Computational Methods for Kinetic Processes in Plasma Physics

Ken Nishikawa

Department of Physics/UAH

Over views:
Simulation methods Macroscopic and Microscopic Processes

June 1, 2015

Context

1. Brief history of PIC simulations
2. Introduction
3. Basic structures of 3D RPIC code
4. Basic structures of 3D GRMHD code (not in this class)
5. Summary and Future Improvements

Brief history of PIC simulations

In late 1950s John Dawson began 1D electrostatic “charge-sheet” experiments at Princeton, later at UCLA.

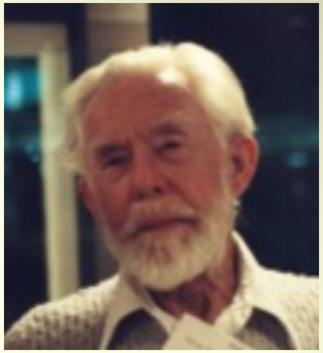
1965 Hockney, Buneman -- introduced grids and direct Poisson solver

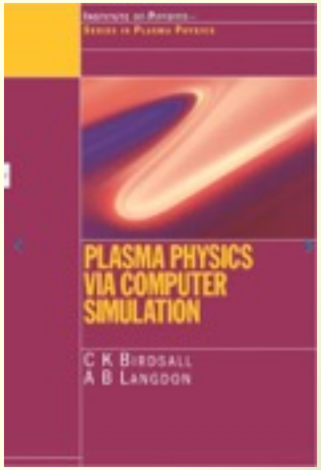
1970-s theory of electrostatic PIC developed (Langdon)

First electromagnetic codes
1980s-90s 3D EM PIC takes off
(Dawson, Rev. Modern Physics, 1983)

“PIC text books” come out in 1988 and 1990

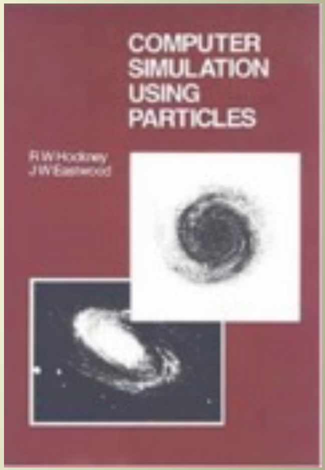
Integrated circuit doubles ~ every two years (Moore’s law)





PLASMA PHYSICS
VIA COMPUTER
SIMULATION

C K BIRDSALL
A B LANGDON



COMPUTER
SIMULATION
USING
PARTICLES

R W Hockney
J W Eastwood

Experiments

Simulations

Plasma Physics

Theory

Nonlinear phenomena

PIC
Hybrid (ion: kinetic, electron: fluid)
Vlasov (collisionless)
Fokker-Planck (with collision)
MHD with PIC
MHD
RMHD
RRMH
GRMHD
Einstein-RMHD

Applications

Particle acceleration
Instabilities
Radiation
Anomalous resistivity
Reconnection
Relativistic jets
Cosmic rays

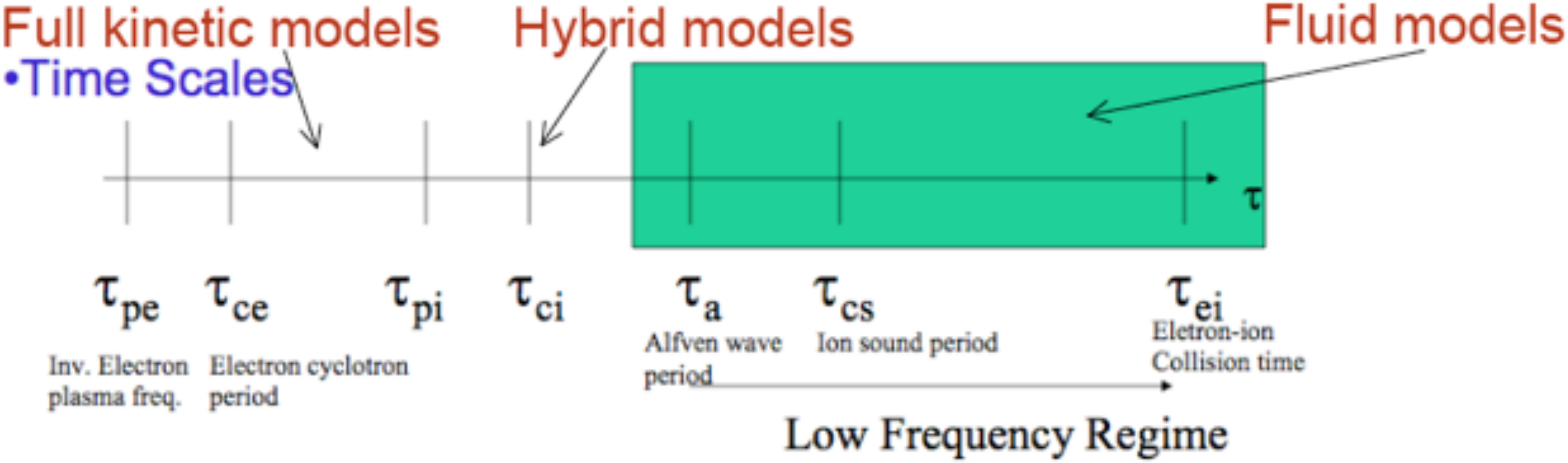
Linear
Quasi-linear

Characteristic time and length scales

$$\omega_P = \left(\frac{4\pi\ ne^2}{m}\right)^{1/2} \qquad \lambda_D = \frac{V_{thermal}}{\omega_p} \propto \left(\frac{T}{n}\right)^{1/2} \qquad \lambda_{skin} = c/\omega_p \qquad \omega_c = \frac{eB}{mc}$$

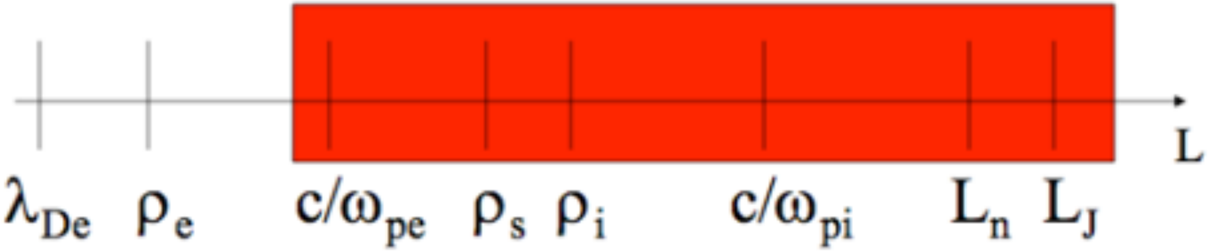
Plasma frequency Debye length skin depth Larmor

- Time Scales



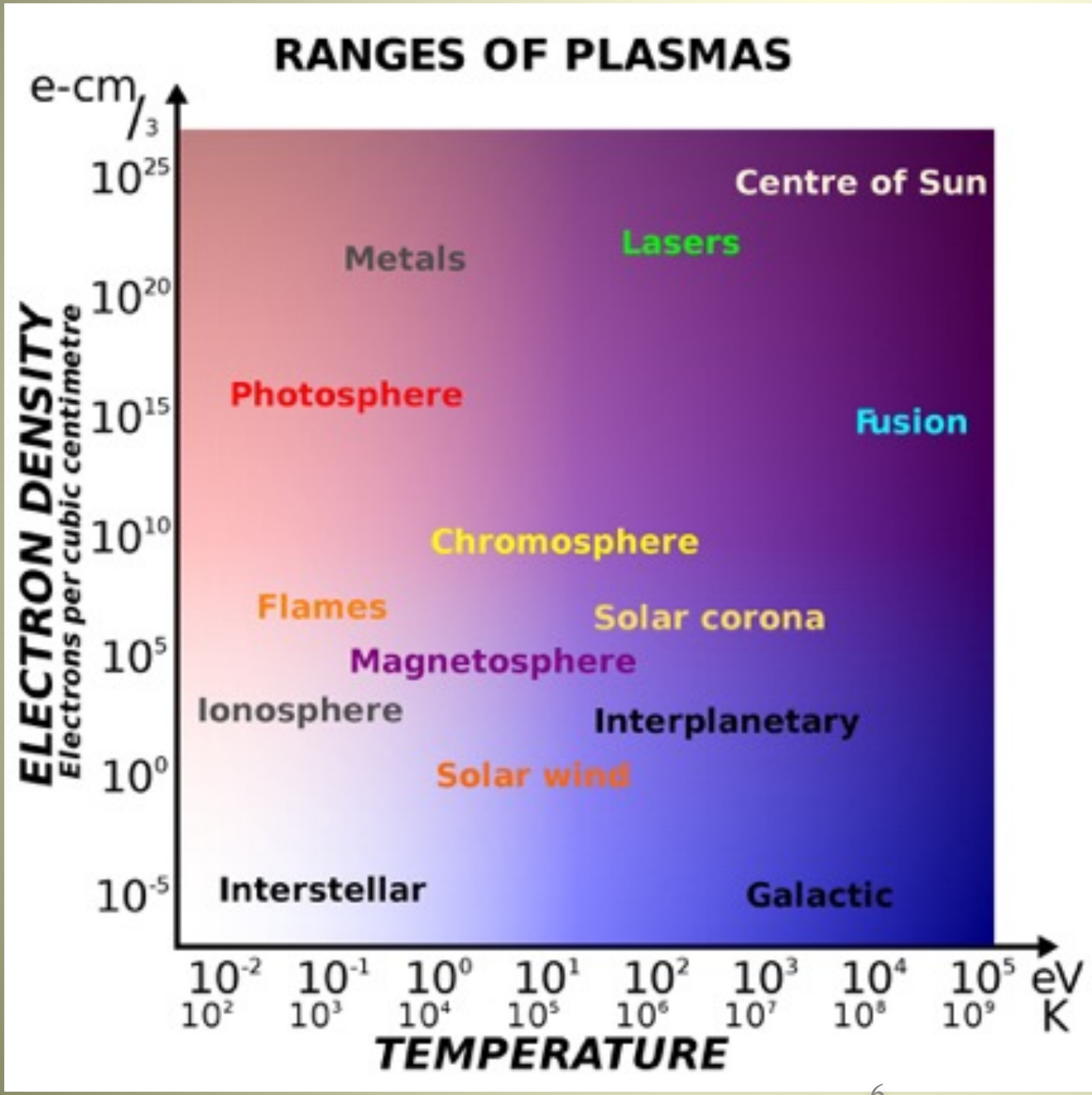
- Length Scales

$\rho_e = v_{te}/\omega_{ce}$ $\rho_s = \sqrt{T_i/T}\rho_i$ $L_n = \nabla n/n$



Collisionless plasmas

$$L \gg \lambda_D, \quad t \gg \omega_p^{-1}, \omega_c^{-1}$$



Number of particles in Debye cubes

$$N_D \equiv n\lambda_D^3 \gg 1$$

Magnetic field in the Universe

- Magnetic and gravitational fields play a important role in determining the evolution of the matter in many astrophysical objects
- Magnetic field can be amplified by instabilities, plasma contraction or shear motion.
- Even when the magnetic field is weak initially, the magnetic field glows in the short time scale and influences plasma dynamics of the system
- Poynting flux dominated jets can be convert its energy

  to kinetic energy due to reconnection

Plasmas in the Universe

- The major constituents of the universe are made of plasmas.
- When the temperature of gas is more than 10^4K, the gas becomes fully ionized plasmas (4th phase of matter).
- Plasmas are applied to many astrophysical phenomena.
- Plasmas are treated in several ways
  - particle-in-cell (PIC) (microscopic)
  - magnetohydrodynamics, MHD (macroscopic) (not covered)
  - hybrid (fluid electron and kinetic ions) (not covered)
  - MHD with test particles (fluid mixed with particles) (not covered)
  - particles with photons (not covered)

3D Relativistic particle-in-cell code

Kinetic processes are included in this code

- Particle acceleration can be investigated
- Calculation of radiation can be calculated
- Simulation system size is limited due to necessity of resolving Debye length (Skin depth)
- Global dynamics of plasma such as large jets cannot be simulated

This simulation method is complimentary to MHD method which will be described briefly later

Collisionless plasma can be described by Vlasov-Maxwell equations with distribution function f(**x**, **v**, t) (6 dimensions):

$$\frac{\partial f}{\partial t} + \vec{v}\cdot\frac{\partial f}{\partial \vec{x}} + \frac{q}{m}\left(\vec{E} + \frac{\vec{v}\times\vec{B}}{c}\right)\cdot\frac{\partial f}{\partial \vec{v}} = 0,$$

$$\nabla\cdot\vec{E} = 4\pi\int qfd^3\vec{v}$$

$$\nabla\times\vec{B} = \frac{1}{c}\frac{\partial\vec{E}}{\partial t} + \frac{4\pi}{c}\int q\vec{v}fd^3\vec{v},$$

$$\nabla\cdot\vec{B} = 0, \quad \nabla\times\vec{E} = -\frac{1}{c}\frac{\partial\vec{B}}{\partial t}.$$

Direct calculation of this set of equations – 6D
Improvements have been made, but difficult
to calculate using this method

$$\frac{d\vec{v}_j}{dt} = \frac{q_j}{m_j}\left(\vec{E} + \frac{\vec{v}_j\times\vec{B}}{c}\right)$$

$$\frac{d\vec{x}}{dt} = \vec{v}$$

$$\nabla\times\vec{E} = -\frac{1}{c}\frac{\partial\vec{B}}{\partial t}$$

$$\nabla\times B = \frac{1}{c}\frac{\partial\vec{E}}{\partial t} + \frac{4\pi\vec{J}}{c}$$

$$\nabla\cdot B = 0, \quad \nabla\cdot E = 4\pi\rho$$

$$\rho(\vec{x}) = \sum_j q_j\delta(\vec{x} - \vec{x}_j)$$

$$\vec{j}(\vec{x}) = \sum_j q_j\vec{v}_j\delta(\vec{x} - \vec{x}_j)$$

Basic controlling equations

Newton-Lorentz equation

$$\frac{d m_{i,e}\gamma_{i,e}\mathbf{v}_{i,e}}{dt} = q_{i,e}(\mathbf{E} + \mathbf{v}_{i,e} \times \mathbf{B})$$

Maxwell equation

$$\frac{\partial \mathbf{B}}{\partial t} = -\nabla \times \mathbf{E}$$

$$\frac{\partial \mathbf{E}}{\partial t} = c^2 \nabla \times \mathbf{B} - \frac{1}{\varepsilon_0}\mathbf{J}$$

$$\mathbf{J} = \sum (n_i q_i \mathbf{v}_i - n_e q_e \mathbf{v}_e)$$

Lorentz factor: $\gamma_{i,e} = (1 - \mathbf{v}_{i,e}^2 / c^2)^{-1/2}$

Fields

$$\frac{\partial \mathbf{B}}{\partial t} = -\nabla \times \mathbf{E} \qquad \frac{\partial \mathbf{D}}{\partial t} = \nabla \times \mathbf{H} - \mathbf{J}$$

$$\mathbf{B} = \mu_0 \mathbf{H} \qquad \mathbf{D} = \varepsilon_0 \mathbf{E}$$

$$\varepsilon_0 = 1 \qquad \mu_0 = 1/c^2$$

In Tristan code

$$\frac{\partial \mathbf{B}}{\partial t} = -c\nabla \times \mathbf{E} \qquad \frac{\partial \mathbf{E}}{\partial t} = c\nabla \times \mathbf{B} - \mathbf{J}$$

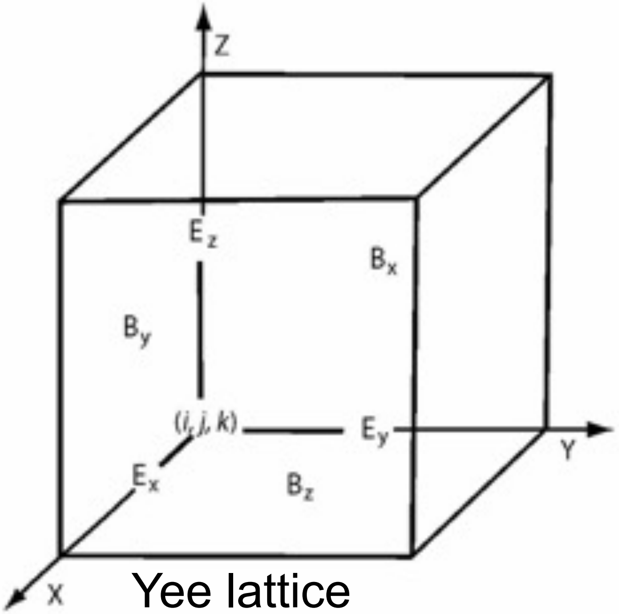
Two major methods of calculating current

1. Spectral method (UPIC code) (note by Decyk)
   We will review this method in details later after we do handout exercises

2. Charge-conserving current deposit (Villasenor & Buneman 1992)
   We will review this method with Umeda's method later (Umeda et al. 2003)

Ampere equation

$$\frac{\partial \mathrm{B}}{\partial t} = -c \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ \frac{\partial}{\partial x} & \frac{\partial}{\partial y} & \frac{\partial}{\partial z} \\ e_x & e_y & e_z \end{vmatrix} = c[\ \mathbf{i}(\frac{\partial e_y}{\partial z} - \frac{\partial e_z}{\partial y}) \\ + \mathbf{j}(\frac{\partial e_z}{\partial x} - \frac{\partial e_x}{\partial z}) \\ + \mathbf{k}(\frac{\partial e_x}{\partial y} - \frac{\partial e_y}{\partial x})]$$



Yee lattice

In Yee Lattice e_x, e_y, e_z, b_x, b_y, b_z are, respectively staggered and shifted on 0.5 from (*I, j, k*) and located at the position

$$\begin{aligned} e_x(i,j,k) &\rightarrow e_x(i+.5,j,k), \\ e_y(i,j,k) &\rightarrow e_y(i,j+.5,k), \\ e_z(i,j,k) &\rightarrow e_z(i,j,k+.5), \end{aligned}$$

and

$$\begin{aligned} b_x(i,j,k) &\rightarrow b_x(i,j+.5,k+.5), \\ b_y(i,j,k) &\rightarrow b_y(i+.5,j,k+.5), \\ b_z(i,j,k) &\rightarrow b_z(i+.5,j+.5,k). \end{aligned}$$

Field update

$$\begin{aligned}\frac{\partial}{\partial t}b_x &= (b_x^{new}(i,j+.5,k+.5) - b_x^{old}(i,j+.5,k+.5))/\delta t\\ &= c[(e_y(i,j+.5,k+1) - e_y(i,j+.5,k))/\delta z\\ &\quad -(e_z(i,j+1,k+.5) - e_z(i,j,k+.5))/\delta y].\end{aligned}$$

Here $\partial t = \partial x = \partial y = \partial z = 1$

$$\begin{aligned}b_x^{new}(i,j,k) &= b_x^{old}(i,j,k)\\ &\quad +c[\ e_y(i,j,k+1) - e_y(i,j,k) - e_z(i,j+1,k) + e_z(i,j,k)\ ].\end{aligned}$$

$$\begin{aligned}b_y^{new}(i,j,k) &= b_y^{old}(i,j,k)\\ &\quad +c[\ e_z(i+1,j,k) - e_z(i,j,k) - e_x(i,j,k+1) + e_x(i,j,k)\ ],\end{aligned}$$

$$\begin{aligned}b_z^{new}(i,j,k) &= b_z^{old}(i,j,k)\\ &\quad +c[\ e_x(i,j+1,k) - e_x(i,j,k) - e_y(i+1,j,k) + e_y(i,j,k)].\end{aligned}$$

```
c  first Maxwell-advance of the magnetic field by half a time-step
c        if(myid.eq.0) print *,'before b-field pusher’
       call B_field_push4(bx,by,bz,ex,ey,ez,mFx,mFy,mFz,DT,c,
     &   FBD_BLx,FBD_BRx,FBD_BLy,FBD_BRy,FBD_BLz,FBD_BRz,dims,coords)

c  second Maxwell-advance of the magnetic field by half a time-step
      if(myid.eq.0) print *,'before b-field pusher’
      call B_field_push4(bx,by,bz,ex,ey,ez,mFx,mFy,mFz,DT,c,
     &   FBD_BLx,FBD_BRx,FBD_BLy,FBD_BRy,FBD_BLz,FBD_BRz,dims,coords)

c  *****************************************************************
      subroutine B_field_push(bx,by,bz,ex,ey,ez,mFx,mFy,mFz,DT,c,
     &          FBD_BLx,FBD_BRx,FBD_BLy,FBD_BRy,FBD_BLz,FBD_BRz)

      integer FBD_BRx,FBD_BRy,FBD_BRz
      integer FBD_BLx,FBD_BLy,FBD_BLz

      dimension ex(mFx,mFy,mFz),ey(mFx,mFy,mFz),ez(mFx,mFy,mFz)
      dimension bx(mFx,mFy,mFz),by(mFx,mFy,mFz),bz(mFx,mFy,mFz)
```

```
   do k = FBD_BLz,FBD_BRz
   do j = FBD_BLy,FBD_BRy
   do i = FBD_BLx,FBD_BRx
              bx(i,j,k)=bx(i,j,k) + DT*(0.5*c)*
&            (ey(i,j,k+1)-ey(i,j,k)-ez(i,j+1,k)+ez(i,j,k))
             by(i,j,k)=by(i,j,k) + DT*(0.5*c)*
&            (ez(i+1,j,k)-ez(i,j,k)-ex(i,j,k+1)+ex(i,j,k))
              bz(i,j,k)=bz(i,j,k) + DT*(0.5*c)*
&            (ex(i,j+1,k)-ex(i,j,k)-ey(i+1,j,k)+ey(i,j,k))
   end do
   end do
   end do

  return
  end
c **********************************************************************
   subroutine B_field_push4(bx,by,bz,ex,ey,ez,mFx,mFy,mFz,DT,
&       c,FBD_BLx,FBD_BRx,FBD_BLy,FBD_BRy,FBD_BLz,FBD_BRz,
&                                  dims,coords)

  integer FBD_BRx,FBD_BRy,FBD_BRz
  integer FBD_BLx,FBD_BLy,FBD_BLz
  integer dims(3),coords(3)
  dimension ex(mFx,mFy,mFz),ey(mFx,mFy,mFz),ez(mFx,mFy,mFz)
  dimension bx(mFx,mFy,mFz),by(mFx,mFy,mFz),bz(mFx,mFy,mFz)

  if(dims(1).eq.1)then
  i=1
  do k = FBD_BLz,FBD_BRz
  do j = FBD_BLy,FBD_BRy
              bx(i,j,k)=bx(i,j,k) + DT*(0.5*c)*
&            (ey(i,j,k+1)-ey(i,j,k)-ez(i,j+1,k)+ez(i,j,k))
   by(i,j,k)=by(i,j,k) + DT*(0.5*c)*
&            (ez(i+1,j,k)-ez(i,j,k)-ex(i,j,k+1)+ex(i,j,k))
              bz(i,j,k)=bz(i,j,k) + DT*(0.5*c)*
&            (ex(i,j+1,k)-ex(i,j,k)-ey(i+1,j,k)+ey(i,j,k))
  end do
  end do

  do k = FBD_BLz,FBD_BRz
  do j = FBD_BLy,FBD_BRy
  do i = FBD_BLx+1,FBD_BRx-1
   bx(i,j,k)=bx(i,j,k) + DT*(0.5*c)*
&      (1.125*(ey(i,j,k+1)-ey(i,j,k)-ez(i,j+1,k)+ez(i,j,k))
&   -(ey(i,j,k+2)-ey(i,j,k-1)-ez(i,j+2,k)+ez(i,j-1,k))/24.)
```

```
   by(i,j,k)=by(i,j,k) + DT*(0.5*c)*
&      (1.125*(ez(i+1,j,k)-ez(i,j,k)-ex(i,j,k+1)+ex(i,j,k))
&   -(ez(i+2,j,k)-ez(i-1,j,k)-ex(i,j,k+2)+ex(i,j,k-1))/24.)
    bz(i,j,k)=bz(i,j,k) + DT*(0.5*c)*
&      (1.125*(ex(i,j+1,k)-ex(i,j,k)-ey(i+1,j,k)+ey(i,j,k))
&   -(ex(i,j+2,k)-ex(i,j-1,k)-ey(i+2,j,k)+ey(i-1,j,k))/24.)
  end do
  end do
  end do

  i=FBD_BRx
  do k = FBD_BLz,FBD_BRz
  do j = FBD_BLy,FBD_Bry
   bx(i,j,k)=bx(i,j,k) + DT*(0.5*c)*
&           (ey(i,j,k+1)-ey(i,j,k)-ez(i,j+1,k)+ez(i,j,k))
   by(i,j,k)=by(i,j,k) + DT*(0.5*c)*
&           (ez(i+1,j,k)-ez(i,j,k)-ex(i,j,k+1)+ex(i,j,k))
   bz(i,j,k)=bz(i,j,k) + DT*(0.5*c)*
&           (ex(i,j+1,k)-ex(i,j,k)-ey(i+1,j,k)+ey(i,j,k))
  end do
  end do
  else

  if(coords(1).eq.0)then
  i=1
  do k = FBD_BLz,FBD_BRz
  do j = FBD_BLy,FBD_BRy
           bx(i,j,k)=bx(i,j,k) + DT*(0.5*c)*
&          (ey(i,j,k+1)-ey(i,j,k)-ez(i,j+1,k)+ez(i,j,k))
   by(i,j,k)=by(i,j,k) + DT*(0.5*c)*
&          (ez(i+1,j,k)-ez(i,j,k)-ex(i,j,k+1)+ex(i,j,k))
           bz(i,j,k)=bz(i,j,k) + DT*(0.5*c)*
&          (ex(i,j+1,k)-ex(i,j,k)-ey(i+1,j,k)+ey(i,j,k))
  end do
  end do

  do k = FBD_BLz,FBD_BRz
  do j = FBD_BLy,FBD_BRy
  do i = FBD_BLx+1,FBD_BRx
   bx(i,j,k)=bx(i,j,k) + DT*(0.5*c)*
&     (1.125*(ey(i,j,k+1)-ey(i,j,k)-ez(i,j+1,k)+ez(i,j,k))
&   -(ey(i,j,k+2)-ey(i,j,k-1)-ez(i,j+2,k)+ez(i,j-1,k))/24.)
```

```
   by(i,j,k)=by(i,j,k) + DT*(0.5*c)*
&      (1.125*(ez(i+1,j,k)-ez(i,j,k)-ex(i,j,k+1)+ex(i,j,k))
&   -(ez(i+2,j,k)-ez(i-1,j,k)-ex(i,j,k+2)+ex(i,j,k-1))/24.)
   bz(i,j,k)=bz(i,j,k) + DT*(0.5*c)*
&      (1.125*(ex(i,j+1,k)-ex(i,j,k)-ey(i+1,j,k)+ey(i,j,k))
&   -(ex(i,j+2,k)-ex(i,j-1,k)-ey(i+2,j,k)+ey(i-1,j,k))/24.)
   end do
   end do
   end do

   else if(coords(1).eq.(dims(1)-1))then
   do k = FBD_BLz,FBD_BRz
   do j = FBD_BLy,FBD_BRy
   do i = FBD_BLx,FBD_BRx-1
    bx(i,j,k)=bx(i,j,k) + DT*(0.5*c)*
&      (1.125*(ey(i,j,k+1)-ey(i,j,k)-ez(i,j+1,k)+ez(i,j,k))
&   -(ey(i,j,k+2)-ey(i,j,k-1)-ez(i,j+2,k)+ez(i,j-1,k))/24.)
    by(i,j,k)=by(i,j,k) + DT*(0.5*c)*
&      (1.125*(ez(i+1,j,k)-ez(i,j,k)-ex(i,j,k+1)+ex(i,j,k))
&   -(ez(i+2,j,k)-ez(i-1,j,k)-ex(i,j,k+2)+ex(i,j,k-1))/24.)
```

```
  bz(i,j,k)=bz(i,j,k) + DT*(0.5*c)*
&     (1.125*(ex(i,j+1,k)-ex(i,j,k)-ey(i+1,j,k)+ey(i,j,k))
&   -(ex(i,j+2,k)-ex(i,j-1,k)-ey(i+2,j,k)+ey(i-1,j,k))/24.)
  end do
  end do
  end do

  i=FBD_BRx
  do k = FBD_BLz,FBD_BRz
  do j = FBD_BLy,FBD_Bry
   bx(i,j,k)=bx(i,j,k) + DT*(0.5*c)*
&           (ey(i,j,k+1)-ey(i,j,k)-ez(i,j+1,k)+ez(i,j,k))
   by(i,j,k)=by(i,j,k) + DT*(0.5*c)*
&           (ez(i+1,j,k)-ez(i,j,k)-ex(i,j,k+1)+ex(i,j,k))
   bz(i,j,k)=bz(i,j,k) + DT*(0.5*c)*
&           (ex(i,j+1,k)-ex(i,j,k)-ey(i+1,j,k)+ey(i,j,k))
  end do
  end do
  else
  do k = FBD_BLz,FBD_BRz
  do j = FBD_BLy,FBD_BRy
  do i = FBD_BLx,FBD_BRx
   bx(i,j,k)=bx(i,j,k) + DT*(0.5*c)*
&      (1.125*(ey(i,j,k+1)-ey(i,j,k)-ez(i,j+1,k)+ez(i,j,k))
&   -(ey(i,j,k+2)-ey(i,j,k-1)-ez(i,j+2,k)+ez(i,j-1,k))/24.)
   by(i,j,k)=by(i,j,k) + DT*(0.5*c)*
&      (1.125*(ez(i+1,j,k)-ez(i,j,k)-ex(i,j,k+1)+ex(i,j,k))
&   -(ez(i+2,j,k)-ez(i-1,j,k)-ex(i,j,k+2)+ex(i,j,k-1))/24.)
   bz(i,j,k)=bz(i,j,k) + DT*(0.5*c)*
&      (1.125*(ex(i,j+1,k)-ex(i,j,k)-ey(i+1,j,k)+ey(i,j,k))
&   -(ex(i,j+2,k)-ex(i,j-1,k)-ey(i+2,j,k)+ey(i-1,j,k))/24.)
  end do
  end do
  end do
  end if

  end if

  return
  end
```

Electric field update

$$\frac{\partial \mathbf{E}}{\partial t} = c \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ \frac{\partial}{\partial x} & \frac{\partial}{\partial y} & \frac{\partial}{\partial z} \\ b_x & b_y & b_z \end{vmatrix} \begin{aligned} &= c[\ \mathbf{i}(\frac{\partial b_z}{\partial y} - \frac{\partial b_y}{\partial z}) \\ &\quad + \mathbf{j}(\frac{\partial b_x}{\partial z} - \frac{\partial b_z}{\partial x}) \\ &\quad + \mathbf{k}(\frac{\partial b_y}{\partial x} - \frac{\partial b_x}{\partial y})] \end{aligned}$$

$$\begin{aligned} \frac{\partial}{\partial t} e_x &= (e_x^{new}(i+.5,j,k) - e_x^{old}(i+.5,j,k))/\delta t \\ &= c[(b_z(i+.5,j+.5,k) - b_z(i+.5,j-.5,k))/\delta y \\ &\quad - (b_y(i+.5,j,k+.5) - b_y(i+.5,j,k-.5))/\delta z], \end{aligned}$$

$$\begin{aligned} e_x^{new}(i,j,k) = {} & e_x^{old}(i,j,k) \\ & + c[\ b_y(i,j,k-1) - b_y(i,j,k) - b_z(i,j-1,k) + b_z(i,j,k)\ ], \end{aligned}$$

```
      call E_field_push4(bx,by,bz,ex,ey,ez,mFx,mFy,mFz,DT,c,
     &   FBD_ELx,FBD_ERx,FBD_ELy,FBD_ERy,FBD_ELz,FBD_ERz,dims,coords)

c  *************************************************************************
      subroutine E_field_push(bx,by,bz,ex,ey,ez,mFx,mFy,mFz,DT,c,
     &          FBD_ELx,FBD_ERx,FBD_ELy,FBD_ERy,FBD_ELz,FBD_ERz)

      integer FBD_ERx,FBD_ERy,FBD_ERz
      integer FBD_ELx,FBD_ELy,FBD_ELz

      dimension ex(mFx,mFy,mFz),ey(mFx,mFy,mFz),ez(mFx,mFy,mFz)
      dimension bx(mFx,mFy,mFz),by(mFx,mFy,mFz),bz(mFx,mFy,mFz)

       do k = FBD_ELz,FBD_ERz
       do j = FBD_ELy,FBD_ERy
       do i = FBD_ELx,FBD_ERx
        ex(i,j,k)=ex(i,j,k) + DT*c*
     &           (by(i,j,k-1)-by(i,j,k)-bz(i,j-1,k)+bz(i,j,k))
        ey(i,j,k)=ey(i,j,k) + DT*c*
     &           (bz(i-1,j,k)-bz(i,j,k)-bx(i,j,k-1)+bx(i,j,k))
```

```
   ez(i,j,k)=ez(i,j,k) + DT*c*
 &           (bx(i,j-1,k)-bx(i,j,k)-by(i-1,j,k)+by(i,j,k))
   end do
   end do
   end do

   return
   end
c  *********************************************************************
   subroutine E_field_push4(bx,by,bz,ex,ey,ez,mFx,mFy,mFz,DT,
 &       c,FBD_ELx,FBD_ERx,FBD_ELy,FBD_ERy,FBD_ELz,FBD_ERz,
 &                          dims,coords)

  integer FBD_ERx,FBD_ERy,FBD_ERz
  integer FBD_ELx,FBD_ELy,FBD_ELz
  integer dims(3),coords(3)

  dimension ex(mFx,mFy,mFz),ey(mFx,mFy,mFz),ez(mFx,mFy,mFz)
  dimension bx(mFx,mFy,mFz),by(mFx,mFy,mFz),bz(mFx,mFy,mFz)
```

```
   if(dims(1).eq.1) then
   i=2
   do k = FBD_ELz,FBD_ERz
   do j = FBD_ELy,FBD_ERy
   ex(i,j,k)=ex(i,j,k) + DT*c*
&          (by(i,j,k-1)-by(i,j,k)-bz(i,j-1,k)+bz(i,j,k))
   ey(i,j,k)=ey(i,j,k) + DT*c*
&          (bz(i-1,j,k)-bz(i,j,k)-bx(i,j,k-1)+bx(i,j,k))
   ez(i,j,k)=ez(i,j,k) + DT*c*
&          (bx(i,j-1,k)-bx(i,j,k)-by(i-1,j,k)+by(i,j,k))
  end do
  end do

  do k = FBD_ELz,FBD_ERz
  do j = FBD_ELy,FBD_ERy
  do i = FBD_ELx+1,FBD_ERx-1
   ex(i,j,k)=ex(i,j,k) + DT*c*
&      (1.125*(by(i,j,k-1)-by(i,j,k)-bz(i,j-1,k)+bz(i,j,k))
&   -(by(i,j,k-2)-by(i,j,k+1)-bz(i,j-2,k)+bz(i,j+1,k))/24.)
   ey(i,j,k)=ey(i,j,k) + DT*c*
&      (1.125*(bz(i-1,j,k)-bz(i,j,k)-bx(i,j,k-1)+bx(i,j,k))
&   -(bz(i-2,j,k)-bz(i+1,j,k)-bx(i,j,k-2)+bx(i,j,k+1))/24.)
```

```
   ez(i,j,k)=ez(i,j,k) + DT*c*
&       (1.125*(bx(i,j-1,k)-bx(i,j,k)-by(i-1,j,k)+by(i,j,k))
&    -(bx(i,j-2,k)-bx(i,j+1,k)-by(i-2,j,k)+by(i+1,j,k))/24.)
  end do
  end do
  end do

  i=FBD_ERx
  do k = FBD_ELz,FBD_ERz
  do j = FBD_ELy,FBD_ERy
   ex(i,j,k)=ex(i,j,k) + DT*c*
&           (by(i,j,k-1)-by(i,j,k)-bz(i,j-1,k)+bz(i,j,k))
   ey(i,j,k)=ey(i,j,k) + DT*c*
&           (bz(i-1,j,k)-bz(i,j,k)-bx(i,j,k-1)+bx(i,j,k))
   ez(i,j,k)=ez(i,j,k) + DT*c*
&           (bx(i,j-1,k)-bx(i,j,k)-by(i-1,j,k)+by(i,j,k))
  end do
  end do

  else
```

```
   if(coords(1).eq.0)then
   i=2
   do k = FBD_ELz,FBD_ERz
   do j = FBD_ELy,FBD_ERy
    ex(i,j,k)=ex(i,j,k) + DT*c*
&            (by(i,j,k-1)-by(i,j,k)-bz(i,j-1,k)+bz(i,j,k))
    ey(i,j,k)=ey(i,j,k) + DT*c*
&            (bz(i-1,j,k)-bz(i,j,k)-bx(i,j,k-1)+bx(i,j,k))
    ez(i,j,k)=ez(i,j,k) + DT*c*
&            (bx(i,j-1,k)-bx(i,j,k)-by(i-1,j,k)+by(i,j,k))
   end do
   end do

   do k = FBD_ELz,FBD_ERz
   do j = FBD_ELy,FBD_ERy
   do i = FBD_ELx+1,FBD_ERx
    ex(i,j,k)=ex(i,j,k) + DT*c*
&       (1.125*(by(i,j,k-1)-by(i,j,k)-bz(i,j-1,k)+bz(i,j,k))
&    -(by(i,j,k-2)-by(i,j,k+1)-bz(i,j-2,k)+bz(i,j+1,k))/24.)
    ey(i,j,k)=ey(i,j,k) + DT*c*
&       (1.125*(bz(i-1,j,k)-bz(i,j,k)-bx(i,j,k-1)+bx(i,j,k))
&    -(bz(i-2,j,k)-bz(i+1,j,k)-bx(i,j,k-2)+bx(i,j,k+1))/24.)
```

```
  ez(i,j,k)=ez(i,j,k) + DT*c*
&       (1.125*(bx(i,j-1,k)-bx(i,j,k)-by(i-1,j,k)+by(i,j,k))
&    -(bx(i,j-2,k)-bx(i,j+1,k)-by(i-2,j,k)+by(i+1,j,k))/24.)
  end do
  end do
  end do
  else if(coords(1).eq.(dims(1)-1))then
  do k = FBD_ELz,FBD_ERz
  do j = FBD_ELy,FBD_ERy
  do i = FBD_ELx,FBD_ERx-1
   ex(i,j,k)=ex(i,j,k) + DT*c*
&       (1.125*(by(i,j,k-1)-by(i,j,k)-bz(i,j-1,k)+bz(i,j,k))
&    -(by(i,j,k-2)-by(i,j,k+1)-bz(i,j-2,k)+bz(i,j+1,k))/24.)
   ey(i,j,k)=ey(i,j,k) + DT*c*
&       (1.125*(bz(i-1,j,k)-bz(i,j,k)-bx(i,j,k-1)+bx(i,j,k))
&    -(bz(i-2,j,k)-bz(i+1,j,k)-bx(i,j,k-2)+bx(i,j,k+1))/24.)
   ez(i,j,k)=ez(i,j,k) + DT*c*
&       (1.125*(bx(i,j-1,k)-bx(i,j,k)-by(i-1,j,k)+by(i,j,k))
&    -(bx(i,j-2,k)-bx(i,j+1,k)-by(i-2,j,k)+by(i+1,j,k))/24.)
  end do
  end do
  end do
  i=FBD_ERx
  do k = FBD_ELz,FBD_ERz
  do j = FBD_ELy,FBD_ERy
   ex(i,j,k)=ex(i,j,k) + DT*c*
&          (by(i,j,k-1)-by(i,j,k)-bz(i,j-1,k)+bz(i,j,k))
   ey(i,j,k)=ey(i,j,k) + DT*c*
&          (bz(i-1,j,k)-bz(i,j,k)-bx(i,j,k-1)+bx(i,j,k))
   ez(i,j,k)=ez(i,j,k) + DT*c*
&          (bx(i,j-1,k)-bx(i,j,k)-by(i-1,j,k)+by(i,j,k))
  end do
  end do

  else
  do k = FBD_ELz,FBD_ERz
  do j = FBD_ELy,FBD_ERy
  do i = FBD_ELx,FBD_ERx
   ex(i,j,k)=ex(i,j,k) + DT*c*
&      (1.125*(by(i,j,k-1)-by(i,j,k)-bz(i,j-1,k)+bz(i,j,k))
&   -(by(i,j,k-2)-by(i,j,k+1)-bz(i,j-2,k)+bz(i,j+1,k))/24.)
   ey(i,j,k)=ey(i,j,k) + DT*c*
&      (1.125*(bz(i-1,j,k)-bz(i,j,k)-bx(i,j,k-1)+bx(i,j,k))
&   -(bz(i-2,j,k)-bz(i+1,j,k)-bx(i,j,k-2)+bx(i,j,k+1))/24.)
```

```
  ez(i,j,k)=ez(i,j,k) + DT*c*
&       (1.125*(bx(i,j-1,k)-bx(i,j,k)-by(i-1,j,k)+by(i,j,k))
&    -(bx(i,j-2,k)-bx(i,j+1,k)-by(i-2,j,k)+by(i+1,j,k))/24.)
  end do
  end do
  end do
  end if

  end if

  return
  end
```

Particle update

Newton-Lorentz equation

$$\mathbf{v}^{new} - \mathbf{v}^{old} = \frac{q\delta t}{m} < \mathbf{E} + \frac{1}{2}(\mathbf{v}^{new} + \mathbf{v}^{old}) \times \mathbf{B} >$$

$$\mathbf{r}^{next} - \mathbf{r}^{present} = \delta t \mathbf{v}^{new}$$

Buneman-Boris method

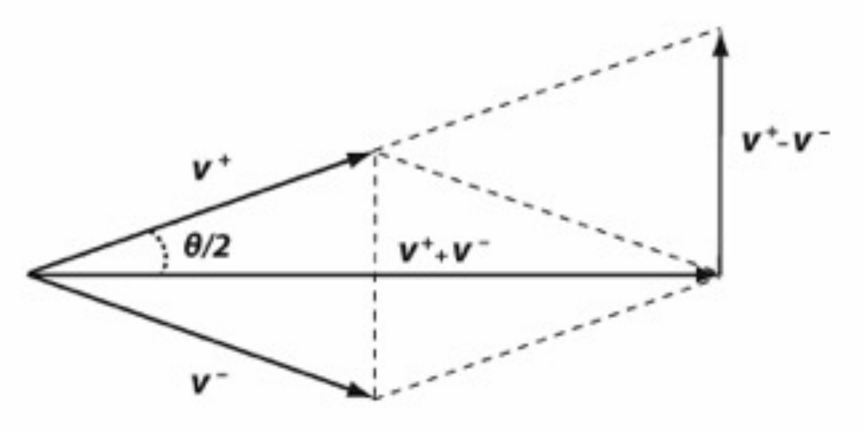
Half an electric acceleration

Pure magnetic rotation

Another half electric acceleration

Buneman-Boris method

$$\frac{\mathbf{v}^{n+1/2} - \mathbf{v}^{n-1/2}}{\Delta t} = \frac{q}{m}\left(\mathbf{E}^n + \frac{\mathbf{v}^{n+1/2} + \mathbf{v}^{n-1/2}}{2} \times \mathbf{B}^n\right)$$



$$\mathbf{v}^- = \mathbf{v}^{n-1/2} + \frac{q}{m}\mathbf{E}^n \frac{\Delta t}{2}$$

$$\mathbf{v}^+ = \mathbf{v}^{n+1/2} - \frac{q}{m}\mathbf{E}^n \frac{\Delta t}{2}$$

rotation $$\frac{\mathbf{v}^+ - \mathbf{v}^-}{\Delta t} = \frac{1}{2}\frac{q}{m}(\mathbf{v}^+ + \mathbf{v}^-) \times \mathbf{B}^n$$

$$\mathbf{v}^+ = \mathbf{v}^- + \frac{2}{1+\mathbf{T}^2}(\mathbf{v}^- + \mathbf{v}^- \times \mathbf{T}) \times \mathbf{T}$$

$$\mathbf{T} = \frac{q}{2m}\Delta t \mathbf{B}^n$$

Buneman-Boris method (cont)

4 steps

$$\mathbf{v}^- = \mathbf{v}^{n-1/2} + \frac{q}{m}\mathbf{E}^n\frac{\Delta t}{2}$$

$$\mathbf{v}^0 = \mathbf{v}^- + \mathbf{v}^- \times \mathbf{T}$$

$$\mathbf{v}^+ = \mathbf{v}^- + \mathbf{v}^0 \times \mathbf{S}$$

$$\mathbf{S} = 2\mathbf{T}/(\mathbf{1} + \mathbf{T}^2)$$

$$\mathbf{v}^{n+1/2} = \mathbf{v}^+ + \frac{q}{m}\mathbf{E}^n\frac{\Delta t}{2}$$

Homework: derive this **S**

$$\mathbf{r}^{n+1} = \mathbf{r}^n + \mathbf{v}^{n+1/2}\Delta t$$

Relativistic generalization

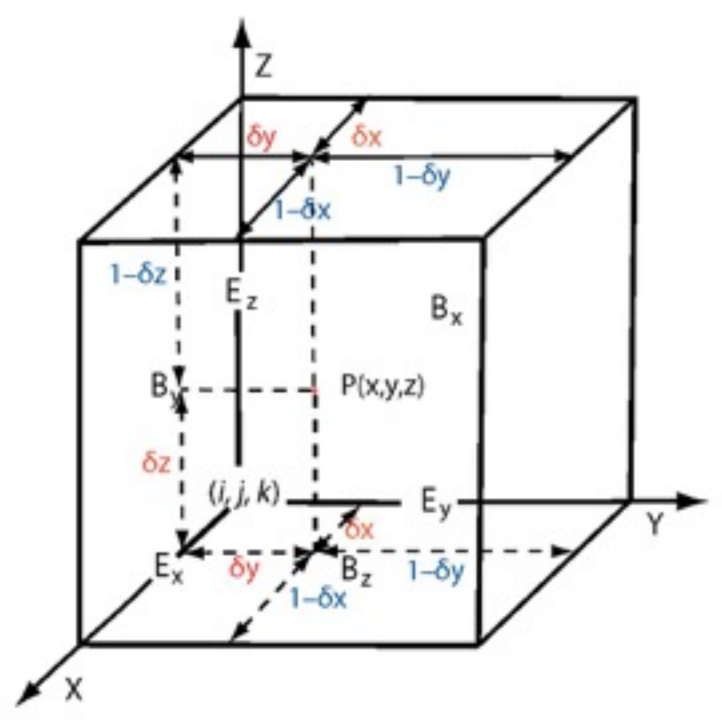
$$\mathbf{u} = \gamma_v \mathbf{v}, \quad \gamma_v^2 = \left(1 - \frac{v^2}{c^2}\right)^{-1} \qquad \gamma^2 = \left(1 + \frac{u^2}{c^2}\right)$$

$$\frac{\mathbf{u}^{n+1/2} - \mathbf{u}^{n-1/2}}{\Delta t} = \frac{q}{m}\left(\mathbf{E}^n + \frac{\mathbf{u}^{n+1/2} + \mathbf{u}^{n-1/2}}{2\gamma^n} \times \mathbf{B}^n\right)$$

$$\mathbf{r}^{n+1} = \mathbf{r}^n + \mathbf{v}^{n+1/2}\Delta t = \mathbf{r}^n + \frac{\mathbf{u}^{n+1/2}}{\gamma^{n+1/2}}\Delta t$$

$$\left(\gamma^{n+1/2}\right)^2 = 1 + \left(\frac{u^{n+1/2}}{c}\right)^2$$

Force interpretations



"volume" weight

$$(i, j, k) \Leftarrow (1-\delta x)(1-\delta y)(1-\delta z) = cx \circlearrowright cy \circlearrowright cz$$

$$(i+1, j+1, k+1) \Leftarrow \delta x \circlearrowright \delta y \circlearrowright \delta z$$

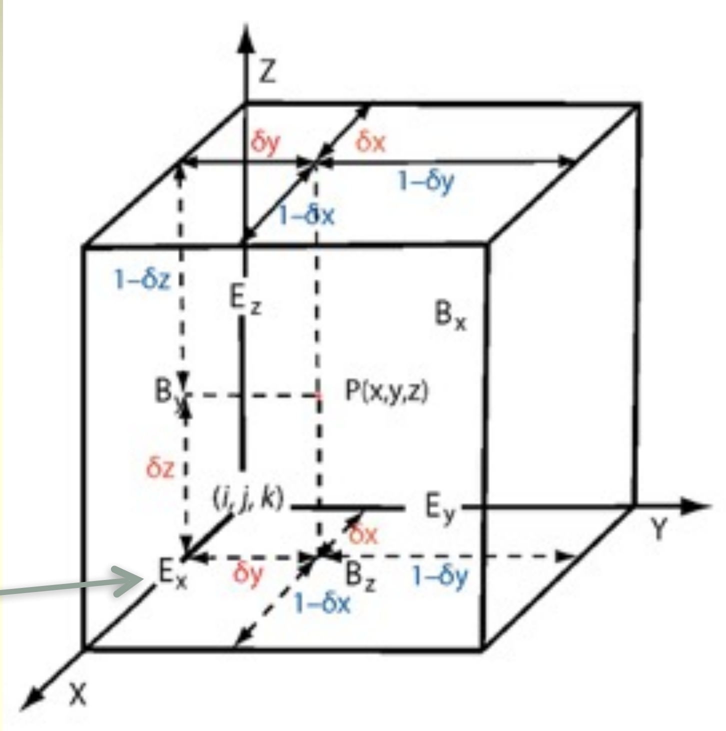
$$\mathbf{F}_{e_x}^{(x,j,k)} = \bar{e}_x(i,j,k) + \left[\bar{e}_x(i+1,j,k) - \bar{e}_x(i,j,k)\right]\delta x$$

$$\bar{e}_x(i,j,k) = \frac{1}{2}\left\{e_x(i,j,k) + e_x(i-1,j,k)\right\} \qquad \bar{e}_x(i+1,j,k) = \frac{1}{2}\left\{e_x(i+1,j,k) + e_x(i,j,k)\right\}$$

on (*x*, *j*, *k*)

$$2\mathbf{F}_{e_x}^{(x,j,k)} = e_x(i,j,k) + e_x(i-1,j,k) + \left[e_x(i+1,j,k) - e_x(i-1,j,k)\right]\delta x$$

Force interpretations



“volume” weight

$$(i,j,k) \Leftarrow (1-\delta x)(1-\delta y)(1-\delta z) = cx \cdot cy \cdot cz$$

$$(i+1,j+1,k+1) \Leftarrow \delta x \cdot \delta y \cdot \delta z$$

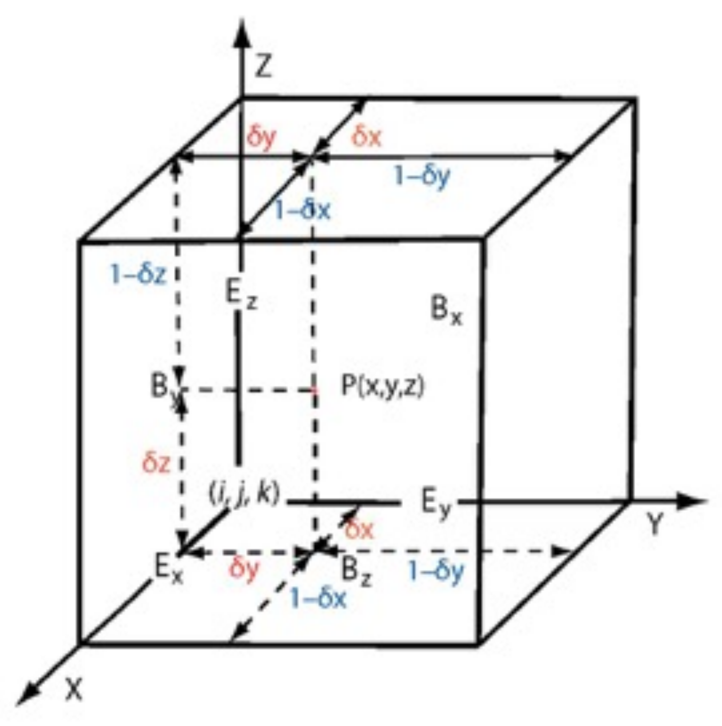
$$\mathbf{F}_{e_x}^{(x,j,k)} = \bar{e}_x(i,j,k) + \left[\bar{e}_x(i+1,j,k) - \bar{e}_x(i,j,k)\right]\delta x$$

$$\bar{e}_x(i,j,k) = \frac{1}{2}\left\{e_x(i,j,k) + e_x(i-1,j,k)\right\} \qquad \bar{e}_x(i+1,j,k) = \frac{1}{2}\left\{e_x(i+1,j,k) + e_x(i,j,k)\right\}$$

on (x, j, k)

$$2\mathbf{F}_{e_x}^{(x,j,k)} = e_x(i,j,k) + e_x(i-1,j,k) + \left[e_x(i+1,j,k) - e_x(i-1,j,k)\right]\delta x$$

Force interpretations



“volume” weight

$$(i, j, k) \Leftarrow (1-\delta x)(1-\delta y)(1-\delta z) = cx \circlearrowright cy \circlearrowright cz$$

$$(i+1, j+1, k+1) \Leftarrow \delta x \circlearrowright \delta y \circlearrowright \delta z$$

$$\mathbf{F}_{e_x}^{(x,j,k)} = \bar{e}_x(i, j, k) + \left[\bar{e}_x(i+1, j, k) - \bar{e}_x(i, j, k)\right]\delta x$$

$$\bar{e}_x(i, j, k) = \frac{1}{2}\left\{e_x(i, j, k) + e_x(i-1, j, k)\right\} \qquad \bar{e}_x(i+1, j, k) = \frac{1}{2}\left\{e_x(i+1, j, k) + e_x(i, j, k)\right\}$$

on (x, j, k)

$$2\mathbf{F}_{e_x}^{(x,j,k)} = e_x(i, j, k) + e_x(i-1, j, k) + \left[e_x(i+1, j, k) - e_x(i-1, j, k)\right]\delta x$$

similarly on (x, j+1, k), (x, j, k+1), (x, j+1, k+1)

$$2\mathbf{F}_{e_x}^{(x,j+1,k)} = e_x(i,j+1,k)+e_x(i-1,j+1,k)+\left[e_x(i+1,j+1,k)-e_x(i-1,j+1,k)\right]\delta x$$

$$2\mathbf{F}_{e_x}^{(x,j,k+1)} = e_x(i,j,k+1)+e_x(i-1,j,k+1)+\left[e_x(i+1,j,k+1)-e_x(i-1,j,k+1)\right]\delta x$$

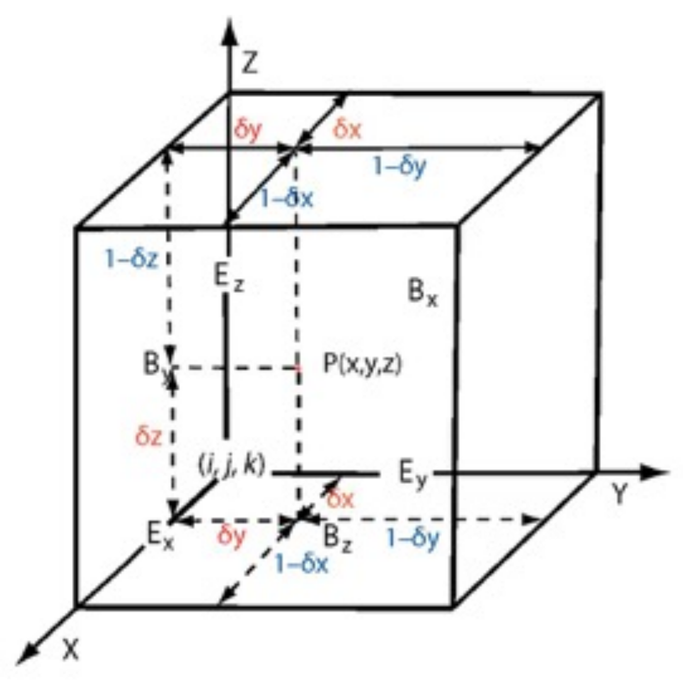
$$2\mathbf{F}_{e_x}^{(x,j+1,k+1)} = e_x(i,j+1,k+1)+e_x(i-1,j+1,k+1)+\left[e_x(i+1,j+1,k+1)-e_x(i-1,j+1,k+1)\right]\delta x$$

$$\mathbf{F}_{e_x}^{(x,y,k)} = \mathbf{F}_{e_x}^{(x,j,k)} + \left[\mathbf{F}_{e_x}^{(x,j+1,k)} - \mathbf{F}_{e_x}^{(x,j,k)}\right]\delta y$$

$$\mathbf{F}_{e_x}^{(x,y,k+1)} = \mathbf{F}_{e_x}^{(x,j,k+1)} + \left[\mathbf{F}_{e_x}^{(x,j+1,k+1)} - \mathbf{F}_{e_x}^{(x,j,k+1)}\right]\delta y$$

$$\mathbf{F}_{e_x}^{(x,y,z)} = \mathbf{F}_{e_x}^{(x,y,k)} + \left[\mathbf{F}_{e_x}^{(x,y,k+1)} - \mathbf{F}_{e_x}^{(x,y,k)}\right]\delta z$$

$$\mathbf{F}_{e_y}^{(x,y,z)},\ \mathbf{F}_{e_z}^{(x,y,z)},\ \mathbf{F}_{b_x}^{(x,y,z)},\ \mathbf{F}_{b_y}^{(x,y,z)},\ \mathbf{F}_{b_z}^{(x,y,z)}$$



similarly on (x, j+1, k), (x, j, k+1), (x, j+1, k+1)

$$2\mathbf{F}_{e_x}^{(x,j+1,k)} = e_x(i,j+1,k) + e_x(i-1,j+1,k) + \left[e_x(i+1,j+1,k) - e_x(i-1,j+1,k)\right]\delta x$$

$$2\mathbf{F}_{e_x}^{(x,j,k+1)} = e_x(i,j,k+1) + e_x(i-1,j,k+1) + \left[e_x(i+1,j,k+1) - e_x(i-1,j,k+1)\right]\delta x$$

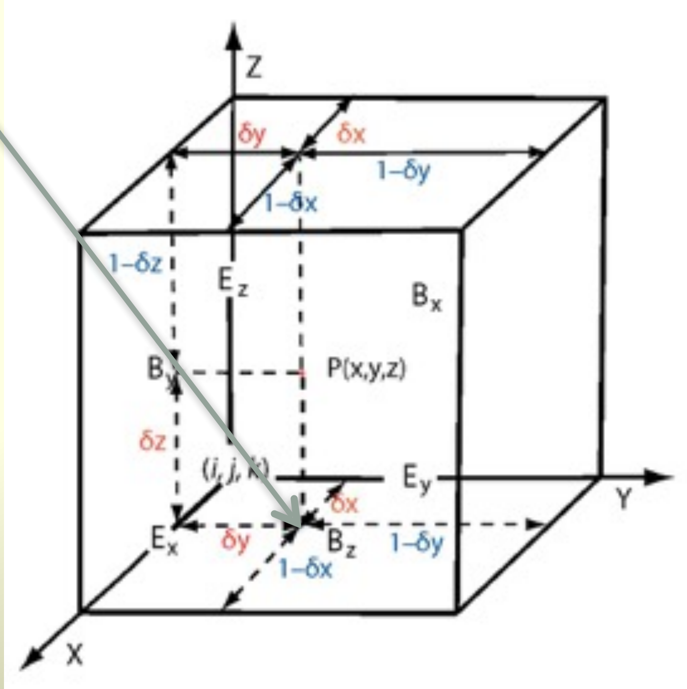
$$2\mathbf{F}_{e_x}^{(x,j+1,k+1)} = e_x(i,j+1,k+1) + e_x(i-1,j+1,k+1) + \left[e_x(i+1,j+1,k+1) - e_x(i-1,j+1,k+1)\right]\delta x$$

$$\mathbf{F}_{e_x}^{(x,y,k)} = \mathbf{F}_{e_x}^{(x,j,k)} + \left[\mathbf{F}_{e_x}^{(x,j+1,k)} - \mathbf{F}_{e_x}^{(x,j,k)}\right]\delta y$$

$$\mathbf{F}_{e_x}^{(x,y,k+1)} = \mathbf{F}_{e_x}^{(x,j,k+1)} + \left[\mathbf{F}_{e_x}^{(x,j+1,k+1)} - \mathbf{F}_{e_x}^{(x,j,k+1)}\right]\delta y$$

$$\mathbf{F}_{e_x}^{(x,y,z)} = \mathbf{F}_{e_x}^{(x,y,k)} + \left[\mathbf{F}_{e_x}^{(x,y,k+1)} - \mathbf{F}_{e_x}^{(x,y,k)}\right]\delta z$$

$$\mathbf{F}_{e_y}^{(x,y,z)},\ \mathbf{F}_{e_z}^{(x,y,z)},\ \mathbf{F}_{b_x}^{(x,y,z)},\ \mathbf{F}_{b_y}^{(x,y,z)},\ \mathbf{F}_{b_z}^{(x,y,z)}$$



similarly on (x, j+1, k), (x, j, k+1), (x, j+1, k+1)

$$2\mathbf{F}_{e_x}^{(x,j+1,k)} = e_x(i,j+1,k) + e_x(i-1,j+1,k) + \left[e_x(i+1,j+1,k) - e_x(i-1,j+1,k)\right]\delta x$$

$$2\mathbf{F}_{e_x}^{(x,j,k+1)} = e_x(i,j,k+1) + e_x(i-1,j,k+1) + \left[e_x(i+1,j,k+1) - e_x(i-1,j,k+1)\right]\delta x$$

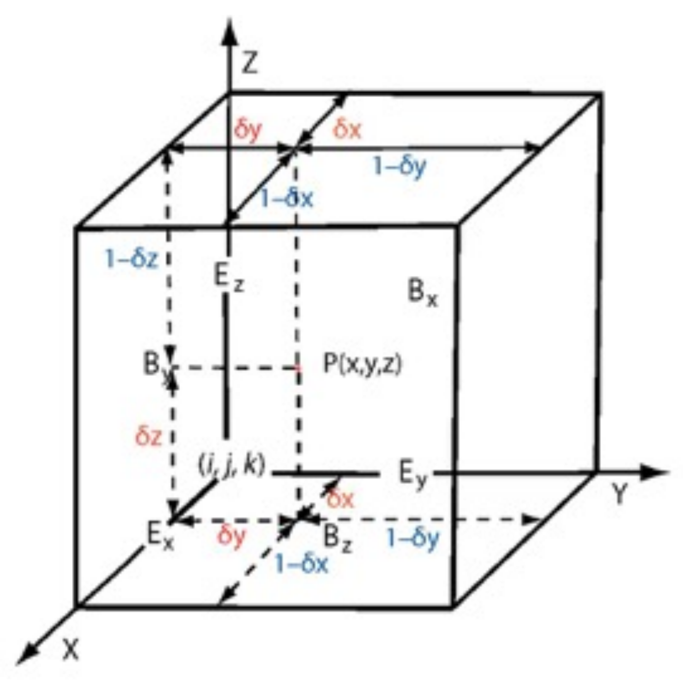
$$2\mathbf{F}_{e_x}^{(x,j+1,k+1)} = e_x(i,j+1,k+1) + e_x(i-1,j+1,k+1) + \left[e_x(i+1,j+1,k+1) - e_x(i-1,j+1,k+1)\right]\delta x$$

$$\mathbf{F}_{e_x}^{(x,y,k)} = \mathbf{F}_{e_x}^{(x,j,k)} + \left[\mathbf{F}_{e_x}^{(x,j+1,k)} - \mathbf{F}_{e_x}^{(x,j,k)}\right]\delta y$$

$$\mathbf{F}_{e_x}^{(x,y,k+1)} = \mathbf{F}_{e_x}^{(x,j,k+1)} + \left[\mathbf{F}_{e_x}^{(x,j+1,k+1)} - \mathbf{F}_{e_x}^{(x,j,k+1)}\right]\delta y$$

$$\mathbf{F}_{e_x}^{(x,y,z)} = \mathbf{F}_{e_x}^{(x,y,k)} + \left[\mathbf{F}_{e_x}^{(x,y,k+1)} - \mathbf{F}_{e_x}^{(x,y,k)}\right]\delta z$$

$$\mathbf{F}_{e_y}^{(x,y,z)},\ \mathbf{F}_{e_z}^{(x,y,z)},\ \mathbf{F}_{b_x}^{(x,y,z)},\ \mathbf{F}_{b_y}^{(x,y,z)},\ \mathbf{F}_{b_z}^{(x,y,z)}$$



similarly on (x, *j*+1, *k*), (*x*, *j*, *k*+1), (*x*, *j*+1, *k*+1)

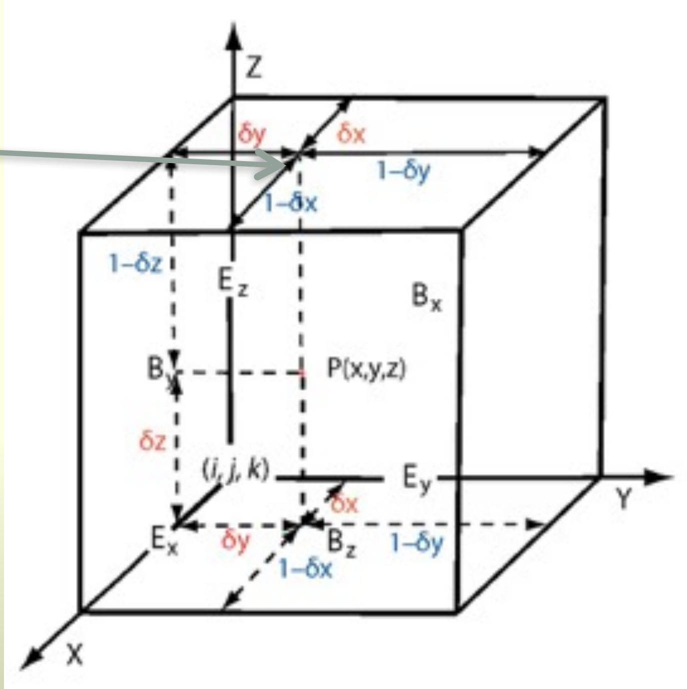
$$2\mathbf{F}_{e_x}^{(x,j+1,k)} = e_x(i, j+1, k) + e_x(i-1, j+1, k) + \left[e_x(i+1, j+1, k) - e_x(i-1, j+1, k) \right] \delta x$$

$$2\mathbf{F}_{e_x}^{(x,j,k+1)} = e_x(i, j, k+1) + e_x(i-1, j, k+1) + \left[e_x(i+1, j, k+1) - e_x(i-1, j, k+1) \right] \delta x$$

$$2\mathbf{F}_{e_x}^{(x,j+1,k+1)} = e_x(i, j+1, k+1) + e_x(i-1, j+1, k+1) + \left[e_x(i+1, j+1, k+1) - e_x(i-1, j+1, k+1) \right] \delta x$$

$$\mathbf{F}_{e_x}^{(x,y,k)} = \mathbf{F}_{e_x}^{(x,j,k)} + \left[\mathbf{F}_{e_x}^{(x,j+1,k)} - \mathbf{F}_{e_x}^{(x,j,k)} \right] \delta y$$

$$\mathbf{F}_{e_x}^{(x,y,k+1)} = \mathbf{F}_{e_x}^{(x,j,k+1)} + \left[\mathbf{F}_{e_x}^{(x,j+1,k+1)} - \mathbf{F}_{e_x}^{(x,j,k+1)} \right] \delta y$$



$$\mathbf{F}_{e_x}^{(x,y,z)} = \mathbf{F}_{e_x}^{(x,y,k)} + \left[\mathbf{F}_{e_x}^{(x,y,k+1)} - \mathbf{F}_{e_x}^{(x,y,k)} \right] \delta z$$

$$\mathbf{F}_{e_y}^{(x,y,z)},\ \mathbf{F}_{e_z}^{(x,y,z)},\ \mathbf{F}_{b_x}^{(x,y,z)},\ \mathbf{F}_{b_y}^{(x,y,z)},\ \mathbf{F}_{b_z}^{(x,y,z)}$$

similarly on (x, *j*+1, *k*), (*x*, *j*, *k*+1), (*x*, *j*+1, *k*+1)

$$2\mathbf{F}_{e_x}^{(x,j+1,k)} = e_x(i,j+1,k) + e_x(i-1,j+1,k) + \left[e_x(i+1,j+1,k) - e_x(i-1,j+1,k)\right]\delta x$$

$$2\mathbf{F}_{e_x}^{(x,j,k+1)} = e_x(i,j,k+1) + e_x(i-1,j,k+1) + \left[e_x(i+1,j,k+1) - e_x(i-1,j,k+1)\right]\delta x$$

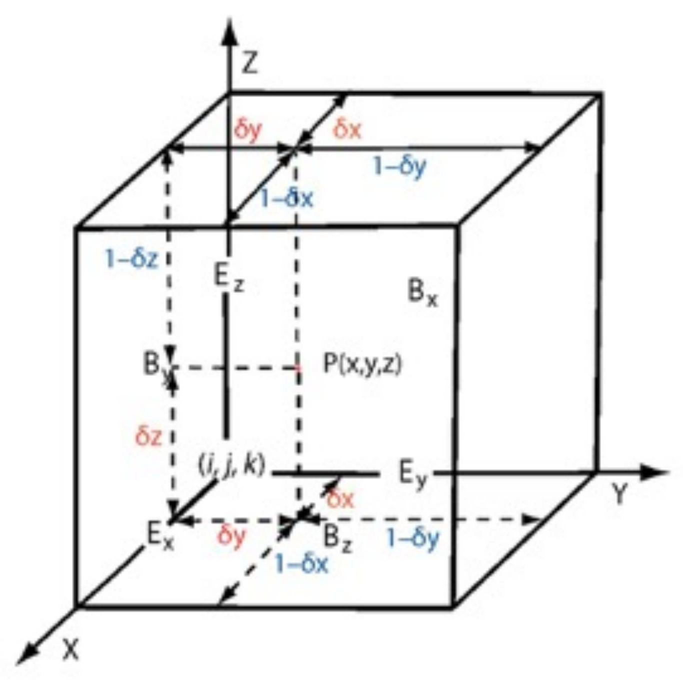
$$2\mathbf{F}_{e_x}^{(x,j+1,k+1)} = e_x(i,j+1,k+1) + e_x(i-1,j+1,k+1) + \left[e_x(i+1,j+1,k+1) - e_x(i-1,j+1,k+1)\right]\delta x$$

$$\mathbf{F}_{e_x}^{(x,y,k)} = \mathbf{F}_{e_x}^{(x,j,k)} + \left[\mathbf{F}_{e_x}^{(x,j+1,k)} - \mathbf{F}_{e_x}^{(x,j,k)}\right]\delta y$$

$$\mathbf{F}_{e_x}^{(x,y,k+1)} = \mathbf{F}_{e_x}^{(x,j,k+1)} + \left[\mathbf{F}_{e_x}^{(x,j+1,k+1)} - \mathbf{F}_{e_x}^{(x,j,k+1)}\right]\delta y$$

$$\mathbf{F}_{e_x}^{(x,y,z)} = \mathbf{F}_{e_x}^{(x,y,k)} + \left[\mathbf{F}_{e_x}^{(x,y,k+1)} - \mathbf{F}_{e_x}^{(x,y,k)}\right]\delta z$$

$$\mathbf{F}_{e_y}^{(x,y,z)},\ \mathbf{F}_{e_z}^{(x,y,z)},\ \mathbf{F}_{b_x}^{(x,y,z)},\ \mathbf{F}_{b_y}^{(x,y,z)},\ \mathbf{F}_{b_z}^{(x,y,z)}$$



similarly on (x, j+1, k), (x, j, k+1), (x, j+1, k+1)

$$2\mathbf{F}_{e_x}^{(x,j+1,k)} = e_x(i,j+1,k)+e_x(i-1,j+1,k)+\left[e_x(i+1,j+1,k)-e_x(i-1,j+1,k)\right]\delta x$$

$$2\mathbf{F}_{e_x}^{(x,j,k+1)} = e_x(i,j,k+1)+e_x(i-1,j,k+1)+\left[e_x(i+1,j,k+1)-e_x(i-1,j,k+1)\right]\delta x$$

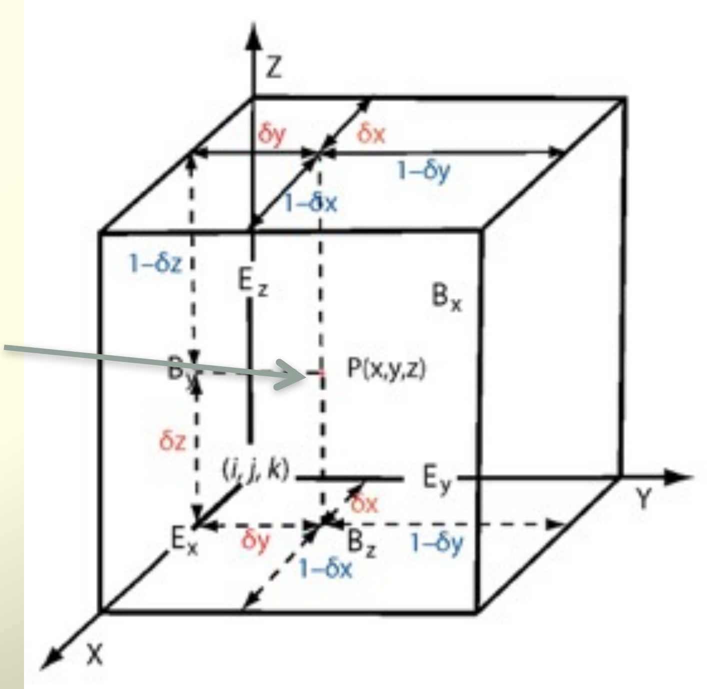
$$2\mathbf{F}_{e_x}^{(x,j+1,k+1)} = e_x(i,j+1,k+1)+e_x(i-1,j+1,k+1)+\left[e_x(i+1,j+1,k+1)-e_x(i-1,j+1,k+1)\right]\delta x$$

$$\mathbf{F}_{e_x}^{(x,y,k)} = \mathbf{F}_{e_x}^{(x,j,k)} + \left[\mathbf{F}_{e_x}^{(x,j+1,k)} - \mathbf{F}_{e_x}^{(x,j,k)}\right]\delta y$$

$$\mathbf{F}_{e_x}^{(x,y,k+1)} = \mathbf{F}_{e_x}^{(x,j,k+1)} + \left[\mathbf{F}_{e_x}^{(x,j+1,k+1)} - \mathbf{F}_{e_x}^{(x,j,k+1)}\right]\delta y$$

$$\mathbf{F}_{e_x}^{(x,y,z)} = \mathbf{F}_{e_x}^{(x,y,k)} + \left[\mathbf{F}_{e_x}^{(x,y,k+1)} - \mathbf{F}_{e_x}^{(x,y,k)}\right]\delta z$$

$$\mathbf{F}_{e_y}^{(x,y,z)},\ \mathbf{F}_{e_z}^{(x,y,z)},\ \mathbf{F}_{b_x}^{(x,y,z)},\ \mathbf{F}_{b_y}^{(x,y,z)},\ \mathbf{F}_{b_z}^{(x,y,z)}$$



similarly on (x, j+1, k), (x, j, k+1), (x, j+1, k+1)

$$2\mathbf{F}_{e_x}^{(x,j+1,k)} = e_x(i,j+1,k) + e_x(i-1,j+1,k) + \left[e_x(i+1,j+1,k) - e_x(i-1,j+1,k)\right]\delta x$$

$$2\mathbf{F}_{e_x}^{(x,j,k+1)} = e_x(i,j,k+1) + e_x(i-1,j,k+1) + \left[e_x(i+1,j,k+1) - e_x(i-1,j,k+1)\right]\delta x$$

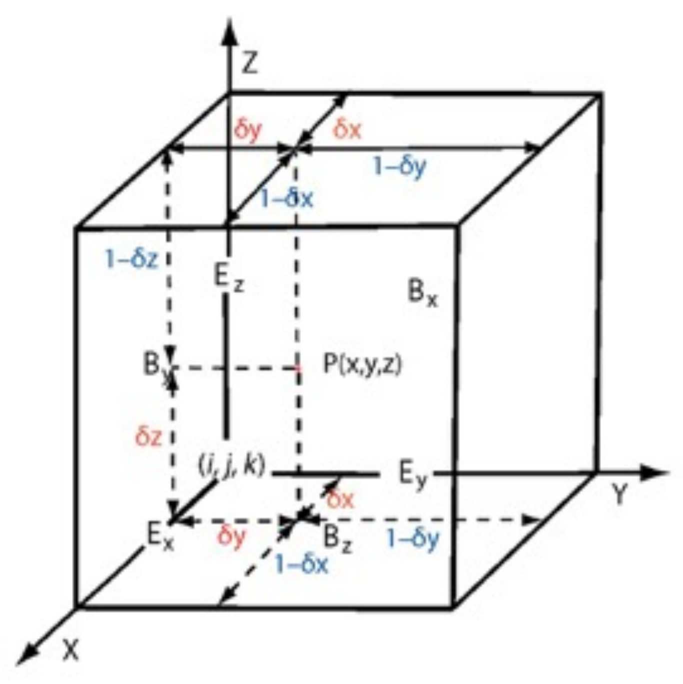
$$2\mathbf{F}_{e_x}^{(x,j+1,k+1)} = e_x(i,j+1,k+1) + e_x(i-1,j+1,k+1) + \left[e_x(i+1,j+1,k+1) - e_x(i-1,j+1,k+1)\right]\delta x$$

$$\mathbf{F}_{e_x}^{(x,y,k)} = \mathbf{F}_{e_x}^{(x,j,k)} + \left[\mathbf{F}_{e_x}^{(x,j+1,k)} - \mathbf{F}_{e_x}^{(x,j,k)}\right]\delta y$$

$$\mathbf{F}_{e_x}^{(x,y,k+1)} = \mathbf{F}_{e_x}^{(x,j,k+1)} + \left[\mathbf{F}_{e_x}^{(x,j+1,k+1)} - \mathbf{F}_{e_x}^{(x,j,k+1)}\right]\delta y$$

$$\mathbf{F}_{e_x}^{(x,y,z)} = \mathbf{F}_{e_x}^{(x,y,k)} + \left[\mathbf{F}_{e_x}^{(x,y,k+1)} - \mathbf{F}_{e_x}^{(x,y,k)}\right]\delta z$$

$$\mathbf{F}_{e_y}^{(x,y,z)},\ \mathbf{F}_{e_z}^{(x,y,z)},\ \mathbf{F}_{b_x}^{(x,y,z)},\ \mathbf{F}_{b_y}^{(x,y,z)},\ \mathbf{F}_{b_z}^{(x,y,z)}$$



```
c  MOVE PARTICLES
c  ambient ions and electrons
c       if(myid.eq.0) print *, 'before mover ambient'
      call mover(ions,xi,yi,zi,ui,vi,wi,mb,ex,ey,ez,bx,by,bz,
    &                   mFx,mFy,mFz,DHDx,DHDy,DHDz,qmi,DT,c)
      call mover(lecs,xe,ye,ze,ue,ve,we,mb,ex,ey,ez,bx,by,bz,
    &                   mFx,mFy,mFz,DHDx,DHDy,DHDz,qme,DT,c)
cU1       call mover2(ions,xi,yi,zi,ui,vi,wi,mb,ex,ey,ez,bx,by,bz,
cU1     &                   mFx,mFy,mFz,DHDx,DHDy,DHDz,qmi,DT,c)
cU1       call mover2(lecs,xe,ye,ze,ue,ve,we,mb,ex,ey,ez,bx,by,bz,
cU1     &                   mFx,mFy,mFz,DHDx,DHDy,DHDz,qme,DT,c)

 if (ionj.ne.0) then
cJET "qmi" for a pair jet injected into electron-ion plasma
c for ion jet
c       qmi2 = qi/me
       qmi2 = qi/mi
       call mover(ionj,xij,yij,zij,uij,vij,wij,mj,ex,ey,ez,bx,by,bz,
    &                        mFx,mFy,mFz,DHDx,DHDy,DHDz,qmi2,DT,c)
cU1        call mover2(ionj,xij,yij,zij,uij,vij,wij,mj,ex,ey,ez,bx,by,bz,
cU1     &                          mFx,mFy,mFz,DHDx,DHDy,DHDz,qmi,DT,c)
      end if
```

```
        if (lecj.ne.0) then
        call mover(lecj,xej,yej,zej,uej,vej,wej,mj,ex,ey,ez,bx,by,bz,
     &                          mFx,mFy,mFz,DHDx,DHDy,DHDz,qme,DT,c)
cU1        call mover2(lecj,xej,yej,zej,uej,vej,wej,mj,ex,ey,ez,bx,by,bz,
cU1     &                          mFx,mFy,mFz,DHDx,DHDy,DHDz,qme,DT,c)
        end if

c  ***************************************************************
       subroutine mover(ipar,x,y,z,u,v,w,mh,ex,ey,ez,bx,by,bz,
     &                    mFx,mFy,mFz,DHDx,DHDy,DHDz,qm,DT,c)

       dimension ex(mFx,mFy,mFz),ey(mFx,mFy,mFz),ez(mFx,mFy,mFz)
       dimension bx(mFx,mFy,mFz),by(mFx,mFy,mFz),bz(mFx,mFy,mFz)
       dimension x(mh),y(mh),z(mh)
       dimension u(mh),v(mh),w(mh)

       DO n = 1,ipar
c  cell index and displacement in cell
        i  = x(n)- DHDx
        dx = x(n)-i -DHDx
        j  = y(n)- DHDy
        dy = y(n)-j -DHDy
        k  = z(n)- DHDz
        dz = z(n)-k -DHDz
```

```
C Field interpolations are tri-linear (linear in x times linear in y
C times linear in z). This amounts to the 3-D generalisation of "area
C weighting". A modification of the simple linear interpolation formula
C         f(i+dx) = f(i) + dx * (f(i+1)-f(i))
C is needed since fields are recorded at half-integer locations in certain
C dimensions: see comments and illustration with the Maxwell part of this
C code. One then has first to interpolate from "midpoints" to "gridpoints"
C by averaging neighbors. Then one proceeds with normal interpolation.
C Combining these two steps leads to:
C    f at location i+dx  = half of f(i)+f(i-1) + dx*(f(i+1)-f(i-1))
C where now f(i) means f at location i+1/2. The halving is absorbed
C in the final scaling (e.g, in ex0 etc.).

c  E-component interpolations:
      f=ex(i,j,k)+ex(i-1,j,k)+dx*(ex(i+1,j,k)-ex(i-1,j,k))
      f=f+dy*(ex(i,j+1,k)+ex(i-1,j+1,k)+dx*(ex(i+1,j+1,k)-
     &          ex(i-1,j+1,k))-f)
      g=ex(i,j,k+1)+ex(i-1,j,k+1)+dx*(ex(i+1,j,k+1)-ex(i-1,j,k+1))
      g=g+dy*(ex(i,j+1,k+1)+ex(i-1,j+1,k+1)+dx*(ex(i+1,j+1,k+1)-
     &          ex(i-1,j+1,k+1))-g)
      ex0=(f+dz*(g-f))*(.25*qm)
      f=ey(i,j,k)+ey(i,j-1,k)+dy*(ey(i,j+1,k)-ey(i,j-1,k))
      f=f+dz*(ey(i,j,k+1)+ey(i,j-1,k+1)+dy*(ey(i,j+1,k+1)-
     &          ey(i,j-1,k+1))-f)
      g=ey(i+1,j,k)+ey(i+1,j-1,k)+dy*(ey(i+1,j+1,k)-ey(i+1,j-1,k))
      g=g+dz*(ey(i+1,j,k+1)+ey(i+1,j-1,k+1)+dy*(ey(i+1,j+1,k+1)-
     &          ey(i+1,j-1,k+1))-g)
      ey0=(f+dx*(g-f))*(.25*qm)

      f=ez(i,j,k)+ez(i,j,k-1)+dz*(ez(i,j,k+1)-ez(i,j,k-1))
      f=f+dx*(ez(i+1,j,k)+ez(i+1,j,k-1)+dz*(ez(i+1,j,k+1)-
     &          ez(i+1,j,k-1))-f)
      g=ez(i,j+1,k)+ez(i,j+1,k-1)+dz*(ez(i,j+1,k+1)-ez(i,j+1,k-1))
      g=g+dx*(ez(i+1,j+1,k)+ez(i+1,j+1,k-1)+dz*(ez(i+1,j+1,k+1)-
     &          ez(i+1,j+1,k-1))-g)
      ez0=(f+dy*(g-f))*(.25*qm)

c  B-component interpolations:
      f=bx(i,j-1,k)+bx(i,j-1,k-1)+dz*(bx(i,j-1,k+1)-bx(i,j-1,k-1))
      f=bx(i,j,k)+bx(i,j,k-1)+dz*(bx(i,j,k+1)-bx(i,j,k-1))+f+
     &    dy*(bx(i,j+1,k)+bx(i,j+1,k-1)+dz*(bx(i,j+1,k+1)-
     &    bx(i,j+1,k-1))-f)
      g=bx(i+1,j-1,k)+bx(i+1,j-1,k-1)+dz*(bx(i+1,j-1,k+1)-
     &    bx(i+1,j-1,k-1))
```

```
  g=bx(i+1,j,k)+bx(i+1,j,k-1)+dz*(bx(i+1,j,k+1)-bx(i+1,j,k-1))+g+
&       dy*(bx(i+1,j+1,k)+bx(i+1,j+1,k-1)+dz*(bx(i+1,j+1,k+1)-
&   bx(i+1,j+1,k-1))-g)
  bx0=(f+dx*(g-f))*(.125*qm/c)

  f=by(i,j,k-1)+by(i-1,j,k-1)+dx*(by(i+1,j,k-1)-by(i-1,j,k-1))
  f=by(i,j,k)+by(i-1,j,k)+dx*(by(i+1,j,k)-by(i-1,j,k))+f+
&   dz*(by(i,j,k+1)+by(i-1,j,k+1)+dx*(by(i+1,j,k+1)-
&   by(i-1,j,k+1))-f)
  g=by(i,j+1,k-1)+by(i-1,j+1,k-1)+dx*(by(i+1,j+1,k-1)-
&   by(i-1,j+1,k-1))
  g=by(i,j+1,k)+by(i-1,j+1,k)+dx*(by(i+1,j+1,k)-by(i-1,j+1,k))+g+
&   dz*(by(i,j+1,k+1)+by(i-1,j+1,k+1)+dx*(by(i+1,j+1,k+1)-
&   by(i-1,j+1,k+1))-g)
  by0=(f+dy*(g-f))*(.125*qm/c)

  f=bz(i-1,j,k)+bz(i-1,j-1,k)+dy*(bz(i-1,j+1,k)-bz(i-1,j-1,k))
  f=bz(i,j,k)+bz(i,j-1,k)+dy*(bz(i,j+1,k)-bz(i,j-1,k))+f+
&   dx*(bz(i+1,j,k)+bz(i+1,j-1,k)+dy*(bz(i+1,j+1,k)-
&   bz(i+1,j-1,k))-f)
```

```
      g=bz(i-1,j,k+1)+bz(i-1,j-1,k+1)+dy*(bz(i-1,j+1,k+1)-
     &   bz(i-1,j-1,k+1))
      g=bz(i,j,k+1)+bz(i,j-1,k+1)+dy*(bz(i,j+1,k+1)-bz(i,j-1,k+1))+g+
     &   dx*(bz(i+1,j,k+1)+bz(i+1,j-1,k+1)+dy*(bz(i+1,j+1,k+1)-
     &   bz(i+1,j-1,k+1))-g)
      bz0=(f+dz*(g-f))*(.125*qm/c)

c  multiplication by time-step
             ex0=DT*ex0
             ey0=DT*ey0
             ez0=DT*ez0
             bx0=DT*bx0
             by0=DT*by0
             bz0=DT*bz0

c  first-half electric acceleration, with relativity's gamma
             g=c/sqrt(c**2-u(n)**2-v(n)**2-w(n)**2)
      u0=g*u(n)+ex0
      v0=g*v(n)+ey0
      w0=g*w(n)+ez0
```

```
c  first-half magnetic rotation, with relativity's gamma
          g=c/sqrt(c**2+u0**2+v0**2+w0**2)
     bx0=g*bx0
     by0=g*by0
     bz0=g*bz0
     f=2.0/(1.0+bx0*bx0+by0*by0+bz0*bz0)
     u1=(u0+v0*bz0-w0*by0)*f
     v1=(v0+w0*bx0-u0*bz0)*f
     w1=(w0+u0*by0-v0*bx0)*f
c  second-half magnetic rotation and electric acceleration
     u0=u0+v1*bz0-w1*by0+ex0
     v0=v0+w1*bx0-u1*bz0+ey0
     w0=w0+u1*by0-v1*bx0+ez0
c  relativity's gamma
     g=c/sqrt(c**2+u0**2+v0**2+w0**2)
     u(n)=g*u0
     v(n)=g*v0
     w(n)=g*w0
c  position advance
     x(n)=x(n)+DT*u(n)
     y(n)=y(n)+DT*v(n)
     z(n)=z(n)+DT*w(n)
     END DO
     return
     end
```

Current deposit scheme (2-D)

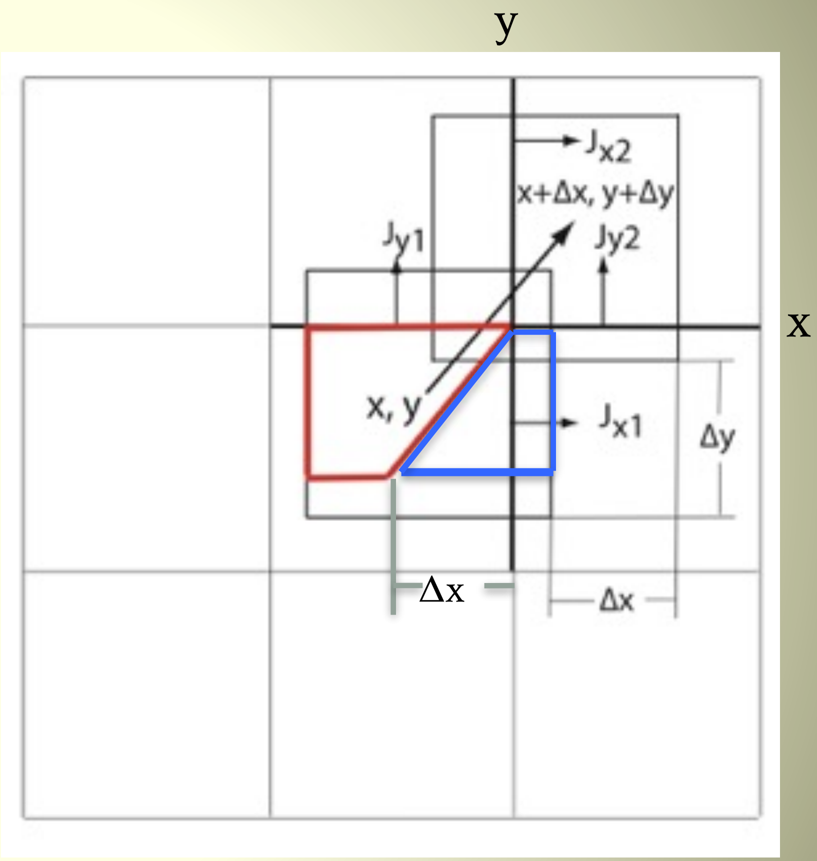
$$J_{x1} = q\Delta x\left(\frac{1}{2} - y - \frac{1}{2}\Delta y\right)$$

$$J_{x2} = q\Delta x\left(\frac{1}{2} + y + \frac{1}{2}\Delta y\right)$$

$$\rightarrow J_{y1} = q\Delta y\left(\frac{1}{2} - x - \frac{1}{2}\Delta x\right)$$

$$J_{y2} = q\Delta y\left(\frac{1}{2} + x + \frac{1}{2}\Delta x\right)$$

$$J_{x1} + J_{x2} = q\Delta x,\ J_{y1} + J_{y2} = q\Delta y$$



current can be calculate
the area of trapezoid indicated
by red lines and blue lines

Current deposit scheme (2-D)

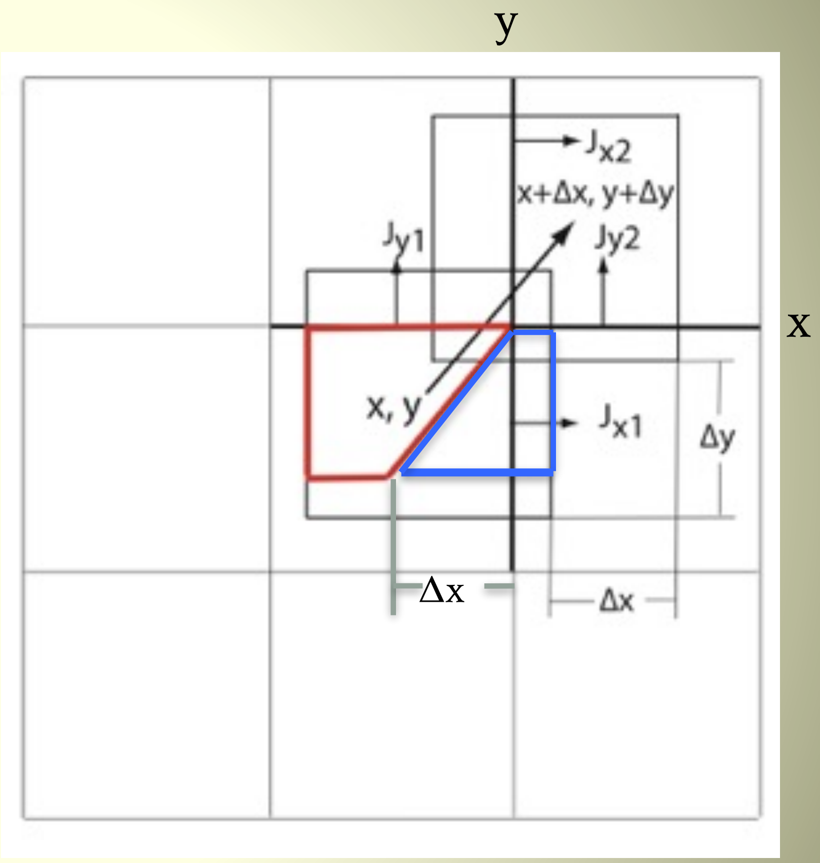
$$J_{x1} = q\Delta x\left(\frac{1}{2} - y - \frac{1}{2}\Delta y\right)$$

$$J_{x2} = q\Delta x\left(\frac{1}{2} + y + \frac{1}{2}\Delta y\right)$$

$$\rightarrow J_{y1} = q\Delta y\left(\frac{1}{2} - x - \frac{1}{2}\Delta x\right)$$

$$\rightarrow J_{y2} = q\Delta y\left(\frac{1}{2} + x + \frac{1}{2}\Delta x\right)$$

$$J_{x1} + J_{x2} = q\Delta x,\ J_{y1} + J_{y2} = q\Delta y$$



current can be calculate
the area of trapezoid indicated
by red lines and blue lines

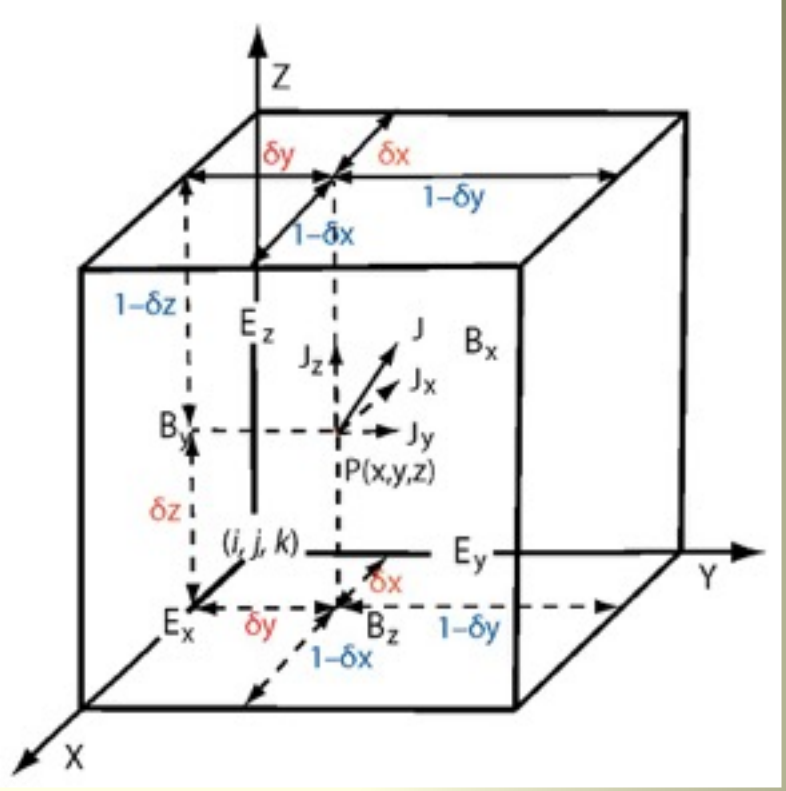
For example, for J_{y1} equation, the depth of charge moved is Δy, the width is $\frac{1}{2} - x$ at the start of the move and decreases linearly to $\frac{1}{2} - x - \Delta x$ at the end; the average width, which is relevant for linear motion, is $\frac{1}{2} - x - \frac{1}{2}\Delta x$.

Current deposit

Charge conservation $$\frac{\partial \rho}{\partial t} = -\nabla \cdot \mathbf{J}$$

$$\begin{aligned}
e_x(i,j,k) &= e_x(i+.5,j,k) \\
&= e_x(i,j,k) - J_x \cdot cy \cdot cz \\
e_x(i,j+1,k) &= e_x(i+.5,j+1,k) \\
&= e_x(i,j+1,k) - J_x \cdot \delta\, y \cdot cz \\
e_x(i,j,k+1) &= e_x(i+.5,j,k+1) \\
&= e_x(i,j,k+1) - J_x \cdot cy \cdot \delta\, z \\
e_x(i,j+1,k+1) &= e_x(i+.5,j+1,k+1) \\
&= e_x(i,j+1,k+1) - J_x \cdot \delta\, y \cdot \delta\, z
\end{aligned}$$

$$\begin{aligned}
e_y(i,j,k) &= e_y(i,j+.5,k) \\
&= e_y(i,j,k) - J_y \cdot cx \cdot cz \\
e_y(i,j+1,k) &= e_y(i+1,j+.5,k) \\
&= e_y(i+1,j,k) - J_y \cdot \delta\, x \cdot cz \\
e_y(i,j,k+1) &= e_y(i,j+.5,k+1) \\
&= e_y(i,j,k+1) - J_y \cdot cx \cdot \delta\, z \\
e_y(i,j+1,k+1) &= e_y(i+1,j+.5,k+1) \\
&= e_y(i+1,j,k+1) - J_y \cdot \delta\, x \cdot \delta\, z
\end{aligned}$$



Villasenor and Buneman 1992

$$cx = 1 - \delta\, x$$

$$cy = 1 - \delta\, y$$

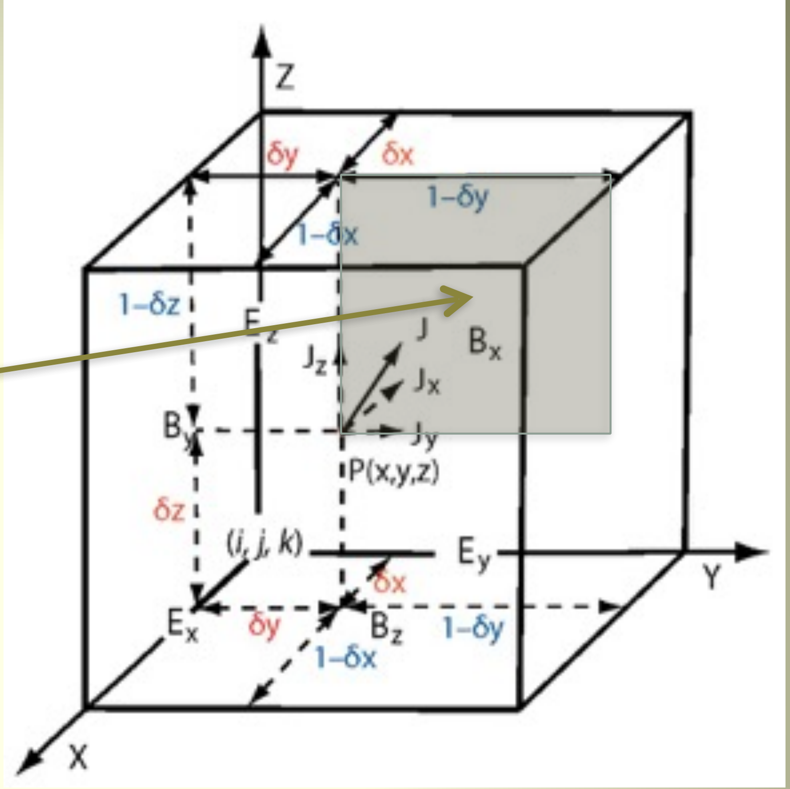
$$cz = 1 - \delta\, z$$

Current deposit

Charge conservation $$\frac{\partial\rho}{\partial t} = -\nabla\cdot\mathbf{J}$$

$$\begin{aligned}
e_x(i,j,k) &= e_x(i+.5,j,k)\\
&= e_x(i,j,k) - J_x\cdot cy\cdot cz\\
e_x(i,j+1,k) &= e_x(i+.5,j+1,k)\\
&= e_x(i,j+1,k) - J_x\cdot\delta y\cdot cz\\
e_x(i,j,k+1) &= e_x(i+.5,j,k+1)\\
&= e_x(i,j,k+1) - J_x\cdot cy\cdot\delta z\\
e_x(i,j+1,k+1) &= e_x(i+.5,j+1,k+1)\\
&= e_x(i,j+1,k+1) - J_x\cdot\delta y\cdot\delta z
\end{aligned}$$

$$\begin{aligned}
e_y(i,j,k) &= e_y(i,j+.5,k)\\
&= e_y(i,j,k) - J_y\cdot cx\cdot cz\\
e_y(i,j+1,k) &= e_y(i+1,j+.5,k)\\
&= e_y(i+1,j,k) - J_y\cdot\delta x\cdot cz\\
e_y(i,j,k+1) &= e_y(i,j+.5,k+1)\\
&= e_y(i,j,k+1) - J_y\cdot cx\cdot\delta z\\
e_y(i,j+1,k+1) &= e_y(i+1,j+.5,k+1)\\
&= e_y(i+1,j,k+1) - J_y\cdot\delta x\cdot\delta z
\end{aligned}$$



Villasenor and Buneman 1992

$$cx = 1-\delta x$$

$$cy = 1-\delta y$$

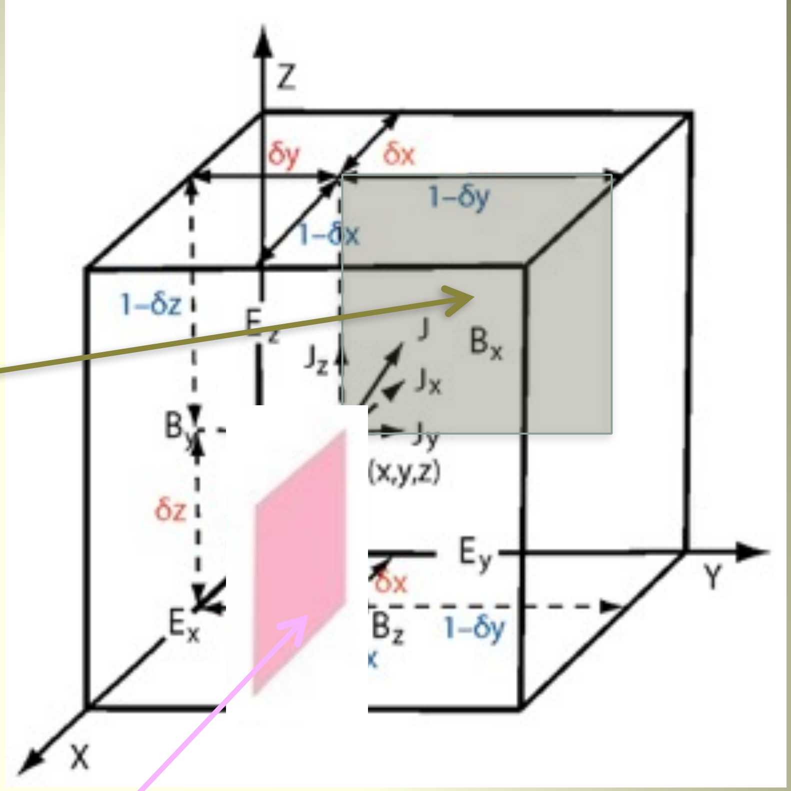
$$cz = 1-\delta z$$

Current deposit

Charge conservation $$\frac{\partial \rho}{\partial t} = -\nabla \cdot \mathbf{J}$$

$$\begin{aligned}
e_x(i,j,k) &= e_x(i+.5,j,k) \\
&= e_x(i,j,k) - J_x \cdot cy \cdot cz \\
e_x(i,j+1,k) &= e_x(i+.5,j+1,k) \\
&= e_x(i,j+1,k) - J_x \cdot \delta y \cdot cz \\
e_x(i,j,k+1) &= e_x(i+.5,j,k+1) \\
&= e_x(i,j,k+1) - J_x \cdot cy \cdot \delta z \\
e_x(i,j+1,k+1) &= e_x(i+.5,j+1,k+1) \\
&= e_x(i,j+1,k+1) - J_x \cdot \delta y \cdot \delta z
\end{aligned}$$

$$\begin{aligned}
e_y(i,j,k) &= e_y(i,j+.5,k) \\
&= e_y(i,j,k) - J_y \cdot cx \cdot cz \\
e_y(i,j+1,k) &= e_y(i+1,j+.5,k) \\
&= e_y(i+1,j,k) - J_y \cdot \delta x \cdot cz \\
e_y(i,j,k+1) &= e_y(i,j+.5,k+1) \\
&= e_y(i,j,k+1) - J_y \cdot cx \cdot \delta z \\
e_y(i,j+1,k+1) &= e_y(i+1,j+.5,k+1) \\
&= e_y(i+1,j,k+1) - J_y \cdot \delta x \cdot \delta z
\end{aligned}$$



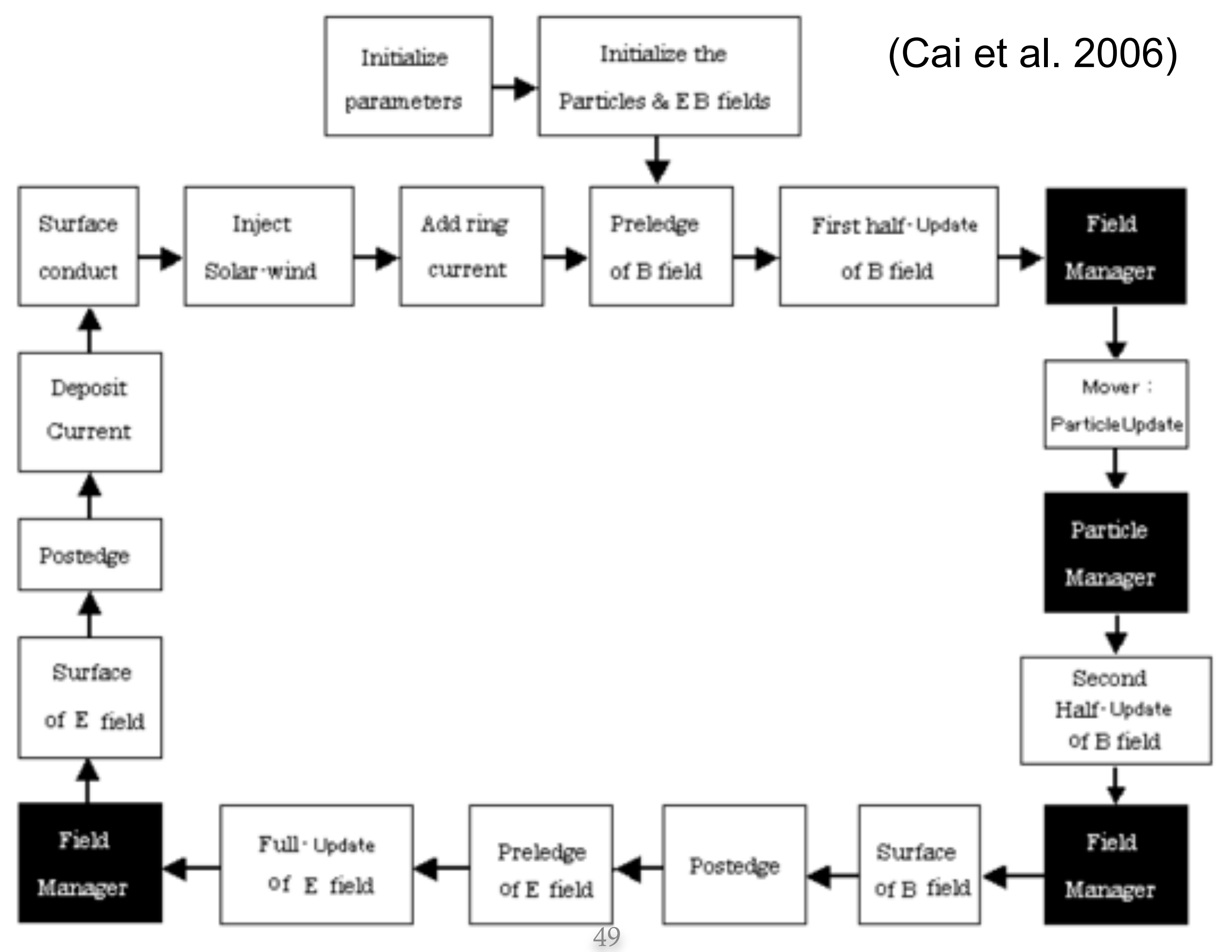
Villasenor and Buneman 1992

$$cx = 1 - \delta x$$

$$cy = 1 - \delta y$$

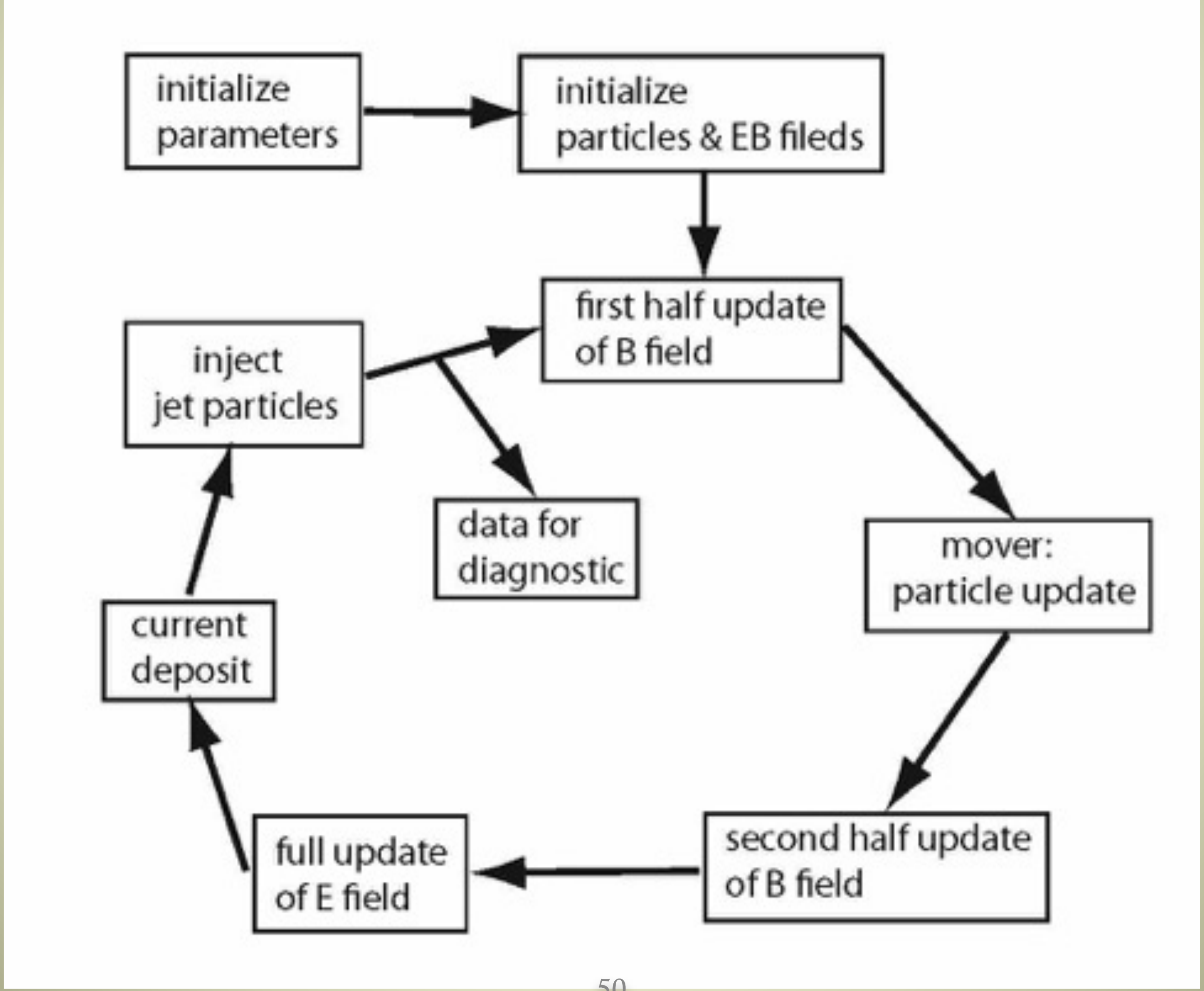
$$cz = 1 - \delta z$$

Schematic computational cycle



(Cai et al. 2006)

Time evolution of RPIC code



initialize parameters

initialize particles & EB fileds

first half update of B field

mover: particle update

second half update of B field

full update of E field

current deposit

inject jet particles

data for diagnostic

Code development

Combine these components

Set initial conditions for each problem you would like to investigate

Apply MPI for speed-up

Develop graphics using NCARGraphic, VisIt, AVSExpress, PARAVIEW, IDL, etc

Analyze simulation results and compare with theory and other simulation results

Prepare reports