```
function emission_measure, atom, line, vy=vy, aplanet=aplanet, ee=ee

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;
;; Computes the emission measure for each packet. This is used in
;; line_of_sight, model_images, and density_track.
;;
;; Required parameters:
;;    * atom
;;    * line = vector of lines to compute emission for in Å
;; Optional depending on the emission type and line
;;    * vy = radial velocity relative to the sun
;;    * aplanet = heliocentric planet distance. If not specified, then resonant scattering
;;        is not computed
;;
;; Outputs:
;;   Function returns the emission measure per atom for the requested lines
;;   ee = resonant scattering emission measure for each line
;;
;; Version 2.0: 19 April 2010
;;    * written based on already existing method in line_of_sight and model_images.
;;    * need a new version to make sure things are done consistently.
;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

nl = n_elements(line)
doresscat = (n_elements(aplanet) EQ 1)
doeimp = 0

;; Correct for Na wavelength issues
if (atom EQ 'Na') then begin
  q = where(line EQ 5890, nq)
  if (nq EQ 1) then line[q] = 5891.
  q = where(line EQ 5896, nq)
  if (nq EQ 1) then line[q] = 5897.
endif

;; Resonant Scattering
if (doresscat) then begin
  q = get_gvalue(aplanet, atom, lines=ll, velocity=radvel, gval=gval)
  w = where(vy LT min(radvel), nw) & if (nw NE 0) then vy[w] = min(radvel)
  w = where(vy GT max(radvel), nw) & if (nw NE 0) then vy[w] = max(radvel)

  ee = fltarr(n_elements(vy),nl)
  for i=0,nl-1 do begin
    q = where(ll EQ line[i], nq)
    if (nq NE 1) $
      then print, 'Error: g-value not found for emission line ' + string(line[i]) $
      else ee[*,i] = interpol(gval[*,q], radvel, vy)/1e6
  endfor
  resscat = (nl EQ 1) ? ee : total(ee,2)
```

```
endif else resscat = 0.

;; Electron Impact
if (doeimp) then begin
  stop
endif else eimp = 0.

result = resscat + eimp
return, result

end
```