Lecture 12

Electrodynamics

Hai-Qing Lin

Beijing Computational Science Research Center

This PowerPoint Notes Is Based on the Textbook '***An Introduction to Computer Simulation Methods : Applications to Physical Systems***', 2nd Edition, Harvey Gould and Jan Tobochnik, Addison-Wesley(1996); "A First Course in Computational Physics"; "Numerical Recipes"; "Elementary Numerical Analysis"; "Computational Methods in Physics and Engineering".

Question and Objective

- We discussed simulation of Mechanical and Statistical systems, how about electromagnetism?
- The **average theorem** and the solution of Laplace’s equation.
- Simulation of Maxwell’s equation.

Static Charges (Review)

The electric field **E**(**r**) at point **r** due to *N* point charges is given by (Coulomb's law)

$$\mathbf{E}(\mathbf{r}) = K\sum_{i}^{N}\frac{q_i}{\left|\mathbf{r}-\mathbf{r_i}\right|^3}(\mathbf{r}-\mathbf{r_i}), \qquad (1)$$

where $\mathbf{r}_i$ is the *fixed* location of the *i*th charge and *K* is a constant that depends on the choice of units.

- SI units: $K = 1/4\pi\varepsilon_0 \approx 9.0 \times 10^9$ Nm^2/C^2
- Gaussian units: $K = 1$

Force Acting on Two Charges

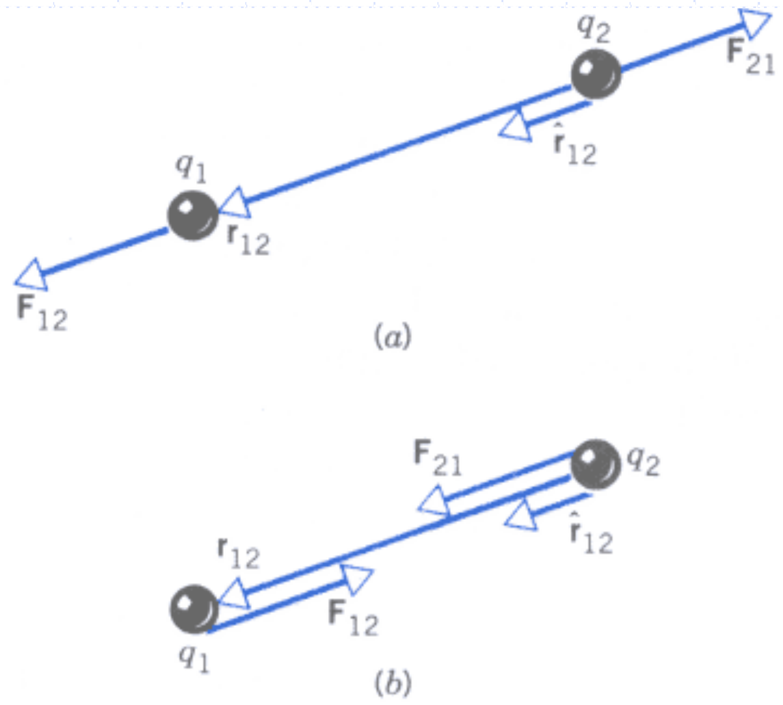


Figure 5 (*a*) Two point charges q_1 and q_2 of the same sign exert equal and opposite repulsive forces on one another. The vector $\mathbf{r}_{12}$ locates q_1 relative to q_2, and the unit vector $\hat{\mathbf{r}}_{12}$ points in the direction of $\mathbf{r}_{12}$. Note that $\mathbf{F}_{12}$ is parallel to $\mathbf{r}_{12}$. (*b*) The two charges now have opposite signs, and the force is attractive. Note that $\mathbf{F}_{12}$ is antiparallel to $\mathbf{r}_{12}$.

The Lorentz Force

- $\mathbf{F} = q\,(\mathbf{E} + \mathbf{v} \times \mathbf{B} \,/\, c)$
- In Gaussian units (K=1).

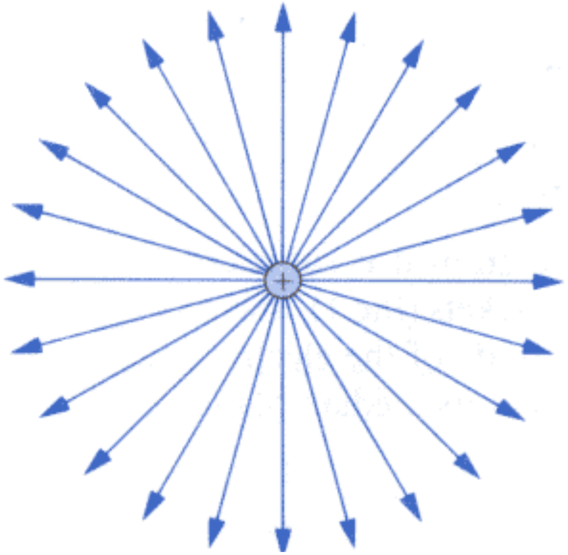


Figure 5 Lines of force surrounding a positive point charge. The direction of the force on a positive test charge, and thus the direction of the electric field at any point, is indicated by the direction of the lines. The relative spacing between the lines at any location indicates the relative strength of the field at that location. The lines are assumed to terminate on distant negative charges that are not shown.

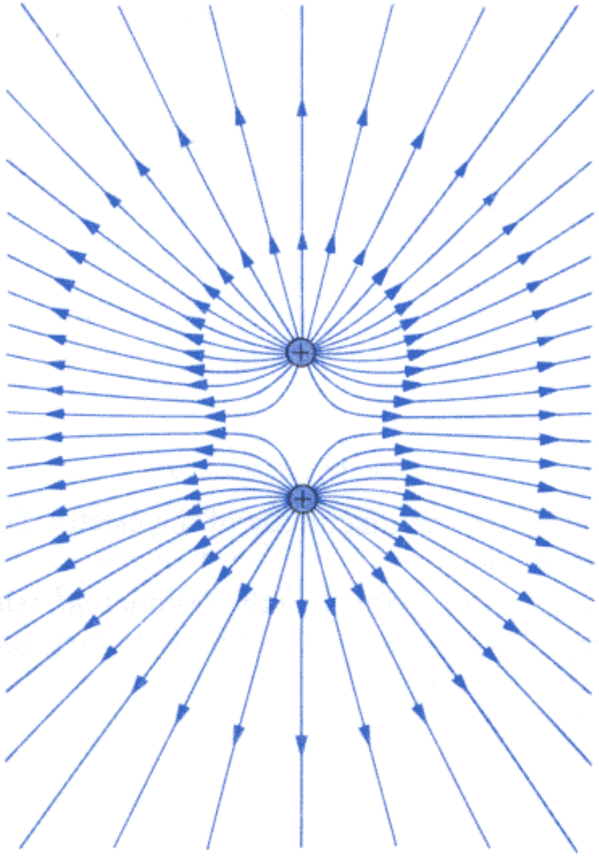
Electric Field Lines

Electric field lines are used to visualising the electric field.

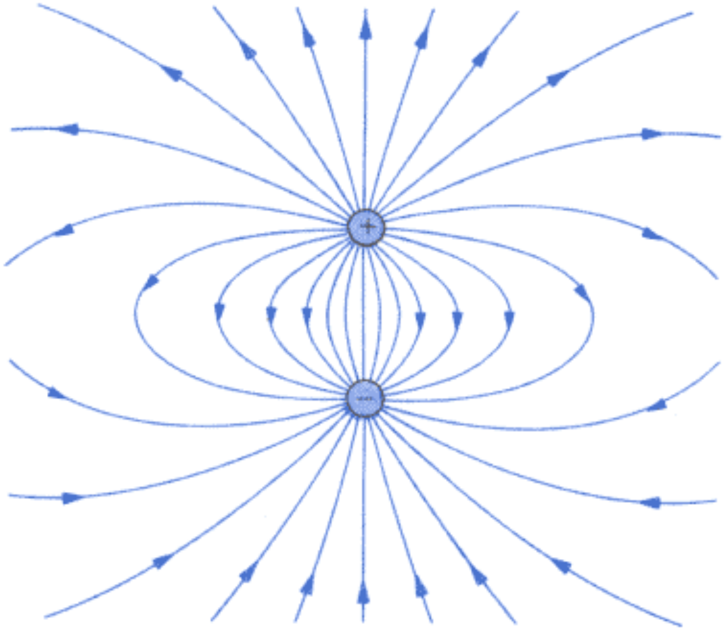
- An electric field line is a directed line whose tangent at every position is parallel to the electric field at that position.
- The lines are smooth and continuous except at singularities such as point charges.
- The density of lines at any point in space is ∝ to the magnitude of the field at that point.

Electric Field Lines

Lines of force surrounding two equal positive charges.



Lines of force surrounding a positive and negative charges of equal magnitude (an electric dipole).



Program *fieldlines* draws electric field lines in 2D starting at +ve charges if the net charge $q_{\text{net}} \geq 0$ or at –ve charges if $q_{\text{net}} < 0$.

The program implements the following algorithm:

1. Begin at a point (x,y) near a charge and compute the components E_x and E_y of the electric field vector **E** using Coulomb's law.
2. Draw a small line segment of size $\Delta s = |\Delta \mathbf{s}|$ tangent to **E** at that point. If $q_{\text{net}} \geq (<) \, 0$, then $\Delta s > (<) \, 0$. The components of the line segment are given by

$$\Delta x = \Delta s \frac{E_x}{|\mathbf{E}|}, \quad \Delta y = \Delta s \frac{E_y}{|\mathbf{E}|}$$

This program uses a small value for **Δ*s*** if the field line is close to the charges or if the field magnitude is large.

Program *fieldlines* draws electric field lines in 2D starting at +ve charges if the net charge $q_{\text{net}} \geq 0$ or at –ve charges if $q_{\text{net}} < 0$.

3. Repeat the process beginning at the new point $(x+\Delta x, y+\Delta y)$. Continue until the field line approaches another charge.
4. Repeat steps 1 to 3 for equally spaced starting positions on a circle around the charge. The spacing is inversely proportional to the magnitude of the charge.
5. Repeat steps 1 to 4 for each charge of the same sign.

Read program listings on page 303 to 305.

Electric Potential

- Definition

$$V\left(\mathbf{r}_2\right)-V\left(\mathbf{r}_1\right)=-\int_{\mathbf{r}_1}^{\mathbf{r}_2}\mathbf{E}\cdot d\mathbf{r}$$

$$\mathbf{E}\left(\mathbf{r}\right)=-\nabla V\left(\mathbf{r}\right)$$

- For a point charge

$$V(\boldsymbol{r}) = q\,/\,r \qquad \text{(Gaussian unit)}$$

- A better approach than the field vector.

Equi-potential Surface

- It is the surface on which the electric potential has an ***equal value*** everywhere.
- The **E** lines are ***orthogonal*** to the equipotential surface at any point.
- This property can be used to draw electric field lines or use the same program to draw equipotential surfaces.

Equipotential Surfaces: Examples

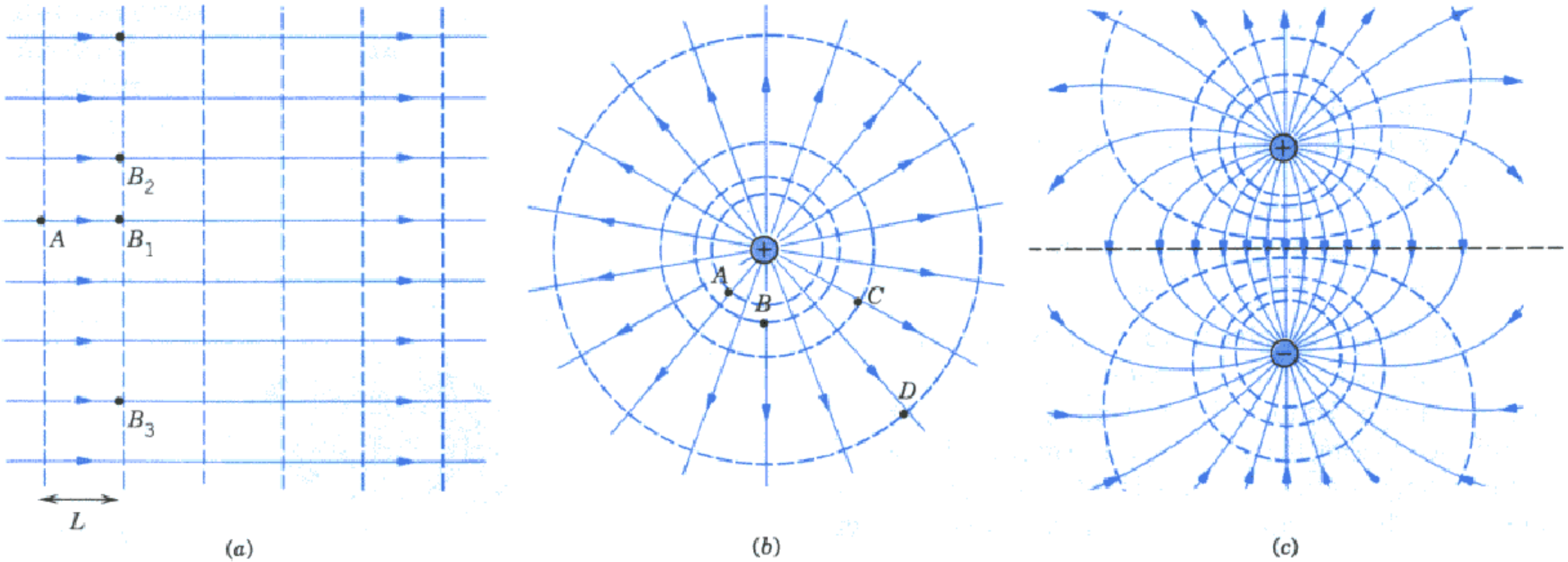


Figure 15 Lines of force (solid lines) and cross sections of equipotential surfaces (dashed lines) for (*a*) a uniform field, (*b*) a positive point charge, and (*c*) an electric dipole.

Numerical Solutions of Laplace's Equation

- The potential *V*(*x*,*y*,*z*) obeys the **Laplace's** equation: (Charge-free case)

$$\nabla^2 V(x,y,z) = \frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2} + \frac{\partial^2 V}{\partial z^2} = 0, \qquad (2)$$

subjects to the specified boundary conditions.

- This is one of the ***boundary value*** problems.
- The only general approach for arbitrary shape of boundary is to use numerical methods.

Discrete Form of Laplace's Equation in 2D

Do Taylor expansion

$$V(x \pm \Delta x, y) = V(x,y) \pm \Delta x \frac{\partial V(x,y)}{\partial x} + \frac{(\Delta x)^2}{2} \frac{\partial^2 V(x,y)}{\partial x^2} \pm \cdots$$

$$V(x, y \pm \Delta y) = V(x,y) \pm \Delta y \frac{\partial V(x,y)}{\partial y} + \frac{(\Delta y)^2}{2} \frac{\partial^2 V(x,y)}{\partial y^2} \pm \cdots$$

$$V(x,y) \approx \frac{1}{4}[V(x+\Delta x, y) + V(x-\Delta x, y) + V(x, y+\Delta y) + V(x, y-\Delta y)]. \qquad (3)$$

V(*x*,*y*) is the ***average*** of the potential at the four nearest neighbour points, a remarkable property!

Discrete Form of Laplace's Equation in 2D

- Program ***verify*** tests Eq. (3) by calculating the potential due to a point charge at points $r \neq 0$.
- Eq. (3) is consistent with Coulomb's law so we adopt it as the ***basis*** for computing the potential for systems which we cannot calculate potential analytically.
- Two popular methods: ***relaxation*** and ***random walk***.

Relaxation Method (Jacobi)

This method is based on the following algorithm:

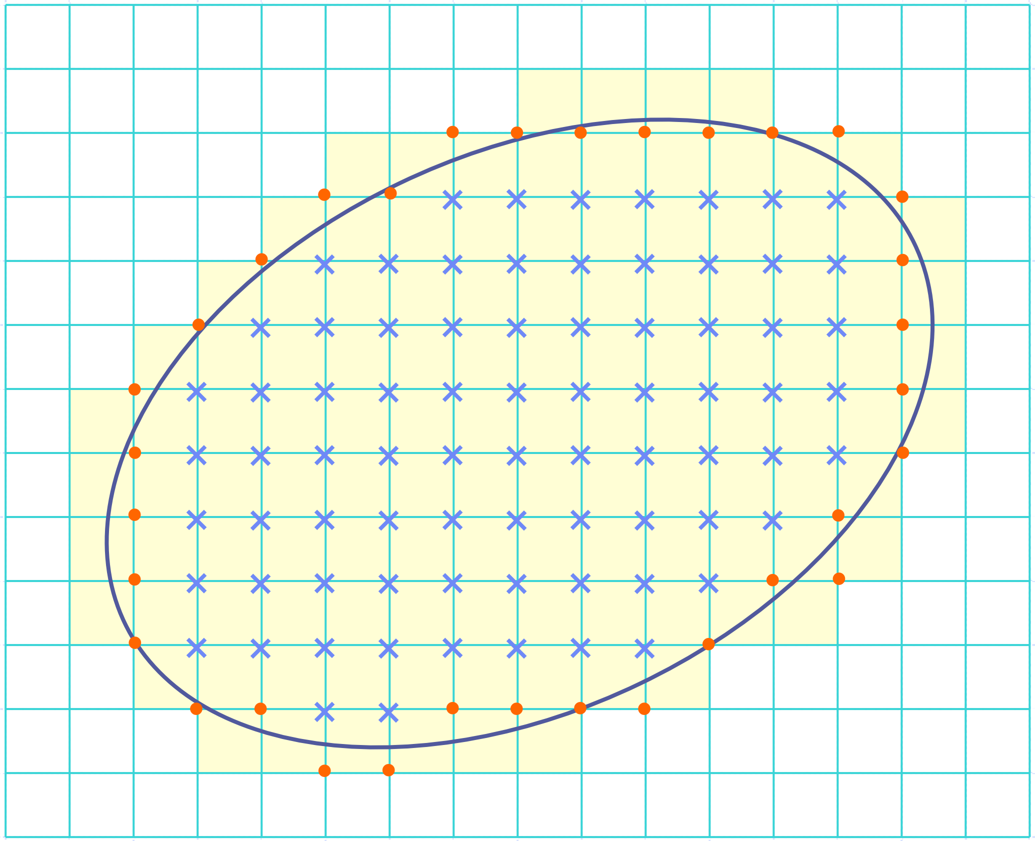
1. Divide the region of interest into rectangular (cubic) **grid** of points spanning the region. The region is enclosed by a **curve (surface)** with specified values of the potential along the curve (surface). The grid need not to be the same size every time, everywhere.

2. Assign to a grid boundary point the potential of the (real) boundary nearest the point.

Relaxation Method

3. Assign all interior points an arbitrary potential (preferably a reasonable guess).
4. Compute new values for the potential V for each interior point. Each new value is obtained by finding the average of the previous values of the potential at the four (six) nearest points.
5. Repeat step **4** using the values of V obtained in the previous iteration. This iterative process is continued until the potential at each interior point is computed to the desired accuracy.

Relaxation Method



V_0

$V(x_i, y_i)$

V

Program Laplace (Sample by HQ)

```
PROGRAM LAPLACE2D
IMPLICIT NONE
integer :: nx,ny
REAL, ALLOCATABLE :: v(:,:)
real :: epsilon,omega
   write(0,*) 'Enter Nx,Ny'
   read(0,*) nx,ny
   allocate( v(0:nx+1,0:ny+1) )
   call ASSIGN(nx,ny,v,epsilon,omega)
   write(0,*) 'Solution of 2D Laplace Eq. with Nx/Ny = ',nx,ny
   call ITERATE(nx,ny,v,epsilon,omega)
   stop
End PROGRAM LAPLACE2D
```

```
SUBROUTINE ASSIGN(nx,ny,v,epsilon,omega)
IMPLICIT NONE
integer :: nx,ny,i,j
REAL :: v(0:nx+1,0:ny+1)
Real :: vx_0,vx_a,vy_0,vy_b,vguess,epsilon,omega
Real :: pi,tmpx

   write(0,*) 'ASSIGN: Nx/Ny = ',nx,ny
   write(0,*) 'Enter V at boundaries: y=0,y=b,x=0,x=a'
   read(0,*) vy_0,vy_b,vx_0,vx_a

! Optimal Relaxation parameter for rectangular
   pi=2.0*acos(0.0)
   tmpx=( cos(pi/float(nx+2)) + cos(pi/float(ny+2)) )/2.0
   omega=2.0/( 1.0 + sqrt(1.0-tmpx*tmpx) )
   write(0,*) 'Optimal Omega = ',omega

   write(0,*) 'Enter Error Tolerance and Relaxation'
   read(0,*) epsilon,omega
```

```
  do i=0,nx+1          ! at y boundary
    v(i,0) = vy_0
    v(i,ny+1) = vy_b
  end do

  do j=0,ny+1          ! at x boundary
    v(0,j) = vx_0
    v(nx+1,j) = vx_a
  end do

  vguess = ( vy_0 + vy_b + vx_0 + vx_a ) / 4.0
  do i=1,nx
  do j=1,ny
    v(i,j) = vguess
  end do
  end do

END SUBROUTINE ASSIGN
```

```
SUBROUTINE ITERATE(nx,ny,v,epsilon,omega)
      IMPLICIT NONE

      integer :: nx,ny,i,j,iteration
      real :: vx_0,vx_a,vy_0,vy_b,vguess,epsilon,omega
      real :: change,omega1,dv,vave

      REAL :: v(0:nx+1,0:ny+1)

      write(0,*) 'ITERATE: Nx/Ny = ',nx,ny

      omega1=1.0-omega
      iteration=0
999   continue
      iteration=iteration+1
      change=0.0
```

```
     do 100 i=1,nx
     do 100 j=1,ny
       vave = ( v(i-1,j)+v(i+1,j)+v(i,j-1)+v(i,j+1) ) / 4.0

       change=change+abs( vave - v(i,j) )
       v(i,j) = omega1*v(i,j) + omega*vave
100  Continue
     if( mod(iteration,100) == 0 ) write(0,*) 'Ite =',iteration

     change=change/float(nx*ny)     ! May do differently
     if( change > epsilon ) go to 999

     write(0,*) 'Number of iterations = ',iteration
     write(0,*) 'Error = ',change

     END SUBROUTINE ITERATE
```

Gauss-Seidel Relaxation Method

The ***Jacobi*** method updates the potential at **each point** simultaneously and it converges very slowly. The reason is that it takes a long time for a change in the potential at one point to affect changes further away. There are methods to take care of this problem and Gauss-Seidel relaxation method is one of them:

$$V(x,y) = w\,\overline{V(x,y)} + (1-w)\,V(x,y)$$

Over-relaxation parameter

$$V(x,y) = w\,\overline{V(x,y)} + (1-w)V(x,y)$$

The ***over-relaxation*** parameter ***w*** is in the range 1 < ***w*** < 2. The effect of ***w*** is to cause the potential to change by a greater amount than in the simple relaxation procedure. For certain geometry, one may obtain optimal ***w*** analytically, e.g., rectangle.

```
      do 100 i=1,nx
      do 100 j=1,ny
         vave = ( v(i-1,j)+v(i+1,j)+v(i,j-1)+v(i,j+1) ) / 4.0
         change=change+abs( vave - v(i,j) )
         v(i,j) = omega1*v(i,j) + omega*vave
100   continue
```

↑

This is correct!

```
      do 100 i=1,nx
      do 100 j=1,ny
           vave(i,j) = ( v(i-1,j)+v(i+1,j)+v(i,j-1)+v(i,j+1) )/4.0
           change=change+abs( vave(i,j) - v(i,j) )
100    continue

       do 200 i=1,nx
       do 200 j=1,nx
            v(i,j) = omega1*v(i,j) + omega*vave(i,j)
200    continue
```

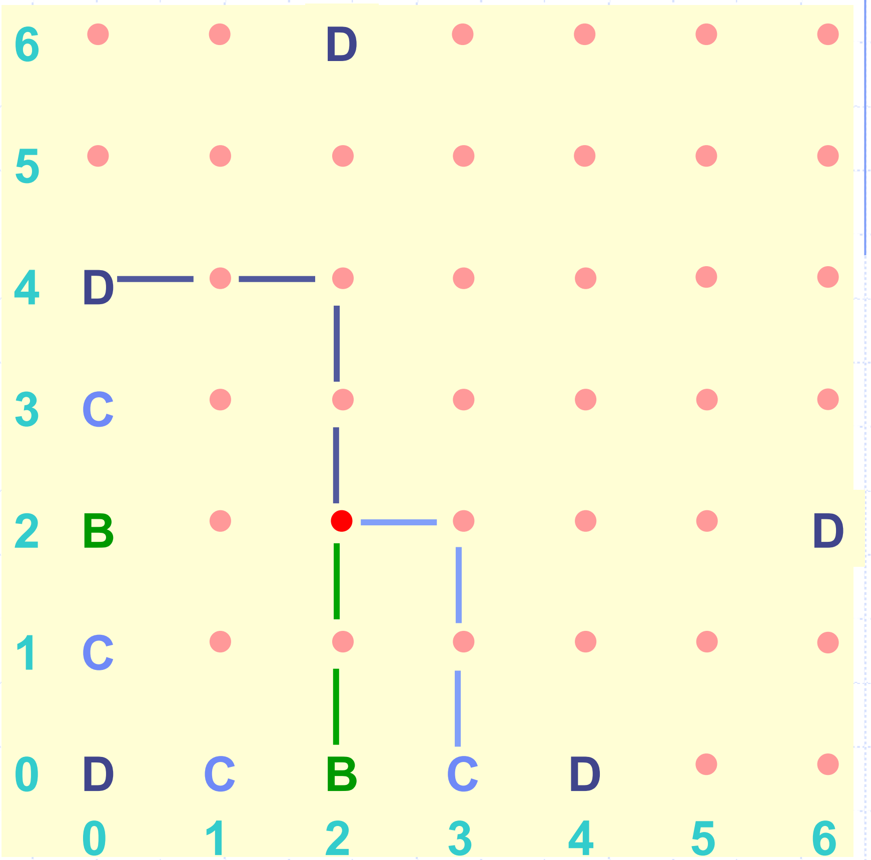
Example: Consider a 5×5 lattice

$$V_{2,2} \xrightarrow[\text{1st}]{} \left(\tfrac{1}{4}\right)[V_{1,2}+V_{3,2}+V_{2,1}+V_{2,3}]$$

$$\xrightarrow[\text{2nd}]{} \left(\tfrac{1}{4}\right)^2 [V_{0,2}+V_{1,1}+V_{1,3}+V_{2,0}$$
$$+V_{3,1}+V_{4,2}+V_{3,3}+V_{2,4}]$$

$$\xrightarrow[\text{3rd}]{} \left(\tfrac{1}{4}\right)^3 [V_{1,0}+V_{0,1}+V_{0,3}+V_{1,4}+V_{2,5}$$
$$+V_{3,4}+V_{4,3}+V_{5,2}+V_{4,1}+_{3,0}]$$

$$\xrightarrow[\text{4th}]{} \left(\tfrac{1}{4}\right)^4 [V_{0,4}+V_{1,5}+V_{2,6}+V_{3,5}+V_{4,4}$$
$$+V_{5,3}+V_{6,2}+V_{5,1}+V_{4,0}+V_{0,0}]$$



Potential at boundary: $\left(\tfrac{1}{4}\right)^2 : V_{0,2}, V_{2,0}$

$\left(\tfrac{1}{4}\right)^3 : V_{1,0}, V_{0,1}, V_{0,3}, V_{3,0}$

$\left(\tfrac{1}{4}\right)^4 : V_{0,0}, V_{0,4}, V_{2,6}, V_{4,0}, V_{6,2}$

Poisson’s Equation

- If there is charge density $\rho(x,y,z)$ in the region, then

$$\nabla^2 V = \frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2} + \frac{\partial^2 V}{\partial z^2} = -4\pi\rho(\mathbf{r}), \qquad (4)$$

- and its difference form in 2D is

$$\begin{aligned} V(x,y) \approx \frac{1}{4}\big[& V(x+\Delta x, y) + V(x-\Delta x, y) \\ + \quad & V(x, y+\Delta y) + V(x, y-\Delta y)\big] \\ & + \frac{1}{4} 4\pi \Delta x \Delta y \rho(x,y,z) \qquad (5) \end{aligned}$$

Random Walk Solution of Laplace's Equation

- The average theorem (Eq. (3)) could be rewritten as

$$V(x,y)=\frac{1}{4}\sum_{i=1}^{4}V(i)\ , \qquad (6)$$

where $V(i)$ is the value of the potential at the ith neighbour. [Generalisation of this result to 3D is straightforward.]

- This relation can be given a probabilistic interpretation in terms of random walk. The random walk algorithm for computing the solution to Laplace's equation is the following:

Random Walk Solution of Laplace's Equation

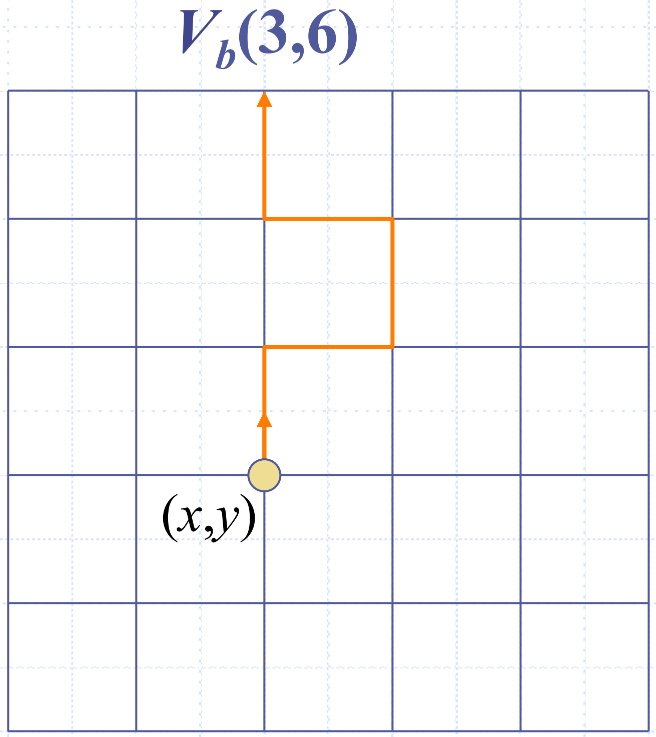
1. Begin at a point (x, y) where the value of the potential is desired, and take a step in a random direction.
2. Continue taking steps until the walker reaches the surface. Record $V_b(i)$, the potential at the boundary point $\boldsymbol{i}$.
3. Repeat step **1** and **2**, ***n*** times and sum the potential found at the surface each time.
4. The value of the potential at the point (x,y) is estimated by

$$V(x, y) = \frac{1}{n} \sum_{i=1}^{n} V_b(i) \qquad (7)$$

where ***n*** is the total number of random walkers.

Random Walk on 6 × 6 Lattice

A random walk on a 6 × 6 grid starting at the point $(x,y) = (3,3)$ and ending at the boundary point $V_b(3,6)$ where the potential is recorded.



Comparison of two algorithms

- The disadvantage of the random walk method is that it required **many walker** to obtain a good estimate of the potential at each point.
- However, it is more appropriate than the relaxation method if the potential is needed at only a small number of points, since the error is proportional to $\boldsymbol{n^{-1/2}}$**.**
- Both methods avoid complicated and difficult (or impossible) mathematical manipulations.

Green's Function Approach

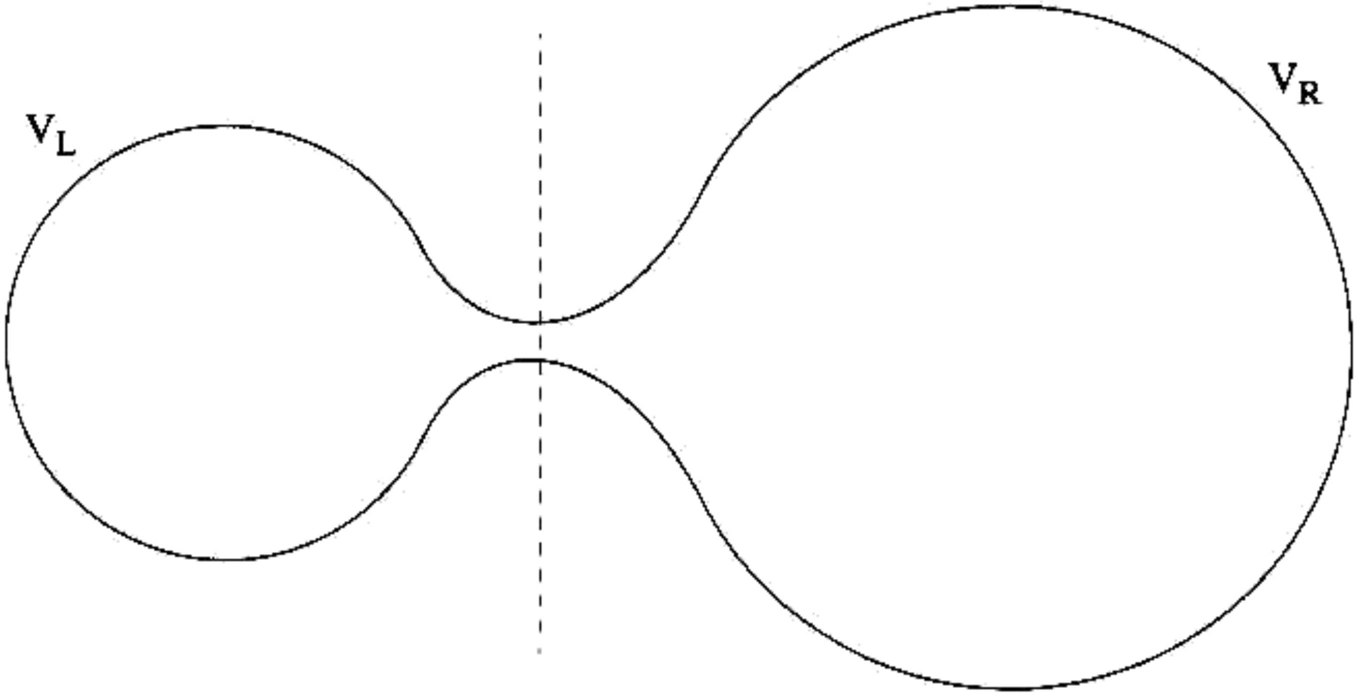
If the number of times that a walker from the point (x,y) lands at the boundary (x_b,y_b) is $G(x,y;\, x_b,y_b)$, then the random walker algorithm is equivalent to the relation.

$$V(x,y) = \frac{1}{n}\sum_b G(x,y;x_b,y_b)\,V(x_b,y_b),$$

where the sum is over all points on the boundary. We can use the ***same function G*** for ***different distribution of the potential on a given boundary***. ***G*** is called the Green's function which depends on the **geometry** of the boundary only.

RW view of Laplace's Equation

The RW algorithm can help us gain additional insight into the nature of Laplace’s equation, e.g, figure below:



Two regions of space connected by a narrow neck

RW Algorithm for Poisson's Equation

$$V(x,y) = \frac{1}{n}\sum_{\alpha} V(\alpha) + \frac{\pi \Delta x \Delta y}{n}\sum_{i,\alpha} \rho(x_{i,\alpha}, y_{i,\alpha}),$$

where α labels the walker, and *i* labels the points visited by the walker. That is, each time a walker is at point *i,* we add the charge density at that point.

Maxwell's Equations

$$\frac{\partial \mathbf{E}}{\partial t} = c\nabla\times\mathbf{B} - 4\pi\mathbf{j} \qquad (8)$$

$$\frac{\partial \mathbf{B}}{\partial t} = -\frac{1}{c}\nabla\times\mathbf{E} \qquad (9)$$

$$\nabla\cdot\mathbf{E} = 4\pi\rho \qquad (10)$$

$$\nabla\cdot\mathbf{B} = 0 \qquad (11)$$

Charge conservation:

$$\frac{\partial \rho}{\partial t} = -\nabla\cdot\mathbf{j} \qquad (12)$$

Maxwell's Equations

- A complete description of electrodynamics requires Eq. (8), (9), (10) and (12) and the initial values of all currents and fields. Eqs. (8)-(12) are first order differential equations and we know how to solve them numerically, if we interpret the ***curl*** and ***divergence*** of a vector correctly.
- A coordinate free definition of the curl of an arbitrary vector **W** is

$$(\nabla \times \mathbf{W}) \cdot \hat{\mathbf{S}} = \lim_{S \to 0} \frac{1}{S} \oint_c \mathbf{W} \cdot d\mathbf{l} \qquad (13)$$

where **S** is the area of ***any surface*** bordered by the closed curve *C*, and $\hat{\mathbf{S}}$ is a unit vector normal to *S*. Eq. (13) is our basic formula for computing the curl numerically.

Maxwell's Equations

- We divide space into cubes of $(\Delta l)^3$. The rectangular components of **W** can be defined either on the ***edges*** or on the ***faces*** of the cubes. Both definition can be used but we must retain consistency.
- Assuming **B** is defined on the ***edge*** so that the curl of **B** is defined on the ***faces*** (Fig. 10.5), then on the front face of the cube (i,j,k), we have

$$(\nabla \times \mathbf{B}) \cdot \hat{\mathbf{S}} = \frac{1}{(\Delta l)^2} \sum_{i=1}^{4} B_i \Delta l_i, \quad S = (\Delta l)^2$$

The curl of **B** defined on the edges of a cube

Fig. a

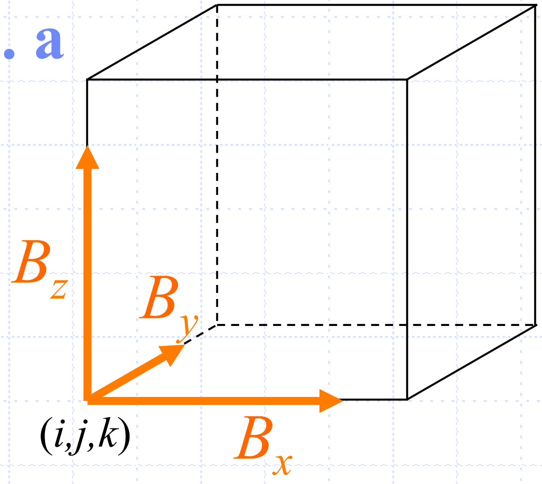


Fig. b

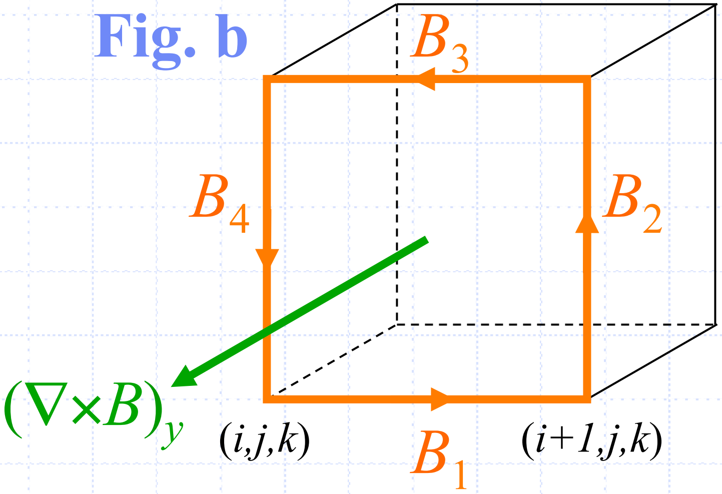
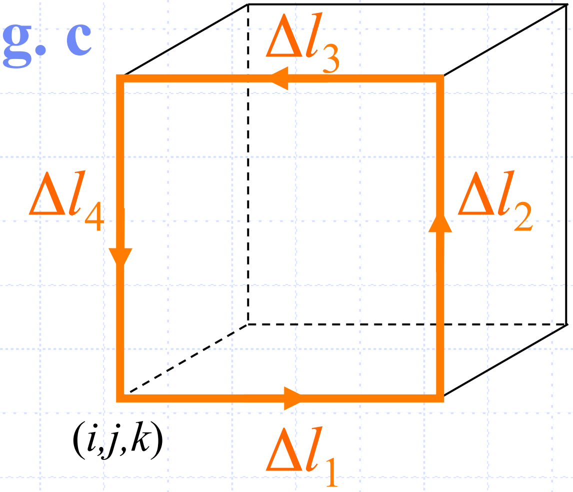
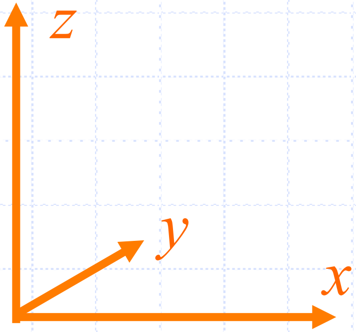


Fig. c





The curl of **E** defined on the faces of a cube

Fig. a

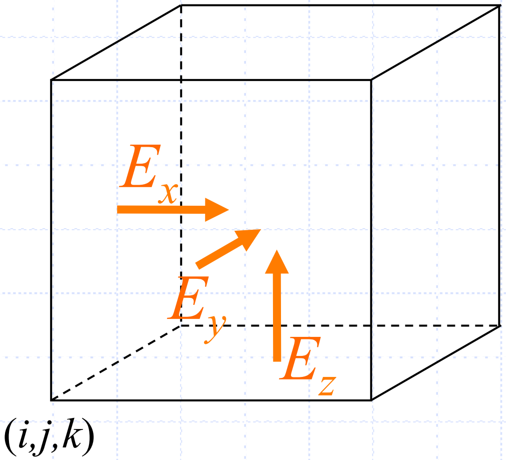


Fig. b

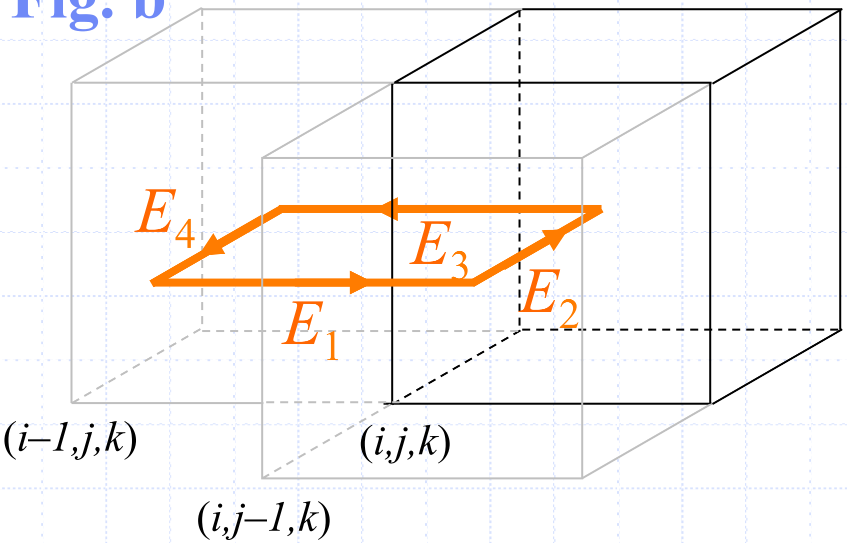
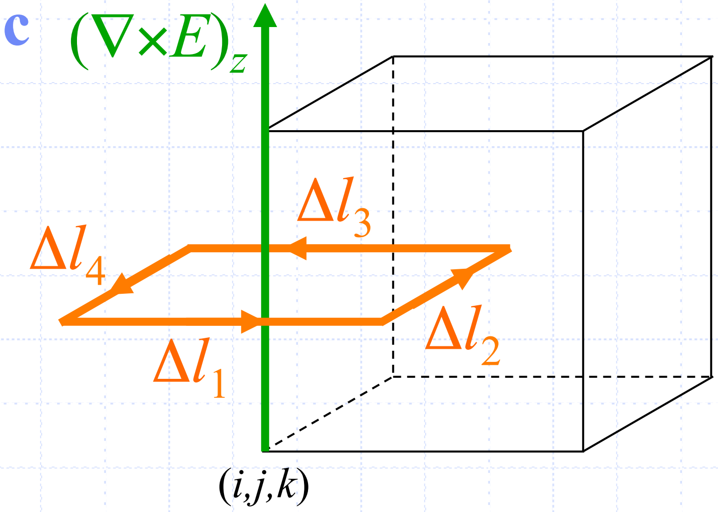
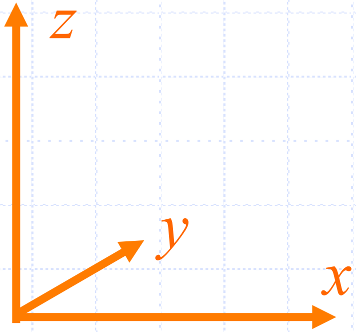


Fig. c





Maxwell's Equations

A co-ordinates free definition of the divergence of the vector field **W** is

$$\nabla \cdot \mathbf{W} = \lim_{V \to 0} \frac{1}{V} \oint_S \mathbf{W} \cdot d\mathbf{S} \qquad (14)$$

where V is the volume enclosed by the closed surface **S.** The divergence measures the average flow of the vector through a closed surface. Its discrete version is similar to the case of the curl.

Numerical Solution of Maxwell's Equations

We divide the region into many small cubes and we are interested in quantities of ρ, **j**, **E** and **B** on the grid. They cannot be defined arbitrary. Our guides are Maxwell's equations. If we define the charge density ρ at the centre of a cube, then we must define

- **j** at the faces of the cube because $\dfrac{\partial \rho}{\partial t} = -\nabla \cdot \mathbf{j}$
- **E** on the faces of the cube because $\dfrac{\partial \mathbf{E}}{\partial t} = c\nabla \times \mathbf{B} - 4\pi\mathbf{j}$
- **B** on the edges of the cube because $\dfrac{\partial \mathbf{B}}{\partial t} = -\dfrac{1}{c}\nabla \times \mathbf{E}$

Maxwell’s Equations

Associated with each cube is one edge vector and one face vector (3 components each). We label the cube by the coordinates corresponding to its ***lower left front corner (i, j, k)***.

For the **B** vector

B_x: $(i, j, k) \rightarrow (i+1, j, k)$

B_y: $(i, j, k) \rightarrow (i, j+1, k)$

B_z: $(i, j, k) \rightarrow (i, j, k+1)$

For the **E** vector

E_x: $(i, j+\frac{1}{2}, k+\frac{1}{2}) \rightarrow (i+\frac{1}{2}, j+\frac{1}{2}, k+\frac{1}{2})$

E_y: $(i+\frac{1}{2}, j, k+\frac{1}{2}) \rightarrow (i+\frac{1}{2}, j+\frac{1}{2}, k+\frac{1}{2})$

E_z: $(i+\frac{1}{2}, j+\frac{1}{2}, k) \rightarrow (i+\frac{1}{2}, j+\frac{1}{2}, k+\frac{1}{2})$

Calculation of the curl of **B** defined on the edges of a cube

Fig.a The edge vector **B** associated with cube (i,j,k).

Fig.b The components B_i along the edges of the front face of the cube.

$B_1 = B_x(i,j,k)$, $B_2 = B_z\,(i+1,j,k)$,

$B_3 = -B_x(i,j,k+1)$, $B_4 = -B_z(i,j,k)$.

Fig.c The vector components Δl_i on the edges of the front face.

(The y-component of $\nabla\times$**B** defined on the face points in the negative y direction.)

Fig. a

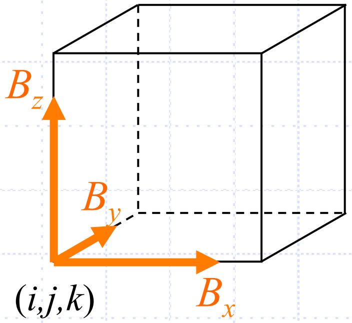


Fig. b

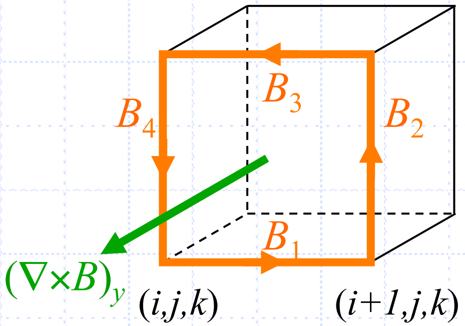
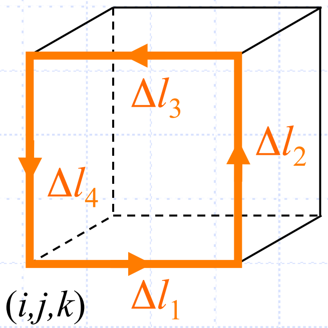
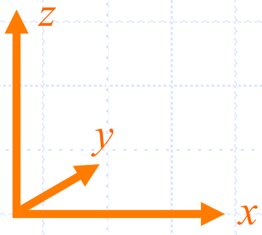


Fig. c





Calculation of the curl of **E** defined on the faces of a cube

Fig.a The face vector **E** associated with the cube (i,j,k). The components associated with the left, front, and bottom faces are $E_x(i,j,k)$, $E_y(i,j,k)$, $E_z(i,j,k)$ respectively.

Fig.b The components E_i on the faces that share the front left edge of the cube (i,j,k). $E_1 = E_x(i,j\text{–}1,k)$, $E_2 = E_y(i,j,k)$, $E_3 = \text{–}E_x(i,j,k)$, $E_4 = \text{–}E_y(i\text{–}1,j,k)$. The cube associated with E_1 and E_4 also are shown.

Fig.c The vector components Δl_i on the faces that share the left front edge of the cube. (The z-component of the curl of **E** defined on the left edge points in the positive z direction.)

Fig. a

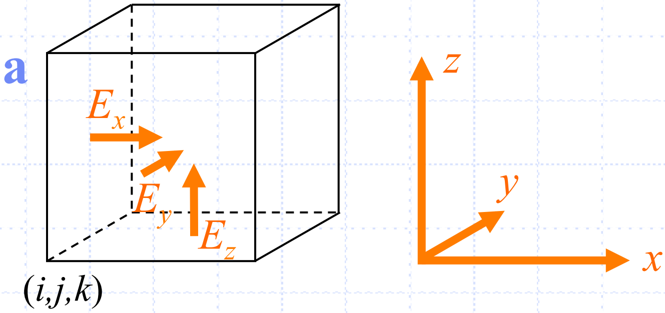


Fig. b

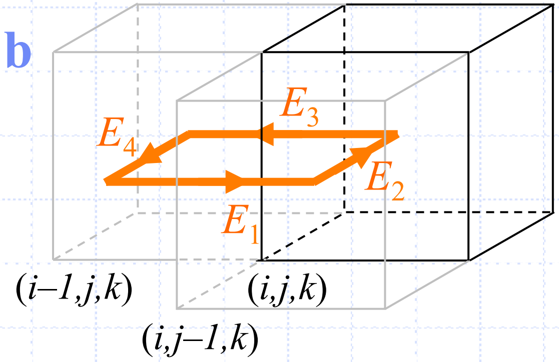
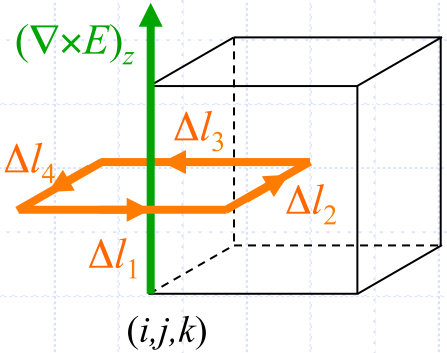


Fig. c



The Yee-Visscher Algorithm

A Finite Difference Time Domain (**FDTD**) scheme.

1. We label
   - c: the cube
   - f_c: the 6 faces of cube c
   - e: the 12 edges of cube c
   - e_l: the 4 edges of the face f_c
   - f_e: the 4 faces that share the same edge e
2. The discrete form of continuity Eq. (12) is

$$\rho\left(c,t+\frac{\Delta t}{2}\right)-\rho\left(c,t-\frac{\Delta t}{2}\right)=-\frac{\Delta t}{\Delta l}\sum_{f_c=1}^{6} j\left(f_c,t\right)$$

The Yee-Visscher Algorithm

3. The discrete form of Faraday's law, Eq. (8) is

$$E\left(f_c, t+\tfrac{\Delta t}{2}\right) - E\left(f_c, t-\tfrac{\Delta t}{2}\right) = \Delta t\left[\nabla\times\mathbf{B} - 4\pi j\left(f_c, t\right)\right]$$

$$= \Delta t\left[\frac{1}{\Delta l}\sum_{e_f=1}^{4} B\left(e_f, t\right) - 4\pi j\left(f_{c,} t\right)\right]$$

4. The discrete form of Ampere's law, Eq. (9) with displacement current is

$$B(e_f, t+\Delta t) - B(e_f, t) = -\frac{\Delta t}{\Delta l}\sum_{f_e=1}^{4} E(f_e, t+\frac{\Delta t}{2})$$

5. The stability requirement is $c\Delta t \le \Delta l/\sqrt{3}$

Notes on the Yee-Vischer Algorithm

- **E** and ρ are defined at different times than **j** and **B**. This "***half-step***" approach leads to well behaved equations that are stable over a range of parameters. (Leap-frog)
- We need to specify the boundary conditions. The easiest method is to use ***fixed boundary conditions*** such that the fields *vanish* at the edges of the lattice. Certainly, this leads to ***non-physical reflections off the edges***, and a variety of approaches have been proposed. …
- The most difficult part of the method is specifying the initial conditions since we cannot simply place a charge somewhere. The reason is that the ***speed of light c*** is finite. The simplest initial condition corresponds to no charge or current before $t = 0$ and then turn on current(s) afterwards.

A Sample Program (see text book)

- Program ***maxwell*** solves Maxwell's equations with a free boundary conditions in the *absence of charge density*.
- A steady current loop is turned on at $t = 0$ and left on.

Lecture 12 Review & Required

- Electric field lines, equipotential surface.
- **Numerical solution of Laplace's equation**.
  - **Discrete form and the average theorem.**
  - **Jacobi and Gauss-Seidel relaxation methods.**
  - **Random walk algorithm, Green's function.**
- Maxwell's equations, the Yee-Vischer algorithm.
- **Program**: *fieldlines*, ***laplace***, *maxwell*.
- *Moving charge, radiation (not discussed)*.