Function GET_EUV – single-thread version

Calling syntax:

```
res = call_external(libname, 'GET_EUV',                  $
                    Lparms, Rparms, Parms,               $
                    logTe_rsp, response,                 $
                    logTe_DEM, DEM_cor_arr, DEM_tr,      $
                    flux)
```

Function parameters:

0. Lparms – 4-element long integer array of dimensions (see below).
1. Rparms – 3-element double array of global real parameters (see below).
2. Parms – array of LOS parameters, 6 × Nz elements, double. Parms[*, i] represents the parameters for *i*th voxel (see below).
3. logTe_rsp – the temperature grid ($\log_{10}T$, where the temperature T is in K) of the instrumental response matrix, NT_rsp elements, double.
4. response – the instrumental response matrix, NT_rsp × Nchannels elements, double.
5. logTe_DEM – the temperature grid ($\log_{10}T$, where the temperature T is in K) of the DEM distribution(s), NT_DEM elements, double. This grid is assumed to be the same in all voxels, and the same for both the coronal and transition region DEMs.
6. DEM_cor_arr – array of coronal DEMs, NT_DEM × Nz elements, double, in cm^{-6} K^{-1}. DEM_arr[*, i] represents the DEM for *i*th voxel.
7. DEM_tr – the integrated DEM of the transition region, NT_DEM elements, double, in cm^{-5} K^{-1}. Note that DEM_tr may be omitted (set to 0); in such a case, contribution of the transition region is not computed.
8. flux – the output array of the computed EUV fluxes, 3 × Nchannels elements, double (see below).

Array of dimensions Lparms:

Lparms = [Nz, Nchannels, NT_rsp, NT_DEM]

0. Nz – number of voxels along LOS;
1. Nchannels – number of EUV channels;
2. NT_rsp – size of the temperature grid of the instrumental response matrix (i.e., the matrix is an NT_rsp × Nchannels array).
3. NT_DEM – size of the temperature grid of the DEM distribution(s).

Array of global real parameters Rparms:

Rparms = [dS_map, dS_rsp, TRfactor]

0. dS_map – visible source area, in arcsec^2.
1. dS_rsp – the default pixel area of the instrumental response matrix, in arcsec^2.

Note that the units of dS_map and dS_rsp can be arbitrary (but the same). Actually, the flux computed by convolving the DEM with the response matrix is then multiplied by the factor dS_map/dS_rsp, to obtain the actual flux corresponding to the chosen pixel size.

2. TRfactor – the factor applied to the contribution of the transition region, to account for the projection effects. The contribution of the transition region is computed if TRfactor > 0 and DEM_tr ≠ 0.

Array of parameters Parms (for a single voxel, 6 parameters):

0. Parms[0] = Δz – voxel length, in cm.
1. Parms[1] = T_0 – plasma temperature, in K (is used if DEM is not specified).
2. Parms[2] = n_0 – plasma density, in cm^{-3} (is used if DEM is not specified).
3. Parms[3] – DEM_on, the key specifying how the EUV emission is computed:
   a. DEM_on ≠ 0: the DEM distribution corresponding to this voxel is used;
   b. DEM_on = 0: DEM is not used; the emission is computed using T_0 and n_0.
4. Parms[4] – reserved.
5. Parms[5] – reserved.

Output array flux:

On output, this array contains the computed EUV fluxes. The units are determined by the used instrumental response matrix, usually DN s^{-1} pixel^{-1}. Each column of this array, flux[*, i], corresponds to *i*th spectral channel.

The first row of this array, flux[0, *], contains the coronal emission (without contribution of the transition region). The second row, flux[1, *], contains the emission from the transition region. The contribution of the transition region is only computed if TRfactor > 0 and DEM_tr ≠ 0; otherwise, flux[1, *] = 0.

The third row, flux[2, *], is always zero (this field is reserved for use by other functions).

Return value: currently, -1 if the input was incorrect (incorrect number of parameters); 0 otherwise.

Function GET_GX_EUV – single-thread version

Calling syntax:

```
res = call_external(libname, 'GET_GX_EUV',        $
                    Lparms, Rparms, Parms,        $
                    logTe_rsp, response,          $
                    Qrun, Lrun, logTe_DEM,        $
                    DEM_cor_run, DEM_tr_run,      $
                    flux)
```

Function parameters:

0. Lparms – 6-element long integer array of dimensions (see below).
1. Rparms – 2-element double array of global real parameters (see below).
2. Parms – array of LOS parameters, 10 × Nz elements, double. Parms[*, i] represents the parameters for *i*th voxel (see below).
3. logTe_rsp – the temperature grid ($\log_{10}T$, where the temperature T is in K) of the instrumental response matrix, NT_rsp elements, float.
4. response – the instrumental response matrix, NT_rsp × Nchannels elements, double.
5. Qrun – the EBTEL Q grid, NQ × NL elements, float, in erg cm^{-3} s^{-1}.
6. Lrun – the EBTEL L grid, NQ × NL elements, float, in cm.
7. logTe_DEM – the EBTEL temperature grid ($\log_{10}T$, where the temperature T is in K), NT_DEM elements, float.
8. DEM_cor_run – the EBTEL table for the corona, NT_DEM × NQ × NL elements, float, in cm^{-6} K^{-1}.
9. DEM_tr_run – the EBTEL table for the transition region, NT_DEM × NQ × NL elements, float, in cm^{-5} K^{-1}.
10. flux – the output array of the computed EUV fluxes, 3 × Nchannels elements, double (see below).

Array of dimensions Lparms:

Lparms = [Nz, Nchannels, NT_rsp, NQ, NL, NT_DEM]

0. Nz – number of voxels along LOS;
1. Nchannels – number of EUV channels;
2. NT_rsp – size of the temperature grid of the instrumental response matrix (i.e., the matrix is an NT_rsp × Nchannels array).
3. NQ – size of the EBTEL Q grid.
4. NL – size of the EBTEL L grid.
5. NT_DEM – size of the EBTEL temperature grid.

Array of global real parameters Rparms:

Rparms = [dS_map, dS_rsp]

0. dS_map – visible source area, in arcsec^2.
1. dS_rsp – the default pixel area of the instrumental response matrix, in arcsec^2.

Note that the units of dS_map and dS_rsp can be arbitrary (but the same). Actually, the flux computed by convolving the DEM with the response matrix is then multiplied by the factor dS_map/dS_rsp, to obtain the actual flux corresponding to the chosen pixel size.

Array of parameters Parms (for a single voxel, 10 parameters):

0. Parms[0] = Δz – voxel length, in cm.
1. Parms[1] = T_0 – plasma temperature, in K (is used if DEM is absent).
2. Parms[2] = n_0 – plasma density, in cm^{-3} (is used if DEM is absent).
3. Parms[3] – Voxel_ID, the key specifying the voxel type:
   a. (Voxel_ID and 4) ne 0 – the voxel belongs to the corona; the coronal emission is computed using Q, L, and the coronal EBTEL table.
   b. (Voxel_ID and 2) ne 0 – the voxel belongs to the transition region; the additional emission is computed using Q, L, and the transition region EBTEL table. If several voxels belong to the transition region, only one of them (the closest to the observer) contributes to the emission.

   Note that if (Voxel_ID and 4) eq 0, the Q and L parameters are ignored, and the coronal emission is computed using T_0 and n_0. If (Voxel_ID and 4) ne 0 or (Voxel_ID and 2) ne 0, and the Q or/and L values fall beyond the EBTEL table, then the coronal emission is computed using T_0 and n_0, and the transition region contribution is absent.
4. Parms[4] = Q – the EBTEL heating rate, in erg cm^{-3} s^{-1}.
5. Parms[5] = L – the EBTEL loop length, in cm.
6. Parms[6] = TRfactor – the factor applied to the contribution of the transition region, to account for the projection effects.
7. Parms[7] – reserved.
8. Parms[8] – reserved.
9. Parms[9] – reserved.

Output array flux:

On output, this array contains the computed EUV fluxes and some information about the line of sight. The flux units are determined by the used instrumental response matrix, usually DN s^{-1} $pixel^{-1}$. Each column of this array, flux[*, i], corresponds to *i*th spectral channel.

The first row of this array, flux[0, *], contains the coronal emission flux. The second row, flux[1, *], contains the emission flux from the transition region.

The third row, flux[2, *], contains the "transition region mask". This field is set to 1 (for all EUV channels), if ((Voxel_ID and 2) ne 0) and ((Voxel_ID and 8) ne 0) somewhere along the line of sight (in the same voxels that contribute to the emission), and 0 otherwise.

Return value: currently, -1 if the input was incorrect (incorrect number of parameters); 0 otherwise.

Function GET_EUV_SLICE – multi-thread version

Calling syntax:

```
res = call_external(libname, 'GET_EUV_SLICE',                    $
                    Lparms_M, Rparms_M, Parms_M,                 $
                    logTe_rsp, response,                         $
                    logTe_DEM, DEM_cor_arr_M, DEM_tr_M,          $
                    flux_M)
```

Function parameters:

0. Lparms_M – 5-element long integer array of dimensions. Lparms_M = [Npix, Nz, Nchannels, NT_rsp, NT_DEM], where Npix is the number of LOSs, and other elements are the same as in the single-thread version (they are assumed to be the same for all LOSs).
1. Rparms_M – array of real parameters common for all voxels within each LOS, 3 × Npix elements, double. Rparms_M[*, i] represents the parameter Rparms of the single-thread version for *i*th LOS.
2. Parms_M – array of voxel parameters, 6 × Nz × Npix elements, double. Parms_M[*, *, i] represents the parameter Parms of the single-thread version for *i*th LOS.
3. logTe_rsp is the same as in the single-thread version (this grid is assumed to be the same for all LOSs).
4. response is the same as in the single-thread version (the response matrix is assumed to be the same for all LOSs).
5. logTe_DEM is the same as in the single-thread version (this grid is assumed to be the same for all LOSs).
6. DEM_cor_arr_M – array of coronal DEMs, NT_DEM × Nz × Npix elements, double. DEM_cor_arr_M[*, *, i] represents the parameter DEM_cor_arr of the single-thread version for *i*th LOS.
7. DEM_tr_M – array of integrated DEMs of the transition region, NT_DEM × Npix elements, double. DEM_tr_M[*, *, i] represents the parameter DEM_tr of the single-thread version for *i*th LOS.
8. flux_M – the output array of the computed EUV fluxes, 3 × Nchannels × Npix elements, double. flux_M[*, *, i] represents the parameter flux of the single-thread version for *i*th LOS.

Return value: currently, -1 if the input was incorrect (incorrect number of parameters); 0 otherwise.

Function GET_GX_EUV_SLICE – multi-thread version

Calling syntax:

```
res = call_external(libname, 'GET_GX_EUV_SLICE',        $
                    Lparms_M, Rparms_M, Parms_M,        $
                    logTe_rsp, response,                $
                    Qrun, Lrun, logTe_DEM,              $
                    DEM_cor_run, DEM_tr_run,            $
                    flux_M)
```

Function parameters:

0. Lparms – 7-element long integer array of dimensions. Lparms_M = [Npix, Nz, Nchannels, NT_rsp, NQ, NL, NT_DEM], where Npix is the number of LOSs, and other elements are the same as in the single-thread version (they are assumed to be the same for all LOSs).
1. Rparms – array of real parameters common for all voxels within each LOS, 2 × Npix elements, double. Rparms_M[*, i] represents the parameter Rparms of the single-thread version for *i*th LOS.
2. Parms – array of voxel parameters, 10 × Nz × Npix elements, double. Parms_M[*, *, i] represents the parameter Parms of the single-thread version for *i*th LOS.
3. logTe_rsp is the same as in the single-thread version (this grid is assumed to be the same for all LOSs).
4. response is the same as in the single-thread version (the response matrix is assumed to be the same for all LOSs).
5. Qrun is the same as in the single-thread version (this grid is assumed to be the same for all LOSs).
6. Lrun is the same as in the single-thread version (this grid is assumed to be the same for all LOSs).
7. logTe_DEM is the same as in the single-thread version (this grid is assumed to be the same for all LOSs).
8. DEM_cor_run is the same as in the single-thread version (this table is assumed to be the same for all LOSs).
9. DEM_tr_run is the same as in the single-thread version (this table is assumed to be the same for all LOSs).
10. flux_M – the output array of the computed EUV fluxes, 3 × Nchannels × Npix elements, double. flux_M[*, *, i] represents the parameter flux of the single-thread version for *i*th LOS.

Return value: currently, -1 if the input was incorrect (incorrect number of parameters); 0 otherwise.