

KEMPO1

Kyoto university ElectroMagnetic Particle cOde: 1d version

Yoshiharu Omura

*Research Institute for Sustainable Humanosphere, Kyoto University, Gokasho Uji, Kyoto
611-0011, Japan
omura@rish.kyoto-u.ac.jp*

```
1 %%README:
2 % The following code 'kempol.m' is programmed in the MATLAB programming language.
3 % The code needs to be installed in a specified directory of the MATLAB system.
4 % In the command window of the MATLAB system, type the following:
5 %     kempol parameter_file
6 % If the 'parameter_file' is omitted, the parameters in 'default.dat' are used.
7 % The parameter file can be generated by modifying the contents of 'default.dat'
8 % attached at the end of this code, and it should be saved in the same directory
9 % where 'kempol.m' is stored.
10 %
11 %*****
12 %
13 % KEMPO1
14 %     Kyoto university ElectroMagnetic Particle cOde: 1d version
15 %
16 %
17 % FORTRAN Version (Ver. 1)
18 %     developed by
19 %     Yoshiharu Omura and Hiroshi Matsumoto
20 %     Research Institute for Sustainable Humanosphere,
21 %     Kyoto University
22 %     Uji, Kyoto, 611-0011, Japan
23 %     E-mail: omurarish.kyoto-u.ac.jp
24 %     FAX: +81-774-31-8463
25 %
26 % MATLAB Version (Ver. 2)
27 %     developed for Lecture on "Electromagnetic Simulations"
28 %     by Yoshiharu Omura and Hideyuki Usui
29 %     Graduate Course of Electrical Engineering, Kyoto University
30 %
31 % MATLAB Version with User Interface and Graphic Diagnostics (Ver. 3)
32 %     developed by Koichi Shin and Yoshiharu Omura
33 %     for 7th International School for Space Simulations (ISSS-7)
34 %     March 26-31, 2005, Kyoto Japan
35 %     Supported by COE21/KAGI Program
36 %
37 %     Copyright(c) 1993-2005, Space Simulation Group,
38 %     RISH, Kyoto University, All rights reserved.
39 %
40 %     Version 3.2   March 30, 2005
41 %
42 %*****}
43 function kempolmain(input_filename)
44     clear global
45     warning off
46     global prm      % input parameters
47     global ren      % normalize factor
```

```

48 global q mass rho0
49 global slx % system length x
50 global nxp1 nxp2 % nx+1, nx+2
51 global X1 X2 X3 % 1:nx, 2:nxp1, 3:nxp3
52 global cs tcs % c^2, 2*c^2
53 % field
54 global ex ey ez
55 global bx by bz
56 global ajx ajy ajz
57 global rho
58 % particles
59 global vx vy vz
60 global x
61 % diagnostics
62 global ifdiag % interval for diagnostics
63 global eng % for energy plot
64 global field % field date for wk plot
65 global flag_exit
66 flag_exit = 0;
67
68 %-- read parameters --
69 if ~exist('input_filename')
70 input_filename = 'default.dat'; % default input filename
71 end
72 prm = input_param(input_filename);
73 if isempty(prm)
74 return
75 end
76
77 %-- initialize --
78 hdiag = diagnostics_init;
79 %--
80 [prm,ren] = renorm(prm);
81 initial(prm, hdiag);
82 x = position(x,vx);
83 if prm.iex
84 rho= charge(x);
85 ex = poisson(ex, rho);
86 end
87 %-----
88 % Main loop
89 %-----
90 jtime = 0;
91 jdiag = 1;
92 %-- diagnostics --
93 hdiag = diagnostics(hdiag, jtime, jdiag);
94 if prm.nplot == 0
95 return
96 end
97
98 %--
99 for jtime = 1:prm.ptime
100 %--
101 if prm.iex == 2
102 [ vx, vy, vz] = rvelocity(vx,vy,vz, ex,ey,ez, by,bz, x);
103 x = position(x,vx);
104 x = position(x,vx);
105 rho= charge(x);
106 ex = poisson(ex, rho);
107 else
108 [ by, bz] = bfield( by,bz, ey,ez);
109 [ vx, vy, vz] = rvelocity(vx,vy,vz, ex,ey,ez, by,bz, x);
110 x = position(x,vx);
111 [ajx,ajy,ajz] = current(ajx,ajy,ajz,vx,vy,vz,x,jtime);
112 [ by, bz] = bfield( by,bz, ey,ez);

```

```

113     [ ex, ey, ez ] = efield(ex,ey,ez, by,bz, ajx,ajy,ajz);
114     x = position(x,vx);
115     end
116     %-- diagnostics --
117     if mod(jtime,ifdiag)==0
118         jdiag = jdiag+1;
119         hdiag = diagnostics(hdiag, jtime, jdiag);
120     end
121     if flag_exit
122         break;
123     end
124     end
125     %-- diagnostics --
126     if ~flag_exit
127         diagnostics_last(hdiag, jtime);
128     end
129     return
130     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
131     function [by,bz] = bfield(by,bz, ey,ez)
132         global nxp1 nxp2
133         global X1 X2 X3
134
135         by(X2) = by(X2) +ez(X2) -ez(X1);
136         bz(X2) = bz(X2) -ey(X3) +ey(X2);
137         by(nxp2)= by(2);
138         bz(nxp2)= bz(2);
139         by(1)   = by(nxp1);
140         bz(1)   = bz(nxp1);
141     return
142     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
143     function rho = charge(x)
144         global prm
145         global nxp1 nxp2
146         global q
147         global rho0
148         rho = rho0;
149         n2 = 0;
150         for k=1:prm.ns
151             n1 = n2;
152             n2 = n1 + prm.np(k);
153             for m = (n1+1):n2
154                 i = floor(x(m)+ 2.0);
155                 i1 = i+1;
156                 s2 = (x(m)+ 2.0 - i)*q(k);
157                 s1 = q(k) - s2;
158                 rho(i ) = rho(i ) + s1;
159                 rho(i1) = rho(i1) + s2;
160             end
161         end
162
163         rho(2)   = rho(2) +rho(nxp2) -rho0(2);
164         rho(1)   = rho(nxp1);
165         rho(nxp2) = rho(2);
166
167     return
168     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
169     function [ajx, ajy, ajz] = current(ajx,ajy,ajz, vx,vy,vz, x, jtime)
170         global prm
171         global nxp1 nxp2
172         global X1 X2 X3
173         global q
174         ajx = zeros(nxp2,1);
175         ajy = zeros(nxp2,1);
176         ajz = zeros(nxp2,1);
177

```

```

178  %--
179  n2=0;
180  for k=1:prm.ns
181      n1 = n2;
182      n2 = n2 + prm.np(k);
183      qh = q(k)*0.5;
184      for m = (n1+1):n2
185          ih = floor( x(m) + 1.5 );
186          s2 = (x(m) + 1.5 - ih)*q(k);
187          s1 = q(k) - s2;
188          ih1= ih+1;
189          ajy(ih) = ajy(ih) + vy(m)*s1;
190          ajy(ih1) = ajy(ih1) + vy(m)*s2;
191          ajz(ih) = ajz(ih) + vz(m)*s1;
192          ajz(ih1) = ajz(ih1) + vz(m)*s2;
193          %-- charge conversion method --
194          if prm.iex
195              qhs = qh * sign(vx(m));
196              avx = abs(vx(m));
197              x1 = x(m) + 2.0 -avx;
198              x2 = x(m) + 2.0 +avx;
199              i1 = floor(x1);
200              i2 = floor(x2);
201              ajx(i1) = ajx(i1) + (i2 - x1)*qhs;
202              ajx(i2) = ajx(i2) + (x2 - i2)*qhs;
203          end
204      end
205  end
206  %-- boundary --
207  ajx(nxp1) = ajx(1) + ajx(nxp1);
208  ajx(2) = ajx(2) + ajx(nxp2);
209  ajy(nxp1) = ajy(1) + ajy(nxp1);
210  ajy(2) = ajy(2) + ajy(nxp2);
211  ajy(1) = ajy(nxp1);
212  ajz(nxp1) = ajz(1) + ajz(nxp1);
213  ajz(2) = ajz(2) + ajz(nxp2);
214  %--
215  i=nxp1:-1:2;
216  ajy(i) = (ajy(i) + ajy(i-1))*0.5;
217  %-- external current source ----
218  if prm.ajamp
219      ajz(prm.nx/2+1) = ajz(prm.nx/2+1)+prm.ajamp*sin(prm.wj*jtime);
220  else
221      %--cancellation of uniform Jx,Jy,Jz components---
222      ajxu = sum(ajx(2:nxp1))/prm.nx;
223      ajyu = sum(ajy(2:nxp1))/prm.nx;
224      ajzu = sum(ajz(2:nxp1))/prm.nx;
225      ajx(2:nxp1) = ajx(2:nxp1) -ajxu;
226      ajy(2:nxp1) = ajy(2:nxp1) -ajyu;
227      ajz(2:nxp1) = ajz(2:nxp1) -ajzu;
228  end
229
230  return
231  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
232  %
233  % diagnostics
234  %
235  % hdiag: handles
236  %
237  function hdiag = diagnostics(hdiag, jtime, jdiag)
238  global prm ren
239  global ex ey ez by bz vx vy vz
240  global bx0 by0
241  global eng
242  global field

```

```

243 global kspec
244 global X1 X2 X3
245
246 global flag_exit;
247 try
248     % pause check
249     flag = get(hdiag.fig,'UserData');
250     if strcmp(flag,'pause')
251         uiwait(hdiag.fig);
252     end
253     % exit chack
254     flag = get(hdiag.fig,'UserData');
255     if strcmp(flag,'exit')
256         flag_exit = 1;
257         return;
258     end
259
260 catch
261     return
262 end
263 %
264 if hdiag.flag_eng
265     eng(:,jdiag) = energy(ex,ey,ez,by,bz,vx,vy,vz);
266 end
267 if hdiag.flag_field
268     field(:,jdiag) = [ex,ey,ez,by-by0,bz]';
269 end
270 if hdiag.flag_kspec
271     kspec(:,jdiag) = kspectr([ex(X2),ey(X2),ez(X2),by(X2)-by0,bz(X2)]);
272 end
273
274 %
275 % graphics
276 %
277 figure(hdiag.fig)
278 %
279 for l=1:length(prm.diagtype)
280     %axes(hdiag.axes(l))
281     set(gcf,'CurrentAxes',hdiag.axes(l));
282     hdiag.nplt = l; % plate number
283
284     type = prm.diagtype(l);
285     switch type
286     case {1,2,3}
287         hdiag = plotphs(hdiag, type, jdiag);
288     case 4
289         hdiag = plotvs(hdiag,jdiag);
290     case {5,6,7,8,9}
291         hdiag = plotfield(hdiag, type-4, jdiag);
292     case 10
293         hdiag = plotwave(hdiag,jdiag);
294     case 11
295         hdiag = plotenergy(hdiag,jdiag);
296     case {12,13,14}
297         hdiag = plotvdist(hdiag,type-11,jdiag);
298     case {15,16,17,18,19}
299         hdiag = plotspectrum(hdiag,type-14,jdiag);
300     case {20,21,22,23,24}
301         % reserved for wk plot
302     case {25,26,27,28,29}
303         hdiag = plotts(hdiag,type-24,jdiag); % time series plot
304     case {30,31,32,33,34}
305         hdiag = plottsk(hdiag,type-29,jdiag); % time series k plot
306     otherwise
307         error(sprintf('Error: %s',mfilename));

```

```

308     end
309 end
310
311 % Title
312 set(hdiag.htitle,'String',sprintf('Time: %5.3f/%5.3f',jtime*prm.dt,prm.ntime*prm.dt)
313 if hdiag.flag_rot
314     rotate3d on
315 end
316 drawnow
317 return
318 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
319 function hdiag = diagnostics_init
320 global prm
321 global flag_exit
322 flag_exit = 0;
323 hdiag.flag_rot = 0;      % for rotate3d
324 hdiag.flag_field = 0;   % 0: do nothing, 1: save field data
325 hdiag.flag_eng = 0;     % 0: do nothing, 1: save energy data
326 hdiag.flag_kspec = 0;  % 0: do nothing, 1: save k-spectrum data
327 for l = 1:length(prm.diagtype)
328 end
329
330 %
331 % initialize graphics
332 %
333 hdiag.fig = figure;
334 set(0,'lang','en'); % for English menu
335 set(0,'DefaultAxesFontSize',10);
336 set(0,'DefaultAxesFontName','Helvetica');
337 %set(hdiag.fig,'Units','normalized','Position',[0,0,1,0.9]);
338 set(hdiag.fig,'DoubleBuffer','on');
339 set(hdiag.fig,'KeyPressFcn','pauseplot(gcbo)');
340 set(hdiag.fig,'DeleteFcn','exitplot(gcbo)');
341 %
342 hdiag.color = [
343     0 0 0.800000]; ... %blue
344     [ 0 0.500000 0]; ... %green
345     [1.000000 0 0]; ... %red
346     [0.750000 0.750000 0]; ... %yellow
347     [0.750000 0 0.750000]; ... %ma
348     [ 0 0 0.750000 0.750000]; ... %cyan
349     [ 0 0 0 0]; ... %black
350 %
351 for l = 1:length(prm.diagtype)
352     hdiag.axes(l) = subplot(2,ceil(length(prm.diagtype)/2),l);
353     set(gca,'DrawMode','fast')
354     set(gca,'NextPlot','ReplaceChildren')
355     box on
356     set(gca,'TickDir','out')
357     set(gca,'TickLength',[0.018 0.07]);
358     set(gca,'Layer','top')
359     set(gca,'ColorOrder',hdiag.color);
360
361     hxlabel = get(gca,'xlabel');
362     set(hxlabel,'Units','Normalized');
363     set(hxlabel,'Position',[0.5,-0.13,10]);
364     switch prm.diagtype(l)
365     case {4,10}
366         view(-37.5,30);
367         grid on
368         hdiag.flag_rot = 1;
369     case 11
370         hdiag.flag_eng = 1;
371     case {20,21,22,23,24}
372         hdiag.flag_field = 1;
373     case {25,26,27,28,29}

```

```

373     hdiag.flag_field = 1;
374     case {30,31,32,33,34}
375         hdiag.flag_kspec = 1;
376     end
377 end
378 axes(hdiag.axes(1))
379 hdiag.htitle = title(sprintf('Time: %5.3f/%5.3f',0,prm.ntime*prm.dt));
380 return;
381 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
382 function diagnostics_last(hdiag, jtime)
383     global prm ren
384     %global eplot bplot aplot
385
386     figure(hdiag.fig)
387     %
388     for k=1:length(prm.diagtype)
389         axes(hdiag.axes(k))
390         n = prm.diagtype(k);
391         switch n
392             %case 10
393             % axes(hdiag.hlegend)
394             case {20,21,22,23,24}
395                 plotspectr(n-19);
396                 %h = get(gca,'xlabel');
397                 %set(h,'Units','Normalized')
398                 %set(h,'Position',[0.5,-0.13,10])
399             end
400         end
401         if hdiag.flag_rot
402             rotate3d on
403         end
404     return;
405 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
406 function [ex,ey,ez] = efield(ex,ey,ez, by,bz, ajx,ajy,ajz)
407     global prm
408     global nxp1 nxp2
409     global X1 X2 X3
410     global tcs
411
412     if prm.iex == 0
413         ex(:) = 0;
414     else
415         ex(X2) = ex(X2)-2.0*ajx(X2);
416         ex(1) = ex(nxp1);
417         ex(nxp2) = ex(2);
418     end
419     ey(X2) = ey(X2) -tcs*(bz(X2)-bz(X1)) -2.0*ajy(X2);
420     ez(X2) = ez(X2) +tcs*(by(X3)-by(X2)) -2.0*ajz(X2);
421     ey(1) = ey(nxp1);
422     ez(1) = ez(nxp1);
423     ey(nxp2) = ey(2);
424     ez(nxp2) = ez(2);
425     return
426 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
427 function eng = energy(ex,ey,ez,by,bz,vx,vy,vz)
428     global prm ren
429     global nxp1 nxp2
430     global mass
431     global cs
432     global X1 X2 X3
433     global by0
434
435     eng = zeros(3,1);
436
437     % electric

```

```

438 te = sum(ex(X2).^2+ey(X2).^2+ez(X2).^2);
439 eng(1) = 0.5*te/prm.nx;
440 % magnetic
441 eng(2) = 0.5*sum((by(X2)-by0).^2 + bz(X2).^2)*cs/prm.nx;
442 % kinetic
443 n2 = 0;
444 for k=1:prm.ns
445     n1 = n2+1;
446     n2 = n2 + prm.np(k);
447
448     m = n1:n2;
449     ke = sum(prm.cv ./sqrt(cs-vx(m).^2-vy(m).^2-vz(m).^2)-1.0);
450     eng(3) = eng(3)+ke*mass(k)*cs/prm.nx;
451 end
452 return
453 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
454 %
455 % exitplot
456 %
457 function exitplot(hfig)
458     global flag_exit;
459     data = get(hfig,'UserData');
460     if strcmp(data,'exit') == 0
461         flag_exit = 1;
462         set(hfig,'UserData','exit');
463         uiresume(hfig);
464
465         delete(hfig);
466     end
467 return
468 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
469 function initial(prm, hdiag)
470     global q mass rho0
471     global npt % total number perticles
472     global slx % system length x
473     global bx0 by0
474     global ex ey ez
475     global bx by bz
476     global ajx ajy ajz
477     global rho
478
479     global vx vy vz
480     global x
481     global nxp1 nxp2 % nx+1, nx+2
482     global X1 X2 X3
483     global cs tcs % cv^2, 2*cv^2
484
485     global ifdiag
486     % diagnostics
487     global field % field data for wk plot
488     global kspec % k-spectrum
489     global eng % energy
490     slx = prm.nx;
491     npt = sum(prm.np(1:prm.ns));
492     nxp1 = prm.nx+1;
493     nxp2 = prm.nx+2;
494     X1 = 1:prm.nx;
495     X2 = 2:(prm.nx+1);
496     X3 = 3:(prm.nx+2);
497     cs = prm.cv*prm.cv;
498     tcs = 2.0*cs;
499     %
500     q = prm.nx./prm.np(1:prm.ns).*(prm.wp(1:prm.ns).^2)./prm.qm(1:prm.ns);
501     mass = q./prm.qm(1:prm.ns);
502     rho0 = -sum(q(1:prm.ns).*prm.np(1:prm.ns))/prm.nx;

```



```

503 rho = rho0*ones(nxp2,1);
504 %
505 theta = pi/180*prm.angle;
506 costh = cos(theta);
507 sinth = sin(theta);
508 %
509 b0 = prm.wc/prm.qm(1);
510 bx0 = b0*costh;
511 by0 = b0*sinth;
512
513
514 %
515 ifdiag = ceil(prm.ntime/prm.nplot);
516 %-- Field Initialization --
517 ex = zeros(nxp2,1);
518 ey = zeros(nxp2,1);
519 ez = zeros(nxp2,1);
520 by = ones(nxp2,1)*by0;
521 bz = zeros(nxp2,1);
522 ajx = zeros(nxp2,1);
523 ajy = zeros(nxp2,1);
524 ajz = zeros(nxp2,1);
525 rho = zeros(nxp2,1);
526 % perticles
527 x = zeros(npt,1);
528 vx = zeros(npt,1);
529 vy = zeros(npt,1);
530 vz = zeros(npt,1);
531 % diagnostics
532 if hdiag.flag_field
533     field = ones(5, nxp2, prm.nplot+1)*NaN;
534 end
535 if hdiag.flag_kspec
536     kspec = ones(5, prm.nx/2, prm.nplot+1)*NaN;
537 end
538 if hdiag.flag_eng
539     eng = ones(3,prm.nplot+1)*0;
540 end
541 %-- Particle Initialization --
542 n2=0;
543 for k=1:prm.ns
544     n1 = n2;
545     n2 = n2 +prm.np(k);
546     phi = pi/180.0*prm.pch(k);
547     vdpa = prm.vd(k)*cos(phi);
548     vdpe = prm.vd(k)*sin(phi);
549     xx = 0.0;
550     nphase = 1;
551     phase = 0.0;
552     for i = (n1+1):n2
553         if mod(i,nphase) == 0
554             phase = 2*pi*rand;
555             xx = xx+ prm.nx/prm.np(k);
556         else
557             phase = phase + 2*pi/nphase;
558         end
559         x(i) = xx;
560         if x(i) < 0.0
561             x(i) = x(i) +slx;
562         end
563         if x(i) >= slx
564             x(i) = x(i) -slx;
565         end
566         uxi = prm.vpa(k)*randn +vdpa;
567         uyi = prm.vpe(k)*randn +vdpe*cos(phase);

```

```

568     uz = prm.vpe(k)*randn +vdpe*sin(phase);
569
570     % rotation to the direction of the magnetic field
571     ux = costh*uxi-sinth*uyi;
572     uy = sinh*uxi+costh*uyi;
573     %
574     g = prm.cv /sqrt(cs +ux*ux +uy*uy +uz*uz);
575     vx(i) = ux*g;
576     vy(i) = uy*g;
577     vz(i) = uz*g;
578     end
579 end
580 return
581 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
582 %
583 % read input parameters
584 %
585 function prm = input_param(input_filename)
586     prm = [];
587     try
588         [str1, str2] = textread(input_filename, ...
589                               '%s%s','delimiter','=','commentstyle','matlab');
590     catch
591         error(dlg(sprintf('Can''t open input file: %s',input_filename),'Error'))
592     return
593 end
594
595 for l = 1:length(str1)
596     value = eval(char(str2(l)));
597     prmname= char(strread(char(str1(l)), '%s'));
598     switch prmname
599     case 'dx'
600         prm.dx = value;
601     case 'dt'
602         prm.dt = value;
603     case 'nx'
604         prm.nx = value;
605     case 'ntime'
606         prm.ntime = value;
607     case 'nplot'
608         prm.nplot = value;
609     case 'cv'
610         prm.cv = value;
611     case 'wc'
612         prm.wc = value;
613     case 'ajamp'
614         prm.ajamp = value;
615     case 'eamp'
616         prm.eamp = value;
617     case 'emax'
618         prm.emax = value;
619     case 'bamp'
620         prm.bamp = value;
621     case 'bmax'
622         prm.bmax = value;
623     case 'iex'
624         prm.iex = value;
625     case 'vmax'
626         prm.vmax = value;
627     case 'nv'
628         prm.nv = value;
629     case 'wj'
630         prm.wj = value;
631     case 'ns'
632         prm.ns = value;

```

```

633     case 'np'
634         prm.np = value;
635     case 'wp'
636         prm.wp = value;
637     case 'qm'
638         prm.qm = value;
639     case 'vpa'
640         prm.vpa = value;
641     case 'vpe'
642         prm.vpe = value;
643     case 'vd'
644         prm.vd = value;
645     case 'pch'
646         prm.pch = value;
647     case 'icolor'
648         prm.icolor = value;
649     case 'iparam'
650         prm.iparam = value;
651     case 'diagtype'
652         prm.diagtype = value;
653     case 'angle'
654         prm.angle = value;
655     otherwise
656         %disp(sprintf('Plese check input parameter %s.',prmname))
657     end
658 end
659 return;
660 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
661 %
662 % k-Spectrum
663 %
664 function ksp = kspectr(f)
665     global prm
666     spec = fft(f);
667     ksp = 2/prm.nx*abs(spec(1:(prm.nx/2),:));
668
669 return
670 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
671 %
672 % pauseplot
673 %
674 function pauseplot(hfig)
675     global flag_exit;
676     % ESC key
677     if double(get(hfig,'CurrentCharacter')) == 27
678         exitplot(hfig);
679         return
680     end
681     % any key
682     flag = get(hfig,'UserData');
683     if isempty(flag)
684         set(hfig,'UserData','pause');
685         set(hfig,'Name','(Pause)');
686     elseif strcmp(flag,'exit') == 0
687         set(hfig,'UserData',[]);
688         set(hfig,'Name','');
689         uiresume(hfig);
690     end
691
692 return
693 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
694 %
695 % Energy plot
696 %
697 function hdiag = plotenergy(hdiag, jdiag)

```

```

698 global eng
699 global prm ren
700 global ifdiag
701 totaleng = sum(eng);
702 en = [totaleng;eng(1,:);eng(2,:);eng(3,:)]*ren.s;
703
704 if jdiag == 1
705     t = (0:ifdiag:prm.ntime)*prm.dt;
706     color = [[0.9  0.9  0]; ...
707             [0    0 0.8]; ...
708             [0.8  0  0]; ...
709             [0    0.6  0]];
710     str = {'T','E','M','K'};
711     % lines & dots
712     hold on
713     for i = 1:4
714         helin(i) = plot(t,en(i,:),'Color',color(i,:));
715         hedot(i) = plot(t(jdiag),en(i,jdiag),'.', ...
716                        'Color',color(i,:), 'MarkerSize',15);
717
718         % text label
719         try
720             set(helin(i), 'DisplayName', char(str(i)));
721             set(hedot(i), 'DisplayName', char(str(i)));
722         catch;
723         end
724         hetxt(i) = text(t(jdiag),en(i,jdiag),str(i));
725         set(hetxt(i), 'VerticalAlignment', 'bottom', ...
726                'HorizontalAlignment', 'right', ...
727                'FontWeight', 'bold')
728     end
729     hold off
730     set(hetxt(1), 'HorizontalAlignment', 'left')
731
732     % label
733     ylabel('Energy');
734     set(gca, 'Yscale', 'log')
735     hxl = xlabel('Time');
736     set(hxl, 'Units', 'Normalized')
737     set(hxl, 'Position', [0.5,-0.13,10])
738     %
739     hdiag.plt(hdiag.nplt).t = t;
740     hdiag.plt(hdiag.nplt).helin = helin;
741     hdiag.plt(hdiag.nplt).hedot = hedot;
742     hdiag.plt(hdiag.nplt).hetxt = hetxt;
743
744 else
745     %
746     t = hdiag.plt(hdiag.nplt).t;
747     helin = hdiag.plt(hdiag.nplt).helin;
748     hedot = hdiag.plt(hdiag.nplt).hedot;
749     hetxt = hdiag.plt(hdiag.nplt).hetxt;
750     %
751     for i = 1:4
752         set(helin(i), 'ydata', en(i,:))
753         set(hedot(i), 'xdata', t(jdiag), 'ydata', en(i,jdiag))
754         set(hetxt(i), 'Position', [t(jdiag), en(i,jdiag), 0])
755     end
756
757 end
758
759 %
760 mmax = max(max(en));
761 mmax = 10^ceil(log10(mmax));
762 idx = find(en>0);

```

```

763   mmin = min(min(en(idx)));
764   mmin = 10^(floor(log10(mmin)));
765   if isnan(mmax)
766       mmax = eps*10;
767       mmin = eps;
768   end
769   axis([0 prm.ntime*prm.dt mmin mmax]);
770   return
771   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
772   %
773   % Field plot (Ex,Ey,Ez,By,Bz)
774   %
775   function hdiag = plotfield(hdiag, n, jdiag)
776   global prm ren
777   global nxp2
778   global X1 X2 X2
779   global ex ey ez
780   global by bz
781   global bx0 by0
782   xx = 1:nxp2;
783   switch n
784   case 1
785       f = ex*ren.e;
786       m = prm.emax;
787       xx = xx - 1.5;
788   case 2
789       f = ey*ren.e;
790       m = prm.emax;
791       xx = xx - 2.0;
792   case 3
793       f = ez*ren.e;
794       m = prm.emax;
795       xx = xx - 1.5;
796   case 4
797       f = (by-by0)*ren.b;
798       m = prm.bmax;
799       xx = xx - 2.0;
800   case 5
801       f = bz*ren.b;
802       m = prm.bmax;
803       xx = xx - 1.5;
804   end
805   if jdiag == 1
806       xx = xx*ren.x;
807       hdiag.plt(hdiag.nplt).hplot(n) = plot(xx, f, 'k', 'LineWidth', 1);
808       %
809       if prm.iparam
810           hold on
811           plot([xx(1),xx(end)], [0,0], 'k:');
812           hold off
813       end
814       %
815       set(gca, 'xlim', [0 prm.nx*ren.x]);
816       if m > 0
817           set(gca, 'ylim', [-m m]);
818       end
819       %
820       xlabel('X');
821       str = {'Ex', 'Ey', 'Ez', 'By', 'Bz'};
822       ylabel(str(n))
823   else
824       set(hdiag.plt(hdiag.nplt).hplot(n), 'ydata', f)
825   end
826
827   return

```

```

828 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
829 %
830 %
831 % m=1: ex
832 % 2: ey
833 % 3: ez
834 % 4: by
835 % 5: bz
836 function hdiag = plotkspectrum(hdiag, m, jdiag)
837     global ren prm
838     global X1 X2 X3
839     global bx0 by0
840
841     global ex ey ez
842     global     by bz
843
844     %
845     switch m
846     case 1
847         f = ex(X2);
848         re = ren.e;
849     case 2
850         f = ey(X2);
851         re = ren.e;
852     case 3
853         f = ez(X2);
854         re = ren.e;
855     case 4
856         f = by(X2)-by0;
857         re = ren.b;
858     case 5
859         f = bz(X2);
860         re = ren.b;
861     end
862     %
863     ksp = kspectr(f);
864     ksp = ksp*re;
865     ksp2 = ksp(2:end);
866
867     %
868     if jdiag == 1
869         kmin = 2*pi/prm.nx/ren.x;
870         kmax = kmin*(prm.nx/2);
871         hdiag.plt(hdiag.nplt).kmax = kmax;
872         hdiag.plt(hdiag.nplt).kmin = kmin;
873         k = 0:kmin:(kmax-kmin);
874         hdiag.plt(hdiag.nplt).hplot = plot(k,ksp,'k-');
875
876         set(gca,'Yscale','log');
877         xlabel('k');
878         str={'Ex','Ey','Ez','By','Bz'};
879         ylabel(str(m));
880     else
881         set(hdiag.plt(hdiag.nplt).hplot,'ydata',ksp);
882     end
883
884     %
885     mmax = max(ksp2);
886     mmin = min(ksp2);
887     if mmax == 0
888         mmax = 1;
889         mmin = 0.01;
890     end
891     mmax=10^ceil(log10(mmax));
892     mmin=10^floor(log10(mmin));

```

```

893
894 kmax = hdiag.plt(hdiag.nplt).kmax;
895 kmin = hdiag.plt(hdiag.nplt).kmin;
896 axis([0 kmax mmin mmax]);
897
898 return
899 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
900 %
901 % phase space plot
902 %
903 function hdiag = plotphs(hdiag, n, jdiag)
904 global prm ren npt
905 global cs
906
907 global vx vy vz
908 global x
909 switch n
910     case 1
911         vv = vx*ren.v;
912     case 2
913         vv = vy*ren.v;
914     case 3
915         vv = vz*ren.v;
916 end
917 xx = x*ren.x;
918 if jdiag == 1
919     hold on
920     mksize = 4+ceil(32/sqrt(npt));
921     if prm.icolor
922         n1 = 0;
923         n2 = 0;
924         color = get(gca,'ColorOrder');
925         for k = 1:prm.ns
926             n1 = n2+1;
927             n2 = n2+prm.np(k);
928             nln2 = n1:n2;
929             hplot(k) = plot(xx(nln2),vv(nln2),'k.', ...
930                 'color',color(mod(k-1,7)+1,:), ...
931                 'MarkerSize',mksize);
932             try;set(hplot(k),'DisplayName',sprintf('Sp. %d',k));catch;end;
933         end
934     else
935         hplot = plot(xx,vv,'k.','MarkerSize',mksize);
936     end
937     %
938     if prm.iparam
939         % CV
940         h = plot([0,prm.nx*ren.x],[prm.cv*ren.v,prm.cv*ren.v] , 'k:');
941         try;set(h,'DisplayName','CV');catch;end;
942         h = plot([0,prm.nx*ren.x],[-prm.cv*ren.v,prm.cv*ren.v] , 'k:');
943         try;set(h,'DisplayName','CV');catch;end;
944         % VD
945         if n == 1
946             for k = 1:prm.ns
947
948                 phi = pi/180.0*prm.pch(k);
949                 uxi = prm.vd(k)*cos(phi);
950                 %uyi = prm.vd(k)*sin(phi);
951                 uyi = 0;
952                 % rotation to the direction of the magnetic field
953                 theta = pi/180*prm.angle;
954                 costh = cos(theta);
955                 sinh = sin(theta);
956                 ux = costh*uxi-sinth*uyi;
957                 %uy = sinh*uxi+costh*uyi;

```

```

958
959         g = prm.cv/sqrt(cs + prm.vd(k)^2);
960         vv = ux*g;
961         h = plot([0,prm.nx*ren.x],[vv*ren.v,vv*ren.v],'k-');
962         try;set(h,'DisplayName',sprintf('VD(%d)',1));catch;end;
963     end
964 end
965
966 %
967     plot([0,prm.nx*ren.x],[0,0],'k:');
968 end
969
970 hold off
971 m = prm.vmax*ren.v;
972 axis([0 prm.nx*ren.x -m m]);
973 xlabel('X');
974 str={'Vx','Vy','Vz'};
975 ylabel(str(n))
976
977 hdiag.plt(hdiag.nplt).hplot = hplot;
978
979 else
980
981     hplot = hdiag.plt(hdiag.nplt).hplot;
982     if prm.icolor
983         n1 = 0;
984         n2 = 0;
985         for k = 1:prm.ns
986             n1 = n2+1;
987             n2 = n2+prm.np(k);
988             n12 = n1:n2;
989             set(hplot(k),'xdata',xx(n12),'ydata',vv(n12));
990         end
991     else
992         set(hplot,'xdata',xx,'ydata',vv);
993     end
994 end
995 end
996
997 return
998 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
999 %
1000 % wk plot
1001 %
1002 function plotspectr(n)
1003     global prm ren
1004     global X1 X2 X3
1005     global cs
1006     global field
1007
1008     str = {'Ex','Ey','Ez','By','Bz'};
1009     fielddata = squeeze(field(n,X2,:));
1010     switch n
1011     case {1,2,3}
1012         re = ren.e;
1013     case {4,5}
1014         re = ren.b;
1015     end
1016     %
1017     wk = wkfft(fielddata,prm.nx,prm.nplot,prm.nx,prm.nplot,0);
1018     wk = [fliplr(wk(4:2:end,1:2:(end-1)))',wk(1:2:end,1:2:(end-1))']';
1019     wk2 = [fliplr(wk(4:2:end,1:2:(end-1)))',wk(3:2:end,1:2:(end-1))']';
1020     wk = wk*re;
1021     wk = log10(wk);
1022     wk2 = log10(wk2*re);

```



```

1023 % omega
1024 isplot=prm.ntime/2;
1025 isdiag = prm.ntime/prm.nplot;
1026 wmin = 2*pi/(prm.dt)/2/(prm.nplot/2)/isdiag;
1027 wmax = wmin*(prm.nplot/2);
1028 w = 0:wmin:(wmax-wmin);
1029
1030 % wave number
1031 kmin = 2*pi/prm.nx/ren.x;
1032 kmax = kmin*(prm.nx/2-1);
1033 k = [-kmax:kmin:-kmin,0,kmin:kmin:kmax];
1034 %
1035 imagesc(k,w,wk);
1036 shading flat;
1037 set(gca,'Yscale','linear');
1038 xlabel('k');
1039 ylabel('\omega');
1040 wkmax = max(max(wk2));
1041 wkmin = min(min(wk2));
1042 if wkmax ~= wkmin
1043     caxis([wkmin, wkmax])
1044 end
1045 title(sprintf('log %s (min: %5.2g, max: %5.2g)',char(str(n)),wkmin,wkmax))
1046
1047 wmaxplot = (wmax-wmin);
1048 kmaxplot = kmax;
1049 axis([-kmaxplot,kmaxplot,0,wmaxplot])
1050 %
1051 if prm.iparam
1052     hold on
1053     kk = [-kmax,0,kmax];
1054     % light speed
1055     h = plot(kk,abs(kk*prm.cv*ren.v),'k:');
1056     try;set(h,'DisplayName','CV');catch;end;
1057     % vd
1058     for k = 1:prm.ns
1059         phi = pi/180.0*prm.pch(k);
1060         uxi = prm.vd(k)*cos(phi);
1061         %uyi = prm.vd(k)*sin(phi);
1062         uyi = 0;
1063         % rotation to the direction of the magnetic field
1064         theta = pi/180*prm.angle;
1065         costh = cos(theta);
1066         sinth = sin(theta);
1067         ux = costh*uxi-sinth*uyi;
1068         %uy = sinth*uxi+costh*uyi
1069         g = prm.cv/sqrt(cs + prm.vd(k)^2);
1070         vv = ux*g;
1071         ww = kk*vv*ren.v;
1072         i = find(ww < 0);
1073         ww(i) = NaN;
1074         h = plot(kk,ww,'k-.');
1075         try;set(h,'DisplayName',sprintf('VD(%d)',l));catch;end;
1076     end
1077
1078     % WP
1079     for k = 1:prm.ns
1080         h = plot([-kmaxplot,kmaxplot],[prm.wp(k),prm.wp(k)]/ren.t,'k--');
1081         try;set(h,'DisplayName',sprintf('WP(%d)',l));catch;end;
1082     end
1083     % WC
1084     h = plot([-kmaxplot,kmaxplot],abs([prm.wc,prm.wc])/ren.t,'k:');
1085     try;set(h,'DisplayName',sprintf('WC',l));catch;end;
1086
1087 hold off

```

```

1088 end
1089
1090 drawnow;
1091 return;
1092 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
1093 %
1094 % Time series field plot (Ex,Ey,Ez,By,Bz)
1095 %
1096 function hdiag = plotts(hdiag,n,jdiag)
1097 global prm ren
1098 global X1 X2 X3
1099 global nxp2;
1100
1101 global ifdiag
1102 global field
1103 fielddata = squeeze(field(n,:,:));
1104 switch n
1105 case {1,2,3}
1106     fielddata = fielddata*ren.e;
1107     m = prm.emax;
1108 case {4,5}
1109     fielddata = fielddata*ren.b;
1110     m = prm.bmax;
1111 end
1112
1113 str = {'Ex','Ey','Ez','By','Bz'};
1114 if jdiag == 1
1115     hold on;
1116     tt = (0:ifdiag:prm.ntime)*prm.dt;
1117     xx = 1:nxp2;
1118     if (n == 2) || (n == 4)
1119         xx = xx -2.0;
1120     else
1121         xx = xx -1.5;
1122     end
1123     xx = xx*ren.x;
1124     hdiag.plt(hdiag.nplt).hplot = imagesc(xx,tt,fielddata');
1125     xlabel('X');
1126     ylabel('Time');
1127     title(sprintf('%s (min: %g, max: %g)',char(str(n)),-m,m));
1128
1129     axis([0, prm.nx*ren.x, 0,tt(end)]);
1130     caxis([-m, m]);
1131
1132     hold off;
1133 else
1134     set(hdiag.plt(hdiag.nplt).hplot,'cdata',fielddata');
1135     end
1136     if m <= 0
1137         m = max(max(abs(fielddata)));
1138         caxis([-m, m]);
1139         title(sprintf('%s (min: %5.3g, max: %5.3g)',char(str(n)),-m,m));
1140     end
1141 return
1142 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
1143 %
1144 % Time series k-spectrum plot (Ex,Ey,Ez,By,Bz)
1145 %
1146 function hdiag = plottsk(hdiag,n,jdiag)
1147 global prm ren
1148 global X1 X2 X3
1149
1150 global ifdiag
1151 global kspec
1152 switch n

```

```

1153     case {1,2,3}
1154         re = ren.e;
1155     case {4,5}
1156         re = ren.b;
1157     end
1158     kspecdata = squeeze(kspec(n, :, :));
1159     kspecdata = kspecdata*re;
1160
1161     mmax = max(max(kspecdata(2:end, :)));
1162     mmin = max(min(kspecdata(2:end, :)));
1163     if mmax == 0
1164         mmax = 0.002;
1165         mmin = 0.001;
1166     end
1167     mmax=10^ceil(log10(mmax));
1168     mmin=10^floor(log10(mmin));
1169     kspecdata = log10(kspecdata);
1170
1171     if jdiag == 1
1172         hold on;
1173         tt = (0:ifdiag:prm.ntime)*prm.dt;
1174         kmin = 2*pi/prm.nx/ren.x;
1175         kmax = kmin*(prm.nx/2);
1176         kk = 0:kmin:(kmax-kmin);
1177         hdiag.plt(hdiag.nplt).hplot = imagesc(kk,tt,kspecdata');
1178         xlabel('k');
1179         ylabel('Time');
1180
1181         axis([0, kmax, 0,tt(end)]);
1182
1183         hold off;
1184     else
1185         set(hdiag.plt(hdiag.nplt).hplot,'cdata',kspecdata');
1186     end
1187     caxis([log10(mmin), log10(mmax)]);
1188     str = {'Ex','Ey','Ez','By','Bz'};
1189     title(sprintf('log %s (min: %4.1f, max: %4.1f)', ...
1190         char(str(n)),log10(mmin),log10(mmax)));
1191     return
1192     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
1193     function hdiag = plotvdist(hdiag, n, jdiag);
1194     global prm ren
1195     global vx vy vz
1196     persistent hplot htext vv
1197
1198     switch n
1199     case 1
1200         fv = vdist(vx)/ren.v;
1201     case 2
1202         fv = vdist(vy)/ren.v;
1203     case 3
1204         fv = vdist(vz)/ren.v;
1205     otherwise
1206         error(sprintf('Error: %s',mfilename))
1207     end
1208     [fvmax,index] = max(fv,[],2);
1209
1210     if jdiag == 1
1211         dv = 2*prm.vmax*ren.v/prm.nv;
1212         vv = -prm.vmax*ren.v:dv:(prm.vmax*ren.v);
1213         hold on
1214         if fvmax(1)
1215             if prm.iparam
1216                 h = plot(vv,fv(1,:),'-','Color',[0.7 0.7 0.8]);
1217                 try;set(h,'DisplayName','Electrons');catch;end;

```

```

1218     end
1219     hplot(1) = plot(vv,fv(1,:), 'Color', [0 0 0.9]);
1220     try;set(hplot(1), 'DisplayName', 'Electrons');catch;end;
1221     htext(1) = text(fv(1,index(1)),fvmax(1), 'e', 'Color', [0 0 0.9], ...
1222                   'VerticalAlignment', 'Bottom', 'FontWeight', 'bold');
1223     end
1224     if fvmax(2)
1225         if prm.iparam
1226             h = plot(vv,fv(2,:), '-', 'Color', [0.8 0.7 0.7]);
1227             try;set(h, 'DisplayName', 'Ions');catch;end;
1228         end
1229         hplot(2) = plot(vv,fv(2,:), 'Color', [0.9 0 0]);
1230         try;set(hplot(2), 'DisplayName', 'Ions');catch;end;
1231         htext(2) = text(fv(2,index(2)),fvmax(2), 'i', 'Color', [0.9 0 0], ...
1232                       'VerticalAlignment', 'Bottom', 'FontWeight', 'bold');
1233     end
1234     hcv = [];
1235     if prm.iparam
1236         ylim = get(gca, 'ylim');
1237         hcv(1) = plot([prm.cv*ren.v, prm.cv*ren.v], ylim, 'k:');
1238         hcv(2) = plot([-prm.cv*ren.v, -prm.cv*ren.v], ylim, 'k:');
1239         try;set(hcv(1), 'DisplayName', 'CV');catch;end;
1240         try;set(hcv(2), 'DisplayName', 'CV');catch;end;
1241     end
1242     hold off
1243     str = {'Vx', 'Vy', 'Vz'};
1244     xlabel(str(n));
1245     ylabel(sprintf('f(%s)', char(str(n))))
1246
1247     set(gca, 'xlim', [-prm.vmax*ren.v prm.vmax*ren.v])
1248
1249     hdiag.plt(hdiag.nplt).hplot = hplot;
1250     hdiag.plt(hdiag.nplt).htext = htext;
1251     hdiag.plt(hdiag.nplt).hcv = hcv;
1252     hdiag.plt(hdiag.nplt).vv = vv;
1253
1254 else
1255     hplot = hdiag.plt(hdiag.nplt).hplot;
1256     htext = hdiag.plt(hdiag.nplt).htext;
1257     hcv = hdiag.plt(hdiag.nplt).hcv;
1258     vv = hdiag.plt(hdiag.nplt).vv;
1259     ylim = get(gca, 'ylim');
1260     for i = 1:2
1261         if fvmax(i)
1262             set(hplot(i), 'ydata', fv(i,:))
1263             set(htext(i), 'Position', [vv(index(i)) fvmax(i) 0])
1264             if prm.iparam
1265                 set(hcv(i), 'ydata', ylim);
1266             end
1267         end
1268     end
1269 end
1270 return
1271 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
1272 %
1273 % plotvs
1274 %   velocity space plot
1275 %
1276 function hdiag = plotvs(hdiag,jdiag)
1277     global prm ren
1278     global nxp1 nxp2
1279     global vx vy vz
1280
1281     [view_az,view_el] = view;
1282

```

```

1283 vvx = vx*ren.v;
1284 vvy = vy*ren.v;
1285 vvz = vz*ren.v;
1286 m = prm.vmax*ren.v;
1287 if jdiag == 1
1288     hold on
1289     if prm.icolor
1290         n1 = 0;
1291         n2 = 0;
1292         for k = 1:prm.ns
1293             n1 = n2+1;
1294             n2 = n2+prm.np(k);
1295             n12 = n1:n2;
1296             hplot(k) = plot3(vvx(n12),vvy(n12),vvz(n12),'.', ...
1297                 'color',hdiag.color(mod(k-1,7)+1,:));
1298             try;set(hplot(k),'DisplayName',sprintf('Sp. %d',k));catch;end;
1299         end
1300     else
1301         hplot = plot3(vvx,vvy,vvz,'k.');
```

end

```

1302     axis equal
1303     axis([-m m -m m -m m]);
1304     xlabel('Vx')
1305     ylabel('Vy')
1306     zlabel('Vz')
1307     hdiag.plt(hdiag.nplt).hplot = hplot;
1308 else
1309     hplot = hdiag.plt(hdiag.nplt).hplot;
1310     if prm.icolor
1311         n1 = 0;
1312         n2 = 0;
1313         for k = 1:prm.ns
1314             n1 = n2+1;
1315             n2 = n2+prm.np(k);
1316             n12 = n1:n2;
1317             set(hplot(k),'xdata',vvx(n12),'ydata',vvy(n12),'zdata',vvz(n12))
1318         end
1319     else
1320         set(hplot,'xdata',vvx,'ydata',vvy,'zdata',vvz)
1321     end
1322 end
1323 view(view_az,view_el)
1324 return
1325 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
1326 %
1327 % plotwave
1328 % Vy,Vz,Ey,Ez,By,Bz - X plot
1329 %
1330 function hdiag = plotwave(hdiag, jdiag)
1331 global prm ren
1332 global nxp1 nxp2
1333 global bx0 by0
1334 global ex ey ez
1335 global by bz
1336 global vx vy vz
1337 global x
1338 global X2
1339 [view_az,view_el] = view;
1340 vvy = vy*ren.v;
1341 vvz = vz*ren.v;
1342 xx = x*ren.x;
```

```

1348 eey = ey(X2)*ren.e;
1349 eez = ez(X2)*ren.e;
1350 bby = (by(X2)-by0)*ren.b;
1351 bbz = bz(X2)*ren.b;
1352 m = prm.vmax*ren.v;
1353 flag_normalize = 1; % normalize
1354 if flag_normalize
1355     vvy = vvy/(prm.vmax*ren.v);
1356     vvz = vvz/(prm.vmax*ren.v);
1357     eey = eey/prm.emax;
1358     eez = eez/prm.emax;
1359     bby = bby/prm.bmax;
1360     bbz = bbz/prm.bmax;
1361     m = 1;
1362 end
1363 if jdiag == 1
1364     hold on
1365
1366     if prm.icolor
1367         n1 = 0;
1368         n2 = 0;
1369         for k = 1:prm.ns
1370             n1 = n2+1;
1371             n2 = n2+prm.np(k);
1372             n12 = n1:n2;
1373             hplot(k) = plot3(xx(n12),vvy(n12),vvz(n12),',', ...
1374                 'Color',hdiag.color(mod(k-1,7)+1,:));
1375             try;set(hplot(k),'DisplayName',sprintf('Sp. %d',k));catch;end;
1376         end
1377     else
1378         hplot = plot3(xx,vvy,vvz,'k.');
```

```

1379     end
1380
1381     xf = 0:prm.dx:(prm.nx-1)*ren.x;
1382     hplotf(1) = plot3(xf,eey,eez,'c-','LineWidth',2);
1383     hplotf(2) = plot3(xf,bby,bbz,'m-','LineWidth',2);
1384     try;set(hplotf(1),'DisplayName',sprintf('Ey, Ez'));catch;end;
1385     try;set(hplotf(2),'DisplayName',sprintf('By, Bz'));catch;end;
1386     hold off;
1387
1388     set(gca, 'DataAspectRatio',[prm.nx*ren.x/3.6,m,m])
1389     axis([0 prm.nx*ren.x -m m -m m]);
1390     xlabel('X')
1391     ylabel('Vy, Ey, By')
1392     zlabel('Vz, Ez, Bz')
1393     title(sprintf('Vmax:%4.2g, Emax:%4.2g, Bmax:%4.2g', ...
1394         prm.vmax*ren.v,prm.emax,prm.bmax))
1395
1396     hdiag.plt(hdiag.nplt).hplot = hplot;
1397     hdiag.plt(hdiag.nplt).hplotf = hplotf;
1398 else
1399     hplot = hdiag.plt(hdiag.nplt).hplot;
1400     hplotf = hdiag.plt(hdiag.nplt).hplotf;
1401     if prm.icolor
1402         n1 = 0;
1403         n2 = 0;
1404         for k = 1:prm.ns
1405             n1 = n2+1;
1406             n2 = n2+prm.np(k);
1407             n12 = n1:n2;
1408             set(hplot(k),'xdata',xx(n12),'ydata',vvy(n12),'zdata',vvz(n12))
1409         end
1410     else
1411         set(hplot,'xdata',xx,'ydata',vvy,'zdata',vvz)
1412     end

```

```

1413     set(hplotf(1),'ydata',eey,'zdata',eez)
1414     set(hplotf(2),'ydata',bby,'zdata',bbz)
1415 end
1416
1417 view(view_az,view_el)
1418 return
1419 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
1420 function ex = poisson(ex, rho)
1421     global nxp1 nxp2
1422     global X1 X2 X3
1423     global prm
1424
1425     for i = 2:nxp1
1426         ex(i)= ex(i-1)+rho(i);
1427     end
1428     ex0 = sum(ex(X2))/prm.nx;
1429     ex(X2) = ex(X2) -ex0;
1430     ex(1) = ex(nxp1);
1431     ex(nxp2)= ex(2);
1432 return
1433 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
1434 function x = position(x,vx)
1435     global prm
1436     global slx
1437     x = x +vx;
1438     x = x +slx.*(x<0.0);
1439     x = x -slx.*(x>=slx);
1440 return
1441 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
1442 %
1443 % renormalize
1444 %
1445 function [prm, ren] = renorm(prm)
1446     ren.x = prm.dx;
1447     ren.t = prm.dt/2;
1448     ren.v = ren.x/ren.t;
1449     ren.e = ren.x/(ren.t^2);
1450     ren.b = 1.0/ren.t;
1451     ren.j = ren.x/(ren.t^3);
1452     ren.r = 1.0/(ren.t^2);
1453     ren.s = (ren.x^2)/(ren.t^4);
1454     prm.cv = prm.cv / ren.v;
1455     prm.wc = prm.wc * ren.t;
1456     prm.wp = prm.wp .* ren.t;
1457     prm.vpa = prm.vpa ./ ren.v;
1458     prm.vpe = prm.vpe ./ ren.v;
1459     prm.vd = prm.vd ./ ren.v;
1460     prm.vmax = prm.vmax ./ ren.v;
1461     prm.wj = prm.wj * ren.t;
1462     prm.ajamp = prm.ajamp / ren.j;
1463
1464 return;
1465 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
1466 function [vx,vy,vz] = rvelocity(vx,vy,vz, ex,ey,ez, by,bz, x)
1467     global prm
1468     global bx0 by0
1469     global cs
1470     global nxp1 nxp2
1471     global X1 X2 X3
1472     work1 = zeros(nxp2,1);
1473     work2 = zeros(nxp2,1);
1474     work1(X2) = 0.5*(ex(X1)+ex(X2));
1475     work1(nxp2)= work1(2);
1476     work2(X2) = 0.5*(by(X3)+by(X2));
1477     work2(1) = work2(nxp1);

```

```

1478 %--
1479 n2=0;
1480 for k=1:prm.ns
1481     n1 = n2;
1482     n2 = n2 + prm.np(k);
1483     for m = (n1+1):n2
1484         i = floor(x(m) + 2.0);
1485         sf2 = (x(m) + 2.0 - i)*prm.qm(k);
1486         sf1 = prm.qm(k) - sf2;
1487
1488         ih = floor(x(m) + 1.5);
1489         sh2 = (x(m) + 1.5 - ih)*prm.qm(k);
1490         sh1 = prm.qm(k) - sh2;
1491         i1 = i+1;
1492         ih1 = ih+1;
1493
1494         ex1 = sf1*work1(i) + sf2*work1(i1);
1495         ey1 = sf1*ey(i) + sf2*ey(i1);
1496         ez1 = sh1*ez(ih) + sh2*ez(ih1);
1497         bx1 = bx0*prm.qm(k);
1498         by1 = sh1*work2(ih) + sh2*work2(ih1);
1499         bz1 = sh1*bz(ih) + sh2*bz(ih1);
1500
1501         g = prm.cv /sqrt(cs - vx(m)^2 - vy(m)^2 - vz(m)^2);
1502         ux = vx(m)*g + ex1;
1503         uy = vy(m)*g + ey1;
1504         uz = vz(m)*g + ez1;
1505         g = prm.cv/sqrt(cs + ux*ux + uy*uy + uz*uz);
1506         bx1 = bx1*g;
1507         by1 = by1*g;
1508         bz1 = bz1*g;
1509         boris = 2.0/(1+ bx1*bx1 +by1*by1 +bz1*bz1);
1510         uxt = ux + uy*bz1 - uz*by1;
1511         uyt = uy + uz*bx1 - ux*bz1;
1512         uzt = uz + ux*by1 - uy*bx1;
1513         ux = ux + boris*(uyt*bz1 -uzt*by1) +ex1;
1514         uy = uy + boris*(uzt*bx1 -uxt*bz1) +ey1;
1515         uz = uz + boris*(uxt*by1 -uyt*bx1) +ez1;
1516         g = prm.cv /sqrt(cs + ux*ux + uy*uy + uz*uz);
1517         vx(m) =ux*g;
1518         vy(m) =uy*g;
1519         vz(m) =uz*g;
1520     end
1521 end
1522 return;
1523 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
1524 function fv = vdist(v)
1525     global prm
1526     global q
1527     v2 = prm.vmax;
1528     v1 = -v2;
1529     dv = (v2 - v1)/prm.nv;
1530     dvi = 1.0/dv;
1531     fv = zeros(2,prm.nv+1);
1532
1533     n1 = 0;
1534     n2 = 0;
1535     for k=1:prm.ns
1536         n1 = n2;
1537         n2 = n1 + prm.np(k);
1538
1539         qabs = abs(q(k));
1540         if q(k) < 0
1541             qsign = 1;
1542         else

```



```

1543     qsign = 2;
1544     end
1545
1546     for m = (n1+1):n2
1547         if (v(m) < v1) | (v(m) >= v2)
1548             continue;
1549         end
1550         vi = (v(m)-v1)*dvi+1;
1551         i = floor(vi);
1552         i1 = i+1;
1553         s2 = (vi -i)*qabs;
1554         s1 = qabs -s2;
1555         fv(qsign,i) = fv(qsign,i) + s1;
1556         fv(qsign,i1) = fv(qsign,i1)+ s2;
1557     end
1558     end
1559     f = sum(fv,2)*dv;
1560     if f(1)
1561         fv(1,:) = fv(1,+)/f(1);
1562     end
1563     if f(2)
1564         fv(2,:) = fv(2,+)/f(2);
1565     end
1566
1567     return
1568     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
1569     %*****
1570     %   fourier transform in space and time
1571     %               by y.omura  rasc, kyoto univ.
1572     %
1573     % icnt=0  fft in both x(z) and y(t) compornents
1574     % icnt=1  fft in y(t) compornent
1575     % icnt=2  fft in x(z) compornent
1576     %
1577     % wk1: work 1
1578     % wk2: work 2
1579     %
1580     %*****
1581     function ar = wkfft(ar,n1,m1,n,m,icnt)
1582     rni=2.0/n;
1583     rmi=2.0/m;
1584     n2=n/2.0;
1585     m2=m/2.0;
1586     ii = 1:n;
1587     jj = 1:m;
1588     % X direction
1589     if icnt ~= 1
1590         for j=1:m
1591             wk1 = ar(ii,j);
1592             wk1 = realfft(wk1,n);
1593             ar(ii,j)=wk1(ii)'*rni;
1594         end
1595         ar(ii,1)=0.5*ar(ii,1);
1596         ar(ii,2)=0.5*ar(ii,2);
1597     end
1598     % Y direction
1599     if icnt ~= 2
1600         for i=1:n
1601             wk2=ar(i,1:m);
1602             wk2 =realfft(wk2,m);
1603             ar(i,jj)=wk2(jj)*rmi;
1604         end
1605     end
1606     ar(ii,1)=abs(ar(ii,1));
1607     ar(ii,2)=abs(ar(ii,2));

```

```

1608 ar1=0.5*ar(1,jj);
1609 ar2=0.5*ar(2,jj);
1610 ar(1,jj)=abs(ar1(jj));
1611 ar(2,jj)=abs(ar2(jj));
1612 %
1613 for i=1:2
1614     j=3;
1615     for l=2:m2
1616         ar1=ar(i,j);
1617         ar2=ar(i,j+1);
1618         sq=ar1*ar1+ar2*ar2;
1619         ara=sqrt(sq);
1620         if ara == 0
1621             ara=0.0001;
1622         end
1623         t1=acos(ar2/ara);
1624         if ar1 < 0
1625             t1=t1+pi;
1626         end
1627         ar(i ,j )=ara;
1628         ar(i ,j+1)=t1;
1629         j=j+2;
1630     end
1631 end
1632 %
1633 for j=1:2
1634     i=3;
1635     for l=2:n2
1636         ar1=ar(i,j);
1637         ar2=ar(i+1,j);
1638         sq=ar1*ar1+ar2*ar2;
1639         ara=sqrt(sq);
1640         ar(i ,j )=ara;
1641         ar(i+1,j )=ara;
1642         i=i+2;
1643     end
1644 end
1645 %
1646 j=3;
1647 for l=2:m2
1648     i=3;
1649     for k=2:n2
1650         cc=ar(i ,j ); % cos cos
1651         cs=ar(i ,j+1); % cos sin
1652         sc=ar(i+1,j ); % sin cos
1653         ss=ar(i+1,j+1); % sin sin
1654         sq=(cs-sc)^2+(cc+ss)^2;
1655         ar(i ,j)=0.5*sqrt(sq);
1656         sq=(cs+sc)^2+(cc-ss)^2;
1657         ar(i+1,j)=0.5*sqrt(sq);
1658         %
1659         ar1=ar(i,j);
1660         if ar1==0;
1661             ar1=0.0001;
1662         end
1663         tc1=0.5*(cs-sc)/ar1;
1664         t1=acos(tc1);
1665         tsign=cc+ss;
1666         if tsign < 0
1667             t1=t1+pi;
1668         end
1669     end
1670     %
1671     ar2=ar(i+1,j);
1672     if ar2==0;

```

```

1673     ar2=0.0001;
1674     end
1675     tc2=0.5*(cs+sc)/ar2;
1676     t2=acos(tc2);
1677     tsign=cc-ss;
1678     if tsign < 0 % note
1679         t2=t2+pi;
1680     end
1681     ar(i ,j+1)=t1;
1682     ar(i+1,j+1)=t2;
1683     i=i+2;
1684     end
1685     j=j+2;
1686 end
1687 return
1688 %
1689 %
1690 %
1691 function ret = realfft(x,n)
1692     x = fft(x,n);
1693     ret(1) = real(x(1));
1694     ret(2) = real(x(n/2+1));
1695     i = 1:(n/2-1);
1696     i2= i*2;
1697     ret(i2+1) = real(x(i+1));
1698     ret(i2+2) = imag(x(i+1));
1699 return;
1700 %%% end of "kempol.m" %%%
1701 % Cut the following text data, and save it as "default.dat"
1702 % in the same working directory where "kempol.m" listed above
1703 % is stored.
1704 %
1705 %%% beginning of "default.dat" %%%
1706 dx = 1.000000;
1707 dt = 0.040000;
1708 nx = 256.000000;
1709 ntime = 512.000000;
1710 cv = 20.000000;
1711 wc = -1.000000;
1712 angle = 0.000000;
1713 ns = 2.000000;
1714 np = [4096.000000, 4096.000000, ];
1715 wp = [2.000000, 2.000000, ];
1716 qm = [-1.000000, -1.000000, ];
1717 vpa = [1.000000, 1.000000, ];
1718 vpe = [1.000000, 1.000000, ];
1719 vd = [0.000000, 20.000000, ];
1720 pch = [0.000000, 60.000000, ];
1721 iex = 1.000000;
1722 ajamp = 0.000000;
1723 wj = 0.000000;
1724 nplot = 256.000000;
1725 nv = 100.000000;
1726 icolor = 1.000000;
1727 iparam = 1.000000;
1728 vmax = 20.000000;
1729 emax = 5.000000;
1730 bmax = 0.500000;
1731 diagtype = [11.000000, 4.000000, 5.000000, 9.000000, ];
1732 %%% end of parameters : default.dat%%

```