

He Triplet Modifier Reference

David P. Huenemoerder, John C. Houck, Michael A. Nowak, Norbert S. Shulz, John E. Davis

Contents

1	Function Reference	5
1.1	<code>he_modifier_init</code>	5
1.2	<code>triplet_line_em</code>	6
1.3	<code>create_he_modifier</code>	6
1.4	<code>W_dilution</code>	7

Chapter 1

Function Reference

1.1 `he_modifier_init`

Synopsis

Load He triplet modifier coefficient tables

Usage

```
he_modifier_init ([; <qualifiers>])
```

Description

This function initializes a number of internal data structures by reading tables of coefficients from FITS files on disk. If called with no arguments, the default coefficient tables are loaded.

Paths to specific data files may be specified using qualifiers `collex` for collisional excitation coefficients, and `photoex` for photoexcitation coefficients.

The coefficient tables currently included in the distribution are:

```
he_modifier_chianti.fits:
    Coefficients for triplet emissivities vs. density for a
    grid of temperatures, derived from Chianti.

he_modifier_apec.fits:
    Coefficients for triplet emissivities vs. density for a
    grid of temperatures, derived from a custom APEC run.

he_photoex_modifier_chianti.fits:
    Coefficients for triplet emissivities vs. photon density
    for a grid of temperatures, derived from Chianti.
```

Example

```
he_modifier_init ( ;collex="he_modifier_chianti.fits",
                  photoex="he_photoex_modifier_chianti.fits");
```

See Also

[1.3](#) (`create_he_modifier`), [1.2](#) (`triplet_line_em`)

1.2 triplet_line_em

Synopsis

Compute He triplet line emissivities for a given (Z,n,T)

Usage

```
Struct_Type = triplet_line_em (Z, ndens[], temp[] [; <qualifiers>])
```

Description

Compute the He triplet line emissivities for element, Z (an integer), number density, **ndens**, and Kelvin temperature, T. **ndens** and T may be one-dimensional arrays. The return value is a structure of the form

```
struct {Z, ndens, temp, w, x, y, z}
```

where Z is an integer, **ndens** and **temp** are arrays of length **nn** and **nT**, respectively. The struct fields **w**, **x**, **y** and **z** are arrays of dimension [**nn**,**nT**] containing the line emissivities.

If the **db_norm** qualifier is present, the triplet line emissivities will be normalized so that the total emissivity **w+x+y+z** matches the value from the currently loaded spectroscopy database rather than the value from the relevant coefficient table. Note that the total emissivity has little dependence on density.

If the **photoex** qualifier is present, the **ndens** parameter must be the photon energy density.

Example

```
% Compute Ne IX line emissivities on a 2D
% density-temperature grid:

ndens = 10^[8:15:0.1];
temp = 10^[6:8:0.05];
s = triplet_line_em (Ne, ndens, temp);
plot (log10(s.ndens), s.z[:,0]);
plot (log10(s.temp), s.z[0,:]);
```

See Also

[1.3](#) ([create_he_modifier](#)), [1.1](#) ([he_modifier_init](#))

1.3 create_he_modifier

Synopsis

Define a custom He triplet modifier function

Usage

```
create_he_modifier (name [, elements] [; photoex])
```

Description

Define a custom He triplet modifier function called **name**. The modifier function can then be used in conjunction with a custom spectral model function created using **create_aped_fun**.

The optional second argument can be used to limit the modifier function's effect to the elements given in a space- or comma-delimited string of element names. Otherwise, all elements found in the coefficient database will be affected.

If the `photoex` qualifier is present, the new modifier function will represent density dependence associated with photoexcitation. Otherwise the modifier function will represent pure collisional density dependence.

Example

```
% Include collisional density dependence for
% Ne, O, and Mg triplets only:

create_he_modifier ("He_a", "Ne,O,Mg");
create_aped_fun ("my_model", default_plasma_state);
fit_fun ("my_model(1, He_a(1)) * wabs(1)");

% Include photoexcitation density dependence for
% all He-like species:

create_he_modifier ("photo"; photoex);
create_aped_fun ("myphoto_model", default_plasma_state);
fit_fun ("my_photomodel(1, photo(1)) * wabs(1)");
```

See Also

?? (create_aped_fun), [1.1](#) (he_modifier_init), ?? (fit_fun)

1.4 W_dilution

Synopsis

Geometric dilution factor for a uniform disk

Usage

```
Double_Type[] = W_dilution (x[])
```

Description

Compute the geometric dilution factor for a mean intensity at position, $x=r/r_{\text{star}}$ from a uniform disk. This is used internally to dilute the radiation field, but may be useful for other purposes. The computed value is

$$W = (1 - \sqrt{1-x^2}) / 2$$