Advanced Methods for Space Simulations, edited by H. Usui and Y. Omura, pp. 209–235.

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

```
%%README:
% The following code ''kempo1.m'' is programmed in the MATLAB programming language.
% The code needs to be installed in a specified directory of the MATLAB system.
% In the command window of the MATLAB system, type the following:
%       kempo1  parameter_file
% If the ''parameter_file'' is omitted, the parameters in ''default.dat'' are used.
% The parameter file can be generated by modifying the contents of ''default.dat''
% attached at the end of this code, and it should be saved in the same directory
% where ''kempo1.m'' is stored.
%
%***************************************************************
%
%  KEMPO1
%    Kyoto university ElectroMagnetic Particle cOde: 1d version
%
%
%  FORTRAN Version (Ver. 1)
%          developed by
%          Yoshiharu Omura and Hiroshi Matsumoto
%          Research Institute for Sustainable Humanosphere,
%          Kyoto Universityy
%          Uji, Kyoto, 611-0011, Japan
%          E-mail: omurarish.kyoto-u.ac.jp
%          FAX:  +81-774-31-8463
%
%  MATLAB Version (Ver. 2)
%          developed for Lecture on "Electromagnetic Simulations"
%          by Yoshiharu Omura and Hideyuki Usui
%          Graduate Course of Electrical Engineering, Kyoto University
%
%  MATLAB Version with User Interface and Graphic Diagnostics (Ver. 3)
%          developed by Koichi Shin and Yoshiharu Omura
%          for 7th International School for Space Simulations (ISSS-7)
%          March 26-31, 2005, Kyoto Japan
%          Supported by COE21/KAGI Program
%
%          Copyright(c) 1993-2005, Space Simulation Group,
%          RISH, Kyoto University, All rights reserved.
%
%          Version 3.2   March 30, 2005
%
%***************************************************************}
function kempo1main(input_filename)
  clear global
  warning off
  global prm       % input parameters
  global ren       % normalize factor
```

```
global q mass rho0
global slx       % system length x
global nxp1 nxp2 % nx+1, nx+2
global X1 X2 X3  % 1:nx, 2:nxp1, 3:nxp3
global cs tcs    % c^2, 2*c^2
% field
global ex ey ez
global bx by bz
global ajx ajy ajz
global rho
% particles
global vx vy vz
global x
% diagnostics
global ifdiag    % interval for diagnostics
global eng       % for energy plot
global field     % field date for wk plot
global flag_exit
flag_exit = 0;

%-- read parameters --
if ~exist('input_filename')
  input_filename = 'default.dat';  % default input filename
end
prm = input_param(input_filename);
if isempty(prm)
  return
end

%-- initialize --
hdiag = diagnostics_init;
%--
[prm,ren] = renorm(prm);
initial(prm, hdiag);
x = position(x,vx);
if prm.iex
  rho= charge(x);
  ex = poisson(ex, rho);
end
%-----------
% Main loop
%-----------
jtime = 0;
jdiag = 1;
%-- diagnostics --
hdiag = diagnostics(hdiag, jtime, jdiag);
if prm.nplot == 0
  return
end

%--
for jtime = 1:prm.ntime
  %--
  if prm.iex == 2
    [ vx, vy, vz] = rvelocity(vx,vy,vz, ex,ey,ez, by,bz, x);
    x = position(x,vx);
    x = position(x,vx);
    rho= charge(x);
    ex = poisson(ex, rho);
  else
    [     by, bz] = bfield(   by,bz, ey,ez);
    [ vx, vy, vz] = rvelocity(vx,vy,vz, ex,ey,ez, by,bz, x);
    x = position(x,vx);
    [ajx,ajy,ajz] = current(ajx,ajy,ajz,vx,vy,vz,x,jtime);
    [     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
```

```
  %--
  n2=0;
  for k=1:prm.ns
    n1  = n2;
    n2  = n2 + prm.np(k);
    qh = q(k)*0.5;
    for m = (n1+1):n2
      ih = floor( x(m) + 1.5 );
      s2 = (x(m) + 1.5 - ih)*q(k);
      s1 = q(k) - s2;
      ih1= ih+1;
      ajy(ih)  = ajy(ih)  + vy(m)*s1;
      ajy(ih1) = ajy(ih1) + vy(m)*s2;
      ajz(ih)  = ajz(ih)  + vz(m)*s1;
      ajz(ih1) = ajz(ih1) + vz(m)*s2;
      %-- charge conversion method --
      if prm.iex
        qhs = qh * sign(vx(m));
        avx = abs(vx(m));
        x1 = x(m) + 2.0 -avx;
        x2 = x(m) + 2.0 +avx;
        i1 = floor(x1);
        i2 = floor(x2);
        ajx(i1) = ajx(i1) + (i2 - x1)*qhs;
        ajx(i2) = ajx(i2) + (x2 - i2)*qhs;
      end
    end
  end
  %-- boundary --
  ajx(nxp1) = ajx(1) + ajx(nxp1);
  ajx(2)    = ajx(2) + ajx(nxp2);
  ajy(nxp1) = ajy(1) + ajy(nxp1);
  ajy(2)    = ajy(2) + ajy(nxp2);
  ajy(1)    = ajy(nxp1);
  ajz(nxp1) = ajz(1) + ajz(nxp1);
  ajz(2)    = ajz(2) + ajz(nxp2);
  %--
  i=nxp1:-1:2;
  ajy(i) = (ajy(i) + ajy(i-1))*0.5;
  %-- external current source ----
  if prm.ajamp
    ajz(prm.nx/2+1) = ajz(prm.nx/2+1)+prm.ajamp*sin(prm.wj*jtime);
  else
  %--cancellation of uniform Jx,Jy,Jz components---
  ajxu = sum(ajx(2:nxp1))/prm.nx;
  ajyu = sum(ajy(2:nxp1))/prm.nx;
  ajzu = sum(ajz(2:nxp1))/prm.nx;
  ajx(2:nxp1) = ajx(2:nxp1) -ajxu;
  ajy(2:nxp1) = ajy(2:nxp1) -ajyu;
  ajz(2:nxp1) = ajz(2:nxp1) -ajzu;
  end

return
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% diagnostics
%
% hdiag: handles
%
function hdiag = diagnostics(hdiag, jtime, jdiag)
  global prm ren
  global ex ey ez by bz vx vy vz
  global bx0 by0
  global eng
  global field
```

```
global kspec
global X1 X2 X3

global flag_exit;
try
  % pause check
  flag = get(hdiag.fig,'UserData');
  if strcmp(flag,'pause')
    uiwait(hdiag.fig);
  end
  % exit chack
  flag = get(hdiag.fig,'UserData');
  if strcmp(flag,'exit')
    flag_exit = 1;
    return;
  end

catch
  return
end
%
if hdiag.flag_eng
  eng(:,jdiag) = energy(ex,ey,ez,by,bz,vx,vy,vz);
end
if hdiag.flag_field
  field(:,:,jdiag) = [ex,ey,ez,by-by0,bz]';
end
if hdiag.flag_kspec
  kspec(:,:,jdiag) = kspectr([ex(X2),ey(X2),ez(X2),by(X2)-by0,bz(X2)])';
end

%
% graphics
%
figure(hdiag.fig)
%
for l=1:length(prm.diagtype)
  %axes(hdiag.axes(l))
  set(gcf,'CurrentAxes',hdiag.axes(l));
  hdiag.nplt = l; % plate number

  type = prm.diagtype(l);
  switch type
   case {1,2,3}
    hdiag = plotphs(hdiag, type, jdiag);
   case 4
    hdiag = plotvs(hdiag,jdiag);
   case {5,6,7,8,9}
    hdiag = plotfield(hdiag, type-4, jdiag);
   case 10
    hdiag = plotwave(hdiag,jdiag);
   case 11
    hdiag = plotenergy(hdiag,jdiag);
   case {12,13,14}
    hdiag = plotvdist(hdiag,type-11,jdiag);
   case {15,16,17,18,19}
    hdiag = plotkspectrum(hdiag,type-14,jdiag);
   case {20,21,22,23,24}
    % reserved for wk plot
   case {25,26,27,28,29}
    hdiag = plotts(hdiag,type-24,jdiag); % time series plot
   case {30,31,32,33,34}
    hdiag = plottsk(hdiag,type-29,jdiag); % time series k plot
   otherwise
    error(sprintf('Error: %s',mfilename));
```

```
    end
  end

  % Title
  set(hdiag.htitle,'String',sprintf('Time: %5.3f/%5.3f',jtime*prm.dt,prm.ntime*prm.dt
  if hdiag.flag_rot
    rotate3d on
  end
  drawnow
return
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function hdiag = diagnostics_init
global prm
global flag_exit
flag_exit = 0;
hdiag.flag_rot = 0;       % for rotate3d
hdiag.flag_field = 0;     % 0: do nothing, 1: save field data
hdiag.flag_eng = 0;       % 0: do nothing, 1: save energy data
hdiag.flag_kspec = 0;     % 0: do nothing, 1: save k-spectrum data
for l = 1:length(prm.diagtype)
end

%
% initialize graphics
%
hdiag.fig = figure;
set(0,'lang','en'); % for English menu
set(0,'DefaultAxesFontSize',10);
set(0,'DefaultAxesFontName','Helvetica');
%set(hdiag.fig,'Units','normalized','Position',[0,0,1,0.9]);
set(hdiag.fig,'DoubleBuffer','on');
set(hdiag.fig,'KeyPressFcn','pauseplot(gcbo)');
set(hdiag.fig,'DeleteFcn','exitplot(gcbo)');
%
hdiag.color = [[       0          0   0.800000]; ... %blue
               [       0   0.500000          0]; ... %green
               [1.000000          0          0]; ... %red
               [0.750000   0.750000          0]; ... %yellow
               [0.750000          0   0.750000]; ... %ma
               [       0   0.750000   0.750000]; ... %cyan
               [       0          0          0]];    %black
%
for l = 1:length(prm.diagtype)
  hdiag.axes(l) = subplot(2,ceil(length(prm.diagtype)/2),l);
  set(gca,'DrawMode','fast')
  set(gca,'NextPlot','ReplaceChildren')
  box on
  set(gca,'TickDir','out')
  set(gca,'TickLength',[0.018 0.07]);
  set(gca,'Layer','top')
  set(gca,'ColorOrder',hdiag.color);

  hxlabel = get(gca,'xlabel');
  set(hxlabel,'Units','Normalized');
  set(hxlabel,'Position',[0.5,-0.13,10]);
  switch prm.diagtype(l)
   case {4,10}
    view(-37.5,30);
    grid on
    hdiag.flag_rot = 1;
   case 11
    hdiag.flag_eng = 1;
   case {20,21,22,23,24}
    hdiag.flag_field = 1;
   case {25,26,27,28,29}
```

```
    hdiag.flag_field = 1;
   case {30,31,32,33,34}
    hdiag.flag_kspec = 1;
  end
end
axes(hdiag.axes(1))
hdiag.htitle = title(sprintf('Time: %5.3f/%5.3f',0,prm.ntime*prm.dt));
return;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function diagnostics_last(hdiag, jtime)
  global prm ren
  %global eplot bplot aplot

  figure(hdiag.fig)
  %
  for k=1:length(prm.diagtype)
    axes(hdiag.axes(k))
    n = prm.diagtype(k);
    switch n
     %case 10
     % axes(hdiag.hlegend)
     case {20,21,22,23,24}
      plotspectr(n-19);
      %h = get(gca,'xlabel');
      %set(h,'Units','Normalized')
      %set(h,'Position',[0.5,-0.13,10])
    end
  end
  if hdiag.flag_rot
    rotate3d on
  end
return;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [ex,ey,ez] = efield(ex,ey,ez, by,bz, ajx,ajy,ajz)
  global prm
  global nxp1 nxp2
  global X1 X2 X3
  global tcs

  if prm.iex == 0
    ex(:) = 0;
  else
    ex(X2) = ex(X2)-2.0*ajx(X2);
    ex(1) = ex(nxp1);
    ex(nxp2) = ex(2);
  end
  ey(X2) = ey(X2) -tcs*(bz(X2)-bz(X1)) -2.0*ajy(X2);
  ez(X2) = ez(X2) +tcs*(by(X3)-by(X2)) -2.0*ajz(X2);
  ey(1) = ey(nxp1);
  ez(1) = ez(nxp1);
  ey(nxp2) = ey(2);
  ez(nxp2) = ez(2);
return
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function eng = energy(ex,ey,ez,by,bz,vx,vy,vz)
  global prm ren
  global nxp1 nxp2
  global mass
  global cs
  global X1 X2 X3
  global by0

  eng = zeros(3,1);

  % electric
```

```
  te = sum(ex(X2).^2+ey(X2).^2+ez(X2).^2);
  eng(1) = 0.5*te/prm.nx;
  % magnetic
  eng(2) = 0.5*sum((by(X2)-by0).^2 + bz(X2).^2)*cs/prm.nx;
  % kinetic
  n2 = 0;
  for k=1:prm.ns
    n1  = n2+1;
    n2  = n2 + prm.np(k);

    m = n1:n2;
    ke = sum(prm.cv ./sqrt(cs-vx(m).^2-vy(m).^2-vz(m).^2)-1.0);
    eng(3) = eng(3)+ke*mass(k)*cs/prm.nx;
  end
return
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% exitplot
%
function exitplot(hfig)
  global flag_exit;
  data = get(hfig,'UserData');
  if strcmp(data,'exit') == 0
    flag_exit = 1;
    set(hfig,'UserData','exit');
    uiresume(hfig);

    delete(hfig);
  end
return
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function initial(prm, hdiag)
  global q mass rho0
  global npt % total number perticles
  global slx % system length x
  global bx0 by0
  global ex ey ez
  global bx by bz
  global ajx ajy ajz
  global rho

  global vx vy vz
  global x
  global nxp1 nxp2 % nx+1, nx+2
  global X1 X2 X3
  global cs tcs    % cv^2, 2*cv^2

  global ifdiag
  % diagnostics
  global field % field data for wk plot
  global kspec % k-spectrum
  global eng   % energy
  slx = prm.nx;
  npt = sum(prm.np(1:prm.ns));
  nxp1 = prm.nx+1;
  nxp2 = prm.nx+2;
  X1 = 1:prm.nx;
  X2 = 2:(prm.nx+1);
  X3 = 3:(prm.nx+2);
  cs = prm.cv*prm.cv;
  tcs = 2.0*cs;
  %
  q = prm.nx./prm.np(1:prm.ns).*(prm.wp(1:prm.ns).^2)./prm.qm(1:prm.ns);
  mass = q./prm.qm(1:prm.ns);
  rho0 = -sum(q(1:prm.ns).*prm.np(1:prm.ns))/prm.nx;
```

```
503   rho0 = 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);
```

```
      uz  = prm.vpe(k)*randn +vdpe*sin(phase);

      % rotation to the direction of the magnetic field
      ux = costh*uxi-sinth*uyi;
      uy = sinth*uxi+costh*uyi;
      %
      g = prm.cv /sqrt(cs +ux*ux +uy*uy +uz*uz);
      vx(i) = ux*g;
      vy(i) = uy*g;
      vz(i) = uz*g;
    end
  end
return
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% read input parameters
%
function prm = input_param(input_filename)
  prm = [];
  try
    [str1, str2] = textread(input_filename, ...
                            '%s%s','delimiter','=;','commentstyle','matlab');
  catch
    errordlg(sprintf('Can''t open input file: %s',input_filename),'Error')
    return
  end

  for l = 1:length(str1)
    value = eval(char(str2(l)));
    prmname= char(strread(char(str1(l)),'%s'));
    switch prmname
    case 'dx'
      prm.dx = value;
    case 'dt'
      prm.dt = value;
    case 'nx'
      prm.nx = value;
    case 'ntime'
      prm.ntime = value;
    case 'nplot'
      prm.nplot = value;
    case 'cv'
      prm.cv = value;
    case 'wc'
      prm.wc = value;
    case 'ajamp'
      prm.ajamp = value;
    case 'eamp'
      prm.eamp = value;
    case 'emax'
      prm.emax = value;
    case 'bamp'
      prm.bamp = value;
    case 'bmax'
      prm.bmax = value;
    case 'iex'
      prm.iex = value;
    case 'vmax'
      prm.vmax = value;
    case 'nv'
      prm.nv = value;
    case 'wj'
      prm.wj = value;
    case 'ns'
      prm.ns = value;
```

```
    case 'np'
     prm.np = value;
    case 'wp'
      prm.wp = value;
    case 'qm'
      prm.qm = value;
    case 'vpa'
      prm.vpa = value;
    case 'vpe'
      prm.vpe = value;
    case 'vd'
      prm.vd = value;
    case 'pch'
      prm.pch = value;
    case 'icolor'
      prm.icolor = value;
    case 'iparam'
      prm.iparam = value;
    case 'diagtype'
      prm.diagtype = value;
    case 'angle'
      prm.angle = value;
    otherwise
      %disp(sprintf('Plese check input parameter %s.',prmname))
    end
  end
return;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% k-Spectrum
%
function ksp = kspectr(f)
  global prm
  spec = fft(f);
  ksp = 2/prm.nx*abs(spec(1:(prm.nx/2),:));

return
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% pauseplot
%
function pauseplot(hfig)
  global flag_exit;
  % ESC key
  if double(get(hfig,'CurrentCharacter')) == 27
    exitplot(hfig);
    return
  end
  % any key
  flag = get(hfig,'UserData');
  if isempty(flag)
    set(hfig,'UserData','pause');
    set(hfig,'Name','(Pause)');
  elseif strcmp(flag,'exit') == 0
    set(hfig,'UserData',[]);
    set(hfig,'Name','');
    uiresume(hfig);
  end

return
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Energy plot
%
function hdiag = plotenergy(hdiag, jdiag)
```

```
global eng
global prm ren
global ifdiag
totaleng = sum(eng);
en = [totaleng;eng(1,:);eng(2,:);eng(3,:)]*ren.s;

if jdiag == 1
  t = (0:ifdiag:prm.ntime)*prm.dt;
  color = [[0.9   0.9   0]; ...
           [0     0 0.8]; ...
           [0.8   0   0]; ...
           [0   0.6   0]];
  str = {'T','E','M','K'};
  % lines & dots
  hold on
  for i = 1:4
    helin(i) = plot(t,en(i,:),'Color',color(i,:));
    hedot(i) = plot(t(jdiag),en(i,jdiag),'.', ...
                    'Color',color(i,:),'MarkerSize',15);

    % text label
    try
      set(helin(i),'DisplayName',char(str(i)));
      set(hedot(i),'DisplayName',char(str(i)));
    catch;
    end
    hetxt(i) = text(t(jdiag),en(i,jdiag),str(i));
    set(hetxt(i),'VerticalAlignment','bottom', ...
                 'HorizontalAlignment','right', ...
                 'FontWeight','bold')
  end
  hold off
  set(hetxt(1),'HorizontalAlignment','left')

  % label
  ylabel('Energy');
  set(gca,'Yscale','log')
  hxl = xlabel('Time');
  set(hxl,'Units','Normalized')
  set(hxl,'Position',[0.5,-0.13,10])
  %
  hdiag.plt(hdiag.nplt).t = t;
  hdiag.plt(hdiag.nplt).helin = helin;
  hdiag.plt(hdiag.nplt).hedot = hedot;
  hdiag.plt(hdiag.nplt).hetxt = hetxt;

else
  %
  t = hdiag.plt(hdiag.nplt).t;
  helin = hdiag.plt(hdiag.nplt).helin;
  hedot = hdiag.plt(hdiag.nplt).hedot;
  hetxt = hdiag.plt(hdiag.nplt).hetxt;
  %
  for i = 1:4
    set(helin(i),'ydata',en(i,:))
    set(hedot(i),'xdata',t(jdiag),'ydata',en(i,jdiag))
    set(hetxt(i),'Position',[t(jdiag),en(i,jdiag),0])
  end

end

%
mmax = max(max(en));
mmax = 10^ceil(log10(mmax));
idx = find(en>0);
```

```
  mmin = min(min(en(idx)));
  mmin = 10^(floor(log10(mmin)));
  if isnan(mmax)
    mmax = eps*10;
    mmin = eps;
  end
  axis([0 prm.ntime*prm.dt mmin mmax]);
return
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Field plot (Ex,Ey,Ez,By,Bz)
%
function hdiag = plotfield(hdiag, n, jdiag)
  global prm ren
  global nxp2
  global X1 X2 X2
  global ex ey ez
  global by bz
  global bx0 by0
  xx = 1:nxp2;
  switch n
   case 1
    f = ex*ren.e;
    m = prm.emax;
    xx = xx -1.5;
   case 2
    f = ey*ren.e;
    m = prm.emax;
    xx = xx - 2.0;
   case 3
    f = ez*ren.e;
    m = prm.emax;
    xx = xx - 1.5;
   case 4
    f = (by-by0)*ren.b;
    m = prm.bmax;
    xx = xx - 2.0;
   case 5
    f = bz*ren.b;
    m = prm.bmax;
    xx = xx - 1.5;
  end
  if jdiag == 1
    xx = xx*ren.x;
    hdiag.plt(hdiag.nplt).hplot(n) = plot(xx, f,'k','LineWidth',1);
    %
    if prm.iparam
      hold on
      plot([xx(1),xx(end)],[0,0],'k:')
      hold off
    end
    %
    set(gca,'xlim',[0 prm.nx*ren.x]);
    if m > 0
      set(gca,'ylim',[-m m]);
    end
    %
    xlabel('X');
    str = {'Ex','Ey','Ez','By','Bz'};
    ylabel(str(n))
  else
    set(hdiag.plt(hdiag.nplt).hplot(n),'ydata',f)
  end

return
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%
% m=1: ex
%   2: ey
%   3: ez
%   4: by
%   5: bz
function hdiag = plotkspectrum(hdiag, m, jdiag)
  global ren prm
  global X1 X2 X3
  global bx0 by0

  global ex ey ez
  global    by bz

  %
  switch m
   case 1
    f = ex(X2);
    re = ren.e;
   case 2
    f = ey(X2);
    re = ren.e;
   case 3
    f = ez(X2);
    re = ren.e;
   case 4
    f = by(X2)-by0;
    re = ren.b;
   case 5
    f = bz(X2);
    re = ren.b;
  end
  %
  ksp = kspectr(f);
  ksp = ksp*re;
  ksp2 = ksp(2:end);

  %
  if jdiag == 1
    kmin = 2*pi/prm.nx/ren.x;
    kmax = kmin*(prm.nx/2);
    hdiag.plt(hdiag.nplt).kmax = kmax;
    hdiag.plt(hdiag.nplt).kmin = kmin;
    k = 0:kmin:(kmax-kmin);
    hdiag.plt(hdiag.nplt).hplot = plot(k,ksp,'k-');

    set(gca,'Yscale','log');
    xlabel('k');
    str={'Ex','Ey','Ez','By','Bz'};
    ylabel(str(m));
  else
    set(hdiag.plt(hdiag.nplt).hplot,'ydata',ksp);
  end

  %
  mmax = max(ksp2);
  mmin = min(ksp2);
  if mmax == 0
    mmax = 1;
    mmin = 0.01;
  end
  mmax=10^ceil(log10(mmax));
  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         n1n2 = n1:n2;
929         hplot(k) = plot(xx(n1n2),vv(n1n2),'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           sinth = sin(theta);
956           ux = costh*uxi-sinth*uyi;
957           %uy = sinth*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)',l));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
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);
```

```
% omega
isplot=prm.ntime/2;
isdiag = prm.ntime/prm.nplot;
wmin = 2*pi/(prm.dt)/2/(prm.nplot/2)/isdiag;
wmax = wmin*(prm.nplot/2);
w = 0:wmin:(wmax-wmin);

% wave number
kmin = 2*pi/prm.nx/ren.x;
kmax = kmin*(prm.nx/2-1);
k = [-kmax:kmin:-kmin,0,kmin:kmin:kmax];
%
imagesc(k,w,wk);
shading flat;
set(gca,'Yscale','linear');
xlabel('k');
ylabel('\omega');
wkmax = max(max(wk2));
wkmin = min(min(wk2));
if wkmax ~= wkmin
  caxis([wkmin, wkmax])
end
title(sprintf('log %s (min: %5.2g, max: %5.2g)',char(str(n)),wkmin,wkmax))

wmaxplot = (wmax-wmin);
kmaxplot = kmax;
axis([-kmaxplot,kmaxplot,0,wmaxplot])
%
if prm.iparam
  hold on
  kk = [-kmax,0,kmax];
  % light speed
  h = plot(kk,abs(kk*prm.cv*ren.v),'k:');
  try;set(h,'DisplayName','CV');catch;end;
  % Vd
    for k = 1:prm.ns
      phi = pi/180.0*prm.pch(k);
      uxi = prm.vd(k)*cos(phi);
      %uyi = prm.vd(k)*sin(phi);
      uyi = 0;
      % rotation to the direction of the magnetic field
      theta = pi/180*prm.angle;
      costh = cos(theta);
      sinth = sin(theta);
      ux = costh*uxi-sinth*uyi;
      %uy = sinth*uxi+costh*uyi
      g = prm.cv/sqrt(cs + prm.vd(k)^2);
      vv = ux*g;
      ww = kk*vv*ren.v;
      i = find(ww < 0);
      ww(i) = NaN;
      h = plot(kk,ww,'k-.');
      try;set(h,'DisplayName',sprintf('VD(%d)',l));catch;end;
    end

  % WP
  for k = 1:prm.ns
    h = plot([-kmaxplot,kmaxplot],[prm.wp(k),prm.wp(k)]/ren.t,'k--');
    try;set(h,'DisplayName',sprintf('WP(%d)',l));catch;end;
  end
  % WC
  h = plot([-kmaxplot,kmaxplot],abs([prm.wc,prm.wc])/ren.t,'k:');
  try;set(h,'DisplayName',sprintf('WC',l));catch;end;

  hold off
```

```
  end

  drawnow;
return;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Time series field plot (Ex,Ey,Ez,By,Bz)
%
function hdiag = plotts(hdiag,n,jdiag)
  global prm ren
  global X1 X2 X3
  global nxp2;

  global ifdiag
  global field
  fielddata = squeeze(field(n,:,:));
  switch n
   case {1,2,3}
    fielddata = fielddata*ren.e;
    m = prm.emax;
   case {4,5}
    fielddata = fielddata*ren.b;
    m = prm.bmax;
  end

  str = {'Ex','Ey','Ez','By','Bz'};
  if jdiag == 1
    hold on;
    tt = (0:ifdiag:prm.ntime)*prm.dt;
    xx = 1:nxp2;
    if (n == 2) || (n == 4)
      xx = xx -2.0;
    else
      xx = xx -1.5;
    end
    xx = xx*ren.x;
    hdiag.plt(hdiag.nplt).hplot = imagesc(xx,tt,fielddata');
    xlabel('X');
    ylabel('Time');
    title(sprintf('%s (min: %g, max: %g)',char(str(n)),-m,m));

    axis([0, prm.nx*ren.x, 0,tt(end)]);
    caxis([-m, m]);

    hold off;
  else
    set(hdiag.plt(hdiag.nplt).hplot,'cdata',fielddata');
  end
  if m <= 0
    m = max(max(abs(fielddata)));
    caxis([-m, m]);
    title(sprintf('%s (min: %5.3g, max: %5.3g)',char(str(n)),-m,m));
  end
return
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Time series k-spectrum plot (Ex,Ey,Ez,By,Bz)
%
function hdiag = plottsk(hdiag,n,jdiag)
  global prm ren
  global X1 X2 X3

  global ifdiag
  global kspec
  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;
```

```
      end
      hplot(1) = plot(vv,fv(1,:),'Color',[0 0 0.9]);
      try;set(hplot(1),'DisplayName','Electrons');catch;end;
      htext(1) = text(fv(1,index(1)),fvmax(1),' e','Color',[0 0 0.9], ...
                      'VerticalAlignment','Bottom','FontWeight','bold');
    end
    if fvmax(2)
      if prm.iparam
        h = plot(vv,fv(2,:),'-','Color',[0.8 0.7 0.7]);
        try;set(h,'DisplayName','Ions');catch;end;
      end
      hplot(2) = plot(vv,fv(2,:),'Color',[0.9 0 0]);
      try;set(hplot(2),'DisplayName','Ions');catch;end;
      htext(2) = text(fv(2,index(2)),fvmax(2),' i','Color',[0.9 0 0], ...
                      'VerticalAlignment','Bottom','FontWeight','bold');
    end
    hcv = [];
    if prm.iparam
      ylim = get(gca,'ylim');
      hcv(1) = plot([prm.cv*ren.v, prm.cv*ren.v], ylim,'k:');
      hcv(2) = plot([-prm.cv*ren.v, -prm.cv*ren.v], ylim,'k:');
      try;set(hcv(1),'DisplayName','CV');catch;end;
      try;set(hcv(2),'DisplayName','CV');catch;end;
    end
    hold off
    str = {'Vx','Vy','Vz'};
    xlabel(str(n));
    ylabel(sprintf('f(%s)',char(str(n))))

    set(gca,'xlim',[-prm.vmax*ren.v prm.vmax*ren.v])

    hdiag.plt(hdiag.nplt).hplot = hplot;
    hdiag.plt(hdiag.nplt).htext = htext;
    hdiag.plt(hdiag.nplt).hcv = hcv;
    hdiag.plt(hdiag.nplt).vv = vv;

  else
    hplot = hdiag.plt(hdiag.nplt).hplot;
    htext = hdiag.plt(hdiag.nplt).htext;
    hcv = hdiag.plt(hdiag.nplt).hcv;
    vv = hdiag.plt(hdiag.nplt).vv;
    ylim = get(gca,'ylim');
    for i = 1:2
      if fvmax(i)
        set(hplot(i),'ydata',fv(i,:))
        set(htext(i),'Position', [vv(index(i)) fvmax(i) 0])
        if prm.iparam
          set(hcv(i),'ydata', ylim);
        end
      end
    end
  end
return
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% plotvs
%   velocity space plot
%
function hdiag = plotvs(hdiag,jdiag)
  global prm ren
  global nxp1 nxp2
  global vx vy vz

  [view_az,view_el] = view;

```

```
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.');
1302     end
1303     hold off;
1304     axis equal
1305     axis([-m m -m m -m m]);
1306     xlabel('Vx')
1307     ylabel('Vy')
1308     zlabel('Vz')
1309     hdiag.plt(hdiag.nplt).hplot = hplot;
1310   else
1311     hplot = hdiag.plt(hdiag.nplt).hplot;
1312     if prm.icolor
1313       n1 = 0;
1314       n2 = 0;
1315       for k = 1:prm.ns
1316         n1 = n2+1;
1317         n2 = n2+prm.np(k);
1318         n12 = n1:n2;
1319         set(hplot(k),'xdata',vvx(n12),'ydata',vvy(n12),'zdata',vvz(n12))
1320       end
1321     else
1322       set(hplot,'xdata',vvx,'ydata',vvy,'zdata',vvz)
1323     end
1324   end
1325
1326   view(view_az,view_el)
1327 return
1328 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
1329 %
1330 % plotwave
1331 %   Vy,Vz,Ey,Ez,By,Bz - X plot
1332 %
1333 function hdiag = plotwave(hdiag, jdiag)
1334   global prm ren
1335   global nxp1 nxp2
1336   global bx0 by0
1337   global ex ey ez
1338   global    by bz
1339   global vx vy vz
1340   global x
1341   global X2
1342
1343   [view_az,view_el] = view;
1344
1345   vvy = vy*ren.v;
1346   vvz = vz*ren.v;
1347   xx  = x*ren.x;
```

```
eey = ey(X2)*ren.e;
eez = ez(X2)*ren.e;
bby = (by(X2)-by0)*ren.b;
bbz = bz(X2)*ren.b;
m = prm.vmax*ren.v;
flag_normalize = 1; % normalize
if flag_normalize
  vvy = vvy/(prm.vmax*ren.v);
  vvz = vvz/(prm.vmax*ren.v);
  eey = eey/prm.emax;
  eez = eez/prm.emax;
  bby = bby/prm.bmax;
  bbz = bbz/prm.bmax;
  m = 1;
end
if jdiag == 1
  hold on

  if prm.icolor
    n1 = 0;
    n2 = 0;
    for k = 1:prm.ns
      n1 = n2+1;
      n2 = n2+prm.np(k);
      n12 = n1:n2;
      hplot(k) = plot3(xx(n12),vvy(n12),vvz(n12),'.', ...
                       'Color',hdiag.color(mod(k-1,7)+1,:));
      try;set(hplot(k),'DisplayName',sprintf('Sp. %d',k));catch;end;
    end
  else
    hplot = plot3(xx,vvy,vvz,'k.');
  end

  xf = 0:prm.dx:(prm.nx-1)*ren.x;
  hplotf(1) = plot3(xf,eey,eez,'c-','LineWidth',2);
  hplotf(2) = plot3(xf,bby,bbz,'m-','LineWidth',2);
  try;set(hplotf(1),'DisplayName',sprintf('Ey, Ez'));catch;end;
  try;set(hplotf(2),'DisplayName',sprintf('By, Bz'));catch;end;
  hold off;

  set(gca, 'DataAspectRatio',[prm.nx*ren.x/3.6,m,m])
  axis([0 prm.nx*ren.x -m m -m m]);
  xlabel('X')
  ylabel('Vy, Ey, By')
  zlabel('Vz, Ez, Bz')
  title(sprintf('Vmax:%4.2g, Emax:%4.2g, Bmax:%4.2g', ...
                prm.vmax*ren.v,prm.emax,prm.bmax))

  hdiag.plt(hdiag.nplt).hplot = hplot;
  hdiag.plt(hdiag.nplt).hplotf = hplotf;
else
  hplot = hdiag.plt(hdiag.nplt).hplot;
  hplotf = hdiag.plt(hdiag.nplt).hplotf;
  if prm.icolor
    n1 = 0;
    n2 = 0;
    for k = 1:prm.ns
      n1 = n2+1;
      n2 = n2+prm.np(k);
      n12 = n1:n2;
      set(hplot(k),'xdata',xx(n12),'ydata',vvy(n12),'zdata',vvz(n12))
    end
  else
    set(hplot,'xdata',xx,'ydata',vvy,'zdata',vvz)
  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);
```

```
  %--
  n2=0;
  for  k=1:prm.ns
    n1  = n2;
    n2  = n2 + prm.np(k);
    for m = (n1+1):n2
      i = floor(x(m) +2.0);
      sf2 = (x(m) +2.0 -i)*prm.qm(k);
      sf1 = prm.qm(k) - sf2;

      ih = floor(x(m) +1.5);
      sh2 = (x(m) +1.5 -ih)*prm.qm(k);
      sh1 = prm.qm(k) - sh2;
      i1 = i+1;
      ih1 = ih+1;

      ex1 = sf1*work1(i) + sf2*work1(i1);
      ey1 = sf1*ey(i)    + sf2*ey( i1);
      ez1 = sh1*ez(ih)   + sh2*ez(ih1);
      bx1 = bx0*prm.qm(k);
      by1 = sh1*work2(ih)+ sh2*work2(ih1);
      bz1 = sh1*bz(ih)   + sh2*bz(ih1);

      g = prm.cv /sqrt(cs - vx(m)^2 - vy(m)^2 - vz(m)^2);
      ux = vx(m)*g + ex1;
      uy = vy(m)*g + ey1;
      uz = vz(m)*g + ez1;
      g = prm.cv/sqrt(cs + ux*ux +uy*uy +uz*uz);
      bx1 = bx1*g;
      by1 = by1*g;
      bz1 = bz1*g;
      boris = 2.0/(1+ bx1*bx1 +by1*by1 +bz1*bz1);
      uxt = ux + uy*bz1 - uz*by1;
      uyt = uy + uz*bx1 - ux*bz1;
      uzt = uz + ux*by1 - uy*bx1;
      ux = ux + boris*(uyt*bz1 -uzt*by1) +ex1;
      uy = uy + boris*(uzt*bx1 -uxt*bz1) +ey1;
      uz = uz + boris*(uxt*by1 -uyt*bx1) +ez1;
      g = prm.cv /sqrt(cs + ux*ux + uy*uy + uz*uz);
      vx(m) =ux*g;
      vy(m) =uy*g;
      vz(m) =uz*g;
    end
  end
return;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function fv = vdist(v)
  global prm
  global q
  v2 =  prm.vmax;
  v1 = -v2;
  dv = (v2 - v1)/prm.nv;
  dvi = 1.0/dv;
  fv = zeros(2,prm.nv+1);

  n1 = 0;
  n2 = 0;
  for k=1:prm.ns
    n1 = n2;
    n2 = n1 + prm.np(k);

    qabs = abs(q(k));
    if q(k) < 0
      qsign = 1;
    else
```

```
      qsign = 2;
    end

    for m = (n1+1):n2
      if (v(m) < v1) | (v(m) >= v2)
        continue;
      end
      vi = (v(m)-v1)*dvi+1;
      i = floor(vi);
      i1 = i+1;
      s2 = (vi -i)*qabs;
      s1 = qabs -s2;
      fv(qsign,i) = fv(qsign,i) + s1;
      fv(qsign,i1)= fv(qsign,i1)+ s2;
    end
  end
  f = sum(fv,2)*dv;
  if f(1)
      fv(1,:) = fv(1,:)/f(1);
  end
  if f(2)
      fv(2,:) = fv(2,:)/f(2);
  end

return
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%*****************************************************************
%   fourier transform in space and time
%                  by y.omura   rasc, kyoto univ.
%
% icnt=0  fft in both x(z) and y(t) compornents
% icnt=1  fft in y(t) compornent
% icnt=2  fft in x(z) compornent
%
% wk1: work 1
% wk2: work 2
%
%*****************************************************************
function ar = wkfft(ar,n1,m1,n,m,icnt)
rni=2.0/n;
rmi=2.0/m;
n2=n/2.0;
m2=m/2.0;
ii = 1:n;
jj = 1:m;
% X direction
if icnt ~= 1
  for j=1:m
    wk1 = ar(ii,j);
    wk1 = realfft(wk1,n);
    ar(ii,j)=wk1(ii)'*rni;
  end
  ar(ii,1)=0.5*ar(ii,1);
  ar(ii,2)=0.5*ar(ii,2);
end
% Y direction
if icnt ~= 2
  for i=1:n
    wk2=ar(i,1:m);
    wk2 =realfft(wk2,m);
    ar(i,jj)=wk2(jj)*rmi;
  end
end
ar(ii,1)=abs(ar(ii,1));
ar(ii,2)=abs(ar(ii,2));
```

```
ar1=0.5*ar(1,jj);
ar2=0.5*ar(2,jj);
ar(1,jj)=abs(ar1(jj));
ar(2,jj)=abs(ar2(jj));
%
for i=1:2
  j=3;
  for l=2:m2
    ar1=ar(i,j);
    ar2=ar(i,j+1);
    sq=ar1*ar1+ar2*ar2;
    ara=sqrt(sq);
    if ara == 0
      ara=0.0001;
    end
    t1=acos(ar2/ara);
    if ar1 < 0
      t1=t1+pi;
    end
    ar(i  ,j  )=ara;
    ar(i  ,j+1)=t1;
    j=j+2;
  end
end
%
for j=1:2
  i=3;
  for l=2:n2
    ar1=ar(i,j);
    ar2=ar(i+1,j);
    sq=ar1*ar1+ar2*ar2;
    ara=sqrt(sq);
    ar(i  ,j  )=ara;
    ar(i+1,j  )=ara;
    i=i+2;
  end
end
%
j=3;
for l=2:m2
  i=3;
  for k=2:n2
    cc=ar(i  ,j  ); % cos cos
    cs=ar(i  ,j+1); % cos sin
    sc=ar(i+1,j  ); % sin cos
    ss=ar(i+1,j+1); % sin sin
    sq=(cs-sc)^2+(cc+ss)^2;
    ar(i  ,j)=0.5*sqrt(sq);
    sq=(cs+sc)^2+(cc-ss)^2;
    ar(i+1,j)=0.5*sqrt(sq);
    %
    ar1=ar(i,j);
    if ar1==0;
      ar1=0.0001;
    end
    tc1=0.5*(cs-sc)/ar1;
    t1=acos(tc1);
    tsign=cc+ss;
    if tsign < 0
      t1=t1+pi;
    end

    %
    ar2=ar(i+1,j);
    if ar2==0;
```

```
      ar2=0.0001;
    end
    tc2=0.5*(cs+sc)/ar2;
    t2=acos(tc2);
    tsign=cc-ss;
    if tsign < 0 % note
      t2=t2+pi;
    end
    ar(i  ,j+1)=t1;
    ar(i+1,j+1)=t2;
    i=i+2;
  end
  j=j+2;
end
return
%
%
%
function ret = realfft(x,n)
  x = fft(x,n);
  ret(1) = real(x(1));
  ret(2) = real(x(n/2+1));
  i = 1:(n/2-1);
  i2= i*2;
  ret(i2+1) = real(x(i+1));
  ret(i2+2) = imag(x(i+1));
return;
%%% end of "kempo1.m" %%%
% Cut the following text data, and save it as "default.dat"
% in the same working directory where "kempo1.m" listed above
% is stored.
%
%%% beginning of "default.dat" %%%
dx = 1.000000;
dt = 0.040000;
nx = 256.000000;
ntime = 512.000000;
cv = 20.000000;
wc = -1.000000;
angle = 0.000000;
ns = 2.000000;
np = [4096.000000, 4096.000000, ];
wp = [2.000000, 2.000000, ];
qm = [-1.000000, -1.000000, ];
vpa = [1.000000, 1.000000, ];
vpe = [1.000000, 1.000000, ];
vd = [0.000000, 20.000000, ];
pch = [0.000000, 60.000000, ];
iex = 1.000000;
ajamp = 0.000000;
wj = 0.000000;
nplot = 256.000000;
nv = 100.000000;
icolor = 1.000000;
iparam = 1.000000;
vmax = 20.000000;
emax = 5.000000;
bmax = 0.500000;
diagtype = [11.000000, 4.000000, 5.000000, 9.000000, ];
 %%% end of parameters : default.dat%%%%
```