

```

1 function results_packet_weighting, output, out_of_shadow
2
3 common constants
4 common results
5
6 case (format.quantity) of
7   'column': weight = *output.frac * stuff.atoms_per_packet ;; atoms per packet
8   'density': weight = *output.frac * stuff.atoms_per_packet
9   'intensity': begin
10     if (max(strcmp(format.emission.mechanism, 'resscat', /fold))) then begin
11       ;; trim min and max vy_sun values
12       w = where(*output.radvel_sun LT min(*gvalue.v), nw)
13       if (nw NE 0) then (*output.radvel_sun)[w] = min(*gvalue.v)
14       w = where(*output.radvel_sun GT max(*gvalue.v), nw)
15       if (nw NE 0) then (*output.radvel_sun)[w] = max(*gvalue.v)
16
17       ;; sum g-value over observed lines
18       gg = 0.
19       for j=0,n_elements(format.emission.line)-1 do begin
20         w = (where(abs(*gvalue.wavelength-(format.emission.line)[j]) LE $
21           1e-2, nw))[0]
22         if (nw NE 1) then stop
23         gg += interpola(*gvalue.g)[*,w], *gvalue.v, $
24           *output.radvel_sun
25       endfor
26
27       ;; Compute emission measure for each packet
28       weight_resscat = (*output.frac*stuff.atoms_per_packet) * out_of_shadow * $
29         (gg/1e6) ;; Ra
30       ;; gg/1e6 = 10^6 photons/atom/sec
31       ;; *output.frac * atoms_per_packet = atoms
32       ;; f_resscat = 10^6 photons/sec
33     endif
34
35     ;; Compute electron impact emission
36     if (total(strcmp(format.emission.mechanism, 'eimp', /fold))) then stop
37
38     ;; Sum emission measures for each process
39     weight = weight_resscat ;; + weighth_eimp
40   end
41   'spectrum': stop
42   else: stop
43 endcase
44
45 q = where(finite(weight) EQ 0, nq) & if (nq NE 0) then stop
46
47 return, weight
48
49 end

```