```
pro combine_iterations, filename, filelist, delete=delete

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;
;; Combine the individual interations of a model run into a single file
;; Set keyword start if this is a startloc structure
;; Set keyword after if combining after all model runs are completed
;;
;; * filename is the name of the new savefile
;; * If filelist is given, it is an array with the names of files to combine
;; * If filelist is not given, then it looks for all the filename.#
;;
;; Version history:
;;   3.2: 4/27/2011
;;      * trackloss is now optional
;;   3.1: 1/6/2011
;;      * Changing the way it finds files
;;   3.0: 7/21/2010
;;      * rewriting for new structure architecture
;;   2.3: 19 Jan 2010
;;      * Combines the deposition and hitfrac, etc. fields.
;;   2.2: 20 November 2009
;;      * Saves the final structures with single precision floating point rather than
;;        double precision.
;;   2.0: Rewritten to conform with new structure definitions.
;;   1.0: originial
;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

if (delete EQ !null) then delete = 0

if (filelist EQ !null) then begin
  filelist = file_search(filename+'.*', count=numit)
  savenames = 0
endif else begin
  numit = n_elements(filelist)
  savenames = 1
endelse

case (numit) of
  0: stop
  1: spawn, 'cp ' + filelist + ' ' + filename
  else: begin
    packets = 0
    for it=0,numit-1 do begin
      restore, filelist[it]
      if (packets EQ 0) $
        then begin
          outnew = temporary(output)
          innew = temporary(input)
          vernew = temporary(version)
```

```
52          endif else begin
53            *outnew.x0 = float([*outnew.x0, *output.x0])
54            *outnew.y0 = float([*outnew.y0, *output.y0])
55            *outnew.z0 = float([*outnew.z0, *output.z0])
56            *outnew.f0 = float([*outnew.f0, *output.f0])
57
58            *outnew.vx0 = float([*outnew.vx0, *output.vx0])
59            *outnew.vy0 = float([*outnew.vy0, *output.vy0])
60            *outnew.vz0 = float([*outnew.vz0, *output.vz0])
61
62            *outnew.phi0 = float([*outnew.phi0, *output.phi0])
63            *outnew.time = float([*outnew.time, *output.time])
64
65            *outnew.x = float([*outnew.x, *output.x])
66            *outnew.y = float([*outnew.y, *output.y])
67            *outnew.z = float([*outnew.z, *output.z])
68            *outnew.frac = float([*outnew.frac, *output.frac])
69
70            *outnew.vx = float([*outnew.vx, *output.vx])
71            *outnew.vy = float([*outnew.vy, *output.vy])
72            *outnew.vz = float([*outnew.vz, *output.vz])
73
74            if (input.options.trackloss) then begin
75              *outnew.lossfrac = float([*outnew.lossfrac, *output.lossfrac])
76              *outnew.ringfrac = float([*outnew.ringfrac, *output.ringfrac])
77              *outnew.leftfrac = float([*outnew.leftfrac, *output.leftfrac])
78
79              s = size(*outnew.hitfrac)
80              if (s[0] EQ 1) $
81                then *outnew.hitfrac = float([*outnew.hitfrac, *output.hitfrac]) $
82                else begin
83                  temp = fltarr(n_elements(*outnew.x),s[2])
84                  temp[0:s[1]-1,*] = float(*outnew.hitfrac)
85                  temp[s[1]:*,*] = float(*output.hitfrac)
86                  *outnew.hitfrac = temp
87                endelse
88              *outnew.deposition.map += *output.deposition.map
89            endif
90
91            outnew.totalsource = total(double(*outnew.f0))
92
93            if (savenames) then *outnew.sourcefile = [*outnew.sourcefile, $
94              *output.sourcefile]
95
96            destroy_structure, output
97            destroy_structure, input
98          endelse
99        packets = n_elements(*outnew.x)
100     endfor
101
102     output = temporary(outnew)
```

```
103       input = temporary(innew)
104       version = temporary(vernew)
105
106       save, output, input, version, file=filename
107       destroy_structure, output
108       destroy_structure, input
109     end
110   endcase
111
112   make_model_header, filename
113
114   ;; Delete intermediate filelist
115   if (delete) then for i=0,numit-1 do spawn, 'rm ' + filelist[i]
116
117   end
118
```