Enter)
```

The resulting files are self-explanatory (see Sect. 2.2.4).
(Caution: The number of the character string “$$$ …$$$” in $$$...$$$.kgf should be even; otherwise it does not work. For example, “932013300.kgf” is invalid.)

If the specified model/line files do not exist, SPSHOW will call “modint” or “makekgf” (from within) appropriately, in order to automatically create these necessary files.

[Preparation of solar flux spectrum (Kurucz CD-ROM No.18) --- command “makesfx”]

Finally, let’s learn how to prepare the spectrum data file of the solar flux spectra (Kurucz et al. 1984; Kurucz CD-ROM No.18) available between 3000A and 13000A.

Type as follows:

```
makesfx   39004000.sfx   (Enter)
makesfx   51005600.sfx   (Enter)
makesfx   77007800.sfx   (Enter)
makesfx   0910010900.sfx   (Enter)
makesfx   1200012800.sfx   (Enter)
```

For example, “39004000.sfx” contains the solar flux spectrum between 3900A and 4000A, while “0910010900.sfx” corresponds to the spectrum between 9100A and 10900A. (Caution: The number of the character string “$$$ …$$$” in $$$...$$$.sfx should be even; otherwise it does not work. For example, “910010900.sfx” is invalid.)

3.1 Typical Example of Analyzing Stellar Spectrum: Case of “tau UMa”

Let us move into c:¥testdata¥working1. Here you find a text file named “cpanel.set” and two spectrum files (tauuma_n.txt and tauma_n.fit), the contents of which are essentially the same (spectrum of the well-known metallic-line star, tau UMa, observed at Okayama Astrophysical Observatory) and the difference is only in the data format (simple text file on the one hand, and the fits-formatted file on the other).

3.1.1 Arranged run

As a first trial, please type as follows.

```
spshow   cpanel.set   (Enter)
```

Then, a light-green interface dialog box will appear. Click the “Go” button on it.
You will see messages of model-atmosphere creation and gf data creation, and eventually blue (observed) and red (simulated) spectra will appear on the screen overplotted.
Don’t you think both are well fit with each other?

You see detailed information of line-identification is shown below the spectra.
This information consists of (species, wavelength, and eta), where eta is
the line-center-to-continuum opacity ratio at tau=0.2.
According to Minnaert's formula, the observed line-depth (R) is approximately
written as 1/R = 1/R0 + 1/eta, where R0 is tha maximum line-depth (near to unity
in the blue region, while it reduced down to 0.5 or so in red or near-IR region).
So, this 'eta' can be an indicator of the line strength. Especially, it is noteworthy
that R becomes nearly similar to eta on the condition of eta <<1 .

However, you may consider that they are too crowded to be readily identified.
While it is possible to magnify (zoom-in) the relevant region by using 'z',
we would recommend the following ways in such a situation.
The first is changing the threshold of eta by using the 'e' command
(press 'e' and enter the value following the prompt).

Namely, since the threshold of eta (i.e., only the lines with eta values above this limit are shown) is set to 0.01 in the default setting, increasing this value (e.g., 0.1 or 1, for example) would naturally mitigate the crowdedness.
(By the way, the lower limit of threshold is 0.0001. You can not specifiy a value below this.)
The second is to display the detailed line information separately on the left-hand side of the screen by using the "L-click with SHIFT+CTRL" (or "R-click with SHIFT+CTRL") in the mouse mode (described below), which is most recommended.

Let us exercise here how to use some of the key commands, which help to control the display range of the spectrum.
Press "x" key, and following the prompt, give "6150 6170 (Enter)".
Press "w" key, and following the prompt, give "5 (Enter)".
Press "c" key, and following the prompt, give "6140 (Enter)".
Press ")" key several times, and press "(" key several times.
Press ">" key several times, and press "<" key several times.
Press "z" key several times, and press "Z" key (SHIFT+"z") several times.
Press "u" key several times, and press "U" key (SHIFT+"u") several times.
Also, try pressing different arrow keys, PgUp, and PgDn button, and see how it works.
You can display the usage of key functions by pressing “h” key.
Also, pressing “Enter” key makes the screen clear.

Generally speaking, when you have a prompt ending with "-->", you are requested to enter a numerical value (or numerical values separated by blanks).
Let us then try using the mouse. Just press '¥' (back-slash), then you receive a message that you have entered the mouse mode.
Point the cursor at any position and click with the left-button (keep pressing). The coordinate of the position (wavelength, intensity) will be indicated at the upper-left corner of the screen, and this information will disappear when you release the mouse button.

There are several functions in the mouse mode, which are used by clicking of the left/right button in combination with SHIFT and/or CTRL keys, as summarized below.

L-click	shows the position (wavelength, intensity) of the cursor
L-click with SHIFT	EW measurement (three clicks: left limit, right limit, continuum)
L-click with CTRL	x-range specification (such as 'x') (two clicks: left limit, right limit)
L-click with SHIFT+CTRL	display id-info (short format) (two clicks: left limit, right limit)
R-click with SHIFT	
R-click with CTRL	specifies mask range for MPFIT (two clicks: left limit, right limit)
R-click with SHIFT+CTRL	display id-info (long format) (two clicks: left limit, right limit)
R-click	Exit from the mouse mode, returning to the normal key mode

(L+R)-click Erase the screen
[First click the L button (keep pressing) and next click the R-button; then release both buttons]

L-drag in near-horizontal direction --> changes the x-range by dragging the viewport
L-drag in near-vertical direction --> magnification change in x-direction (same as 'z' or 'u')
R-drag with SHIFT in near-vertical direction --> magni. change in y-direction (same as 'Z' or 'U')

Please try by yourself how these mouse commands work.
You can exit the mouse mode and return to the normal key mode by clicking the right button.

OK now, for the time being, close the window by CTRL+c.

3.1.2 Step-by-step run

Next, let us begin from the scratch. Just type as follows:

spshow (Enter)

You will again see the light-green dialog box. Here, input “tauuma_n.txt’ (or “tauuma_n.fit”) and ‘t07500g400x+030m0’ for the names of the observed spectrum and the model atmosphere, respectively. And set IFMOL=0 by the toggle button. Assign 5 km/s for the value of the microturbulence. The, clich the “Go” button, which again displays blue and red spectra. In this case, however, we observe a discordance in the wavelength scale between the two spectra. Let us first shift the observed (red) spectrum horizontally so as to fit the theoretical (blue) spectrum. This can be done by using the “s” command (specifying the incremental horizontal shift in A). Try, for example assigning a shift of -0.05 A followed by pressing “s” And repeat it several times with different values in both directions. When you have accomplished the best match, you will see at the lower part of the screen ds = -0.25 and dVrad = -12.2, for example, which means that the total shift is about -0.25A and the corresponding radial velocity is -12.2 km/s. (Actual values you have obtained may be slightly different from these, but do not mind.) Nest, we adjust the width of the lines. Assign the Gaussian e-folding width (in km/s) followed by “b” command. After several trials, you will find that the best fit is observed at around 10 km/s. Now the red and blue spectra should match well each other on the screen. Then, press “G” (i.e., “g” while pressing Shift key), then the light-green dialog box again appears. Here please try assigning the name of the observed spectrum as “{-12.2}tauuma_n.txt” and then click “Go.” In this case, the wavelength scales are already adjusted from the outset, because the radial velocity correction was taken into account in the observed spectrum by appending {-12.2} to the filename.

3.1.3 Abundance derivation by automatic spectrum fitting with MPFIT

(Remark: In doing the example in this section, it is pre-requisite to well adjust the wavelength scale of the theoretical and observed spectra by applying an appropriate radial velocity shift, according to the manner described above.)

At this point, the observed stellar spectrum and the simulated theoretical spectrum should be over-plotted on the screen in red and blue lines, respectively.
While these two may be more or less match with each other, the fit may not be so good. In order to determine the abundances to a sufficient precision, however, we have to pursue a very good fit.
Let us determine the abundances of tau UMa for some representative elements by the technique of spectrum fitting with the help of MPFIT, a program designed to accomplish the best fit by an automatic-fitting algorithm.

As an example, we will do a synthetic spectrum fitting in the wavelength range of [6153A, 6160A]. So, after pressing 'x', enter "6153 6160".
Press '@', then a list will appear on the screen, which shows the top-ten elements in the descending order of eta values for the lines included in this wavelength range.
Here, you have to determine which elemental abundances should treated as variables to be adjusted. You should never consider "the many, the better". Too many variables would lead to an numerical instability and/or an excessively large computing time. As a rule of thumb, those elements with eta> 1 should be included, while those with eta <<1 had better be discarded. In the present case, the eta values are 0.5 for Na and 0.1 for Ca.
So, we decide to include down to Na, which makes the abundances of four elements (Si, Fe, O, Na) as free parameters to be determined. Therefore, in response to the prompt of "how many elements do you want to vary?", we reply as "4 (Enter)".
Then, since we are asked to input the atomic numbers of four elements, we enter as
"14 26 8 11 (Enter)" (where the order is irrelevant).
While it is prompted to input the value of the wavelength resolution (wavelength divided by the FWHM of the instrumental profile), this value is important only for the case of precise determination of the rotational velocity or macro-turbulence. Namely, you can assign an arbitrary value to this if you are interested only in abundance determinations, where its exact choice is almost irrelevant because macro-turbulence/rotation broadening and instrumental broadening are treated as being a "combined" macro-broadening with one parameter (fudge parameter introduced to accomplish the best fit without any real physical meaning). So, you can just press "Enter", then the default (parenthesized) value is set. Note here that the value of 0 corresponds to the case of infinite wavelength resolution; i.e., instrumental profile is delta-function. (In case where rotational broadening was adopted by assigning a negative vw in the ‘b’ command, it is additionally prompted to set the limb-darkening coefficient next to this inquiry of wavelength resolution, though this step is skipped in the present case of Gaussian broadening with a positive vw.)
The next step is which value should be adopted for the micro-turbulence. In default, the value used for calculating the currently displayed spectrum (in blue line) is assigned in parenthesis. If you want change it, input a new value here. Otherwise, just press “Enter” to adopt the default value.
Then, you are asked to enter the session name of the MPFIT job. Assign an appropriate name here, such as “test1” by typing “test1 (Enter)”. The iteration log or the solution of MPFIT is saved in a file named “sesson_name.sav” (e.g., in this example, “test1.sav”).

Finally, a chance is given is address some comment (within 60 characters) for a remark. If you do not have anything to write here, just press “Enter” key to skip.
Now, after several preparation programs have been executed, the screen is reset and cleared. We are now ready to run MPFIT.

Click the task bar and make the (so-far hidden) command prompt window active.
MPFIT is run by specifying the iteration number at the first argument (if this argument is omitted, only one iteration is assumed). Here, we try 3 iterations as a trial.
Type “mpfit 3 (Enter)”. You will see the fitted spectra three times displayed on the black background corresponding to each of the three iterations. When finished, you are asked if you wish to close the windows. For the present, however, keep all the windows open as they are by answering “no”. Since the window showing the results of calculations is hidden behind the graphic window, drag the task bar of the latter to make the former appear. Here are given the current solution, correction, and the new solution for the next iteration for each variable parameter. We can see the solutions are satisfactorily converging in the right direction. OK, then finish MPFIT by selecting file-exit. We next do three more iterations to accomplish the final convergence, which may be sufficient. Again type in the command prompt window as “mpfit 3 (Enter)”. Then a new message appears. Just type “y” here.
[The meaning of the inquiry is as follows. SPSHOW created the input data file named “minput.in” according to the “@” command containing the starting solutions. Using this file, MPFIT did three iterations and automatically create a new file “fornext.in”, which is similar to “minput.in” but the starting solutions have been replaced by the new updated solutions. If both “minput.in” and “fornext.in” exist in the current directory, MPFIT compares the time-stamps of both. And if “minput.in” is newer, it will be automatically adopted as the input file. If, however, “fornext.in” is newer, MPFIT asks whether “fornext.in” can be used as the starting solution the next iteration (case “y”); otherwise (case “n”) the original “mpfit.in” will be used (this is nothing but a repetition of the previous iterations)].
Three more iterations (4th to 6th) are performed, which result in a almost complete convergence. Let us close MPFIT windows by “file-exit”. The resulting solutions for each of the iteration steps (including the fitted spectrum) are written out in a file “test1.sav”. (You can examine the contents of this output file by yourself. One thing you should keep in mind in doing so is, however, that the wavelength scale is expressed in nanometer, not in angstrom.) In case you want make a data-file for plotting the fitted theoretical spectrum along with the observed spectrum, you can use the “readsav” program (the executable file is c:¥sptool¥exefiles¥readsav.exe) for this purpose. Just type “readsav test1.sav (Enter)” and follow the prompts. In any case, now that the MPFIT job has ended successfully, you can return to SPSHOW and continue your work. If you want to stop (and exit from) SPSHOW, press “q” (or CTRL+”c”). For deleting unnecessary intermediary files, you can just type “clean (Enter)” by using the batch file “c:¥sptool¥exefiles¥clean.bat”. (Or, “del *.tmp” or “del *.out” may be more safe and thus recommendable.)

Note: If your trial of MPFIT iterations turned out to be unsuccessful with a divergence of some parameters’ solutions, check first of all which elemental abundances have caused the problem. You can do this by examining the log file (e.g., “test1.sav” in the above case).
In such a case, fixing the relevant parameter (i.e., not regarded as a variable) may often be

a remedy, which can be done by editing the input data file of “minput.in”. For example, let us assume that the problematic element showing the divergence of solution is calcium (Ca). Open the “minput.in” file with an editor; then you should find a line “ 1 20 <Ca> ….”. Replace “1” in the 2nd column by “0” and save the file. Then, run MPFIT again. Probably, the job will go well this time. (As a matter of fact, this number in the 2nd column is the control parameter of whether this elemental abundance is varied or not. So, setting it to zero makes this parameter fixed unvaried.)

3.1.4 Equivalent widths measurements

Next, we explain how to measure the equivalent widths of spectral lines.
Generally speaking, the procedure consists of the following three consecutive steps:

(1) Enter the mouse mode by pressing the “¥” (back-slash) key.

(2) Left-button click while pressing the Shift key at the short-wavelength limit of the integration range.

(3) Left-button click while pressing the Shift key at the long-wavelength limit of the integration range.

(4) Left-button click while pressing the Shift key at the continuum level.

SPSHOW, having automatically returned to the normal key mode, then shows (at the left space on the screen) the list of lines relevant to this integration range, the equivalent-width value for the Gaussian fitting [ewma(g)], the equivalent-width value for the direct integration [ewma(d)], On the display, in the meanwhile, the fitted Gaussian function profile is drawn in light-blue, the specified position of the continuum level and the integration range are expressed by pink horizontal segments, and the original line profile points are shown by pink circles.

(5) As you see, you are prompted to specify the line number (in the list) to identify the line you are measuring. So, enter here an appropriate number selected from the displayed list. (However, the strongest line within the range is the relevant one in most cases, as already shown in the square bracket; if so, you can just press the “Enter” key).

(6) Then, SPSHOW asks you, which of ewma(g) and ewma(d) should be chosen as the finally adopted value. If you select the former, press “g” (and “Enter”), while you should enter “d” for the latter. Note that, since the former is chosen in a majority of cases, you can just press “Enter” to do the same function as “g”+”Enter”. Now, the measurement has been completed and the measured EW value is written out to appropriate files (the detailed logs of measurements are recorded in a file “ewcal.log”, and the input data used for the WIDTH program are stored in a temporary file named “linedat.tmp”).

As such, the EW measurement for a spectral line has been completed. The next measurement for another line may be carried out by repeating the processes (1) through (6).

(To be continued)