Poisson's equation can also be solved using the random walk method. In this case the potential is given by

$$V(x, y) = \frac{1}{n}\sum_{\alpha} V(\alpha) + \frac{\pi \Delta x \Delta y}{n} \sum_{i,\alpha} \rho(x_{i,\alpha}, y_{i,\alpha}), \tag{10.27}$$

where α labels the walker and i labels the site visited by the walker. That is, each time a walker is at site i, we add the charge density at that site to the second sum in (10.27).

*10.7 ■ FIELDS DUE TO MOVING CHARGES

The fact that accelerating charges radiate electromagnetic waves is one of the more important results in the history of physics. In this section we discuss a numerical algorithm for computing the electric and magnetic fields due to the motion of charged particles. The algorithm is very general, but requires some care in its application.

To understand the algorithm, we need a few results that can be conveniently found in Feynman's lectures. We begin with the fact that the scalar potential at the observation point $\mathbf{R}$ due to a stationary particle of charge q is

$$V(\mathbf{R}) = \frac{q}{|\mathbf{R} - \mathbf{r}|}, \tag{10.28}$$

where $\mathbf{r}$ is the position of the charged particle. The electric field is given by

$$\mathbf{E}(\mathbf{R}) = -\frac{\partial V(\mathbf{R})}{\partial \mathbf{R}}, \tag{10.29}$$

where $\partial V(\mathbf{R})/\partial \mathbf{R}$ is the gradient with respect to the coordinates of the observation point. (Note that our notation for the observation point differs from that used in other sections of this chapter.) How do the relations (10.28) and (10.29) change when the particle is moving? We might guess that because it takes a finite time for the disturbance due to a charge to reach the point of observation, we should modify (10.28) by writing

$$V(\mathbf{R}) \stackrel{?}{=} \frac{q}{r_{\rm ret}}, \tag{10.30}$$

where

$$r_{\rm ret} = |\mathbf{R} - \mathbf{r}(t_{\rm ret})|. \tag{10.31}$$

The quantity $r_{\rm ret}$ is the separation of the charged particle from the observation point $\mathbf{R}$ at the retarded time $t_{\rm ret}$. The latter is the time at which the particle was at $\mathbf{r}(t_{\rm ret})$ such that a disturbance starting at $\mathbf{r}(t_{\rm ret})$ and traveling at the speed of light would reach $\mathbf{R}$ at time t; $t_{\rm ret}$ is given by the implicit equation

$$t_{\rm ret} = t - \frac{r_{\rm ret}(t_{\rm ret})}{c}, \tag{10.32}$$

where t is the observation time and c is the speed of light.

Although the above reasoning is plausible, the relation (10.30) is not quite correct (cf. Feynman et al. for a derivation of the correct result). We need to take into account

that the potential due to the charge is a maximum if the particle is moving toward the observation point and a minimum if it is moving away. The correct result can be written as

$$V(\mathbf{R}, t) = \frac{q}{r_{\rm ret}(1 - \hat{\mathbf{r}}_{\rm ret} \cdot \mathbf{v}_{\rm ret}/c)}, \tag{10.33}$$

where

$$\mathbf{v}_{\rm ret} = \frac{d\mathbf{r}(t)}{dt}\Big|_{t=t_{\rm ret}}, \tag{10.34}$$

and $\hat{\mathbf{r}} = \mathbf{r}/r$.

To find the electric field of a moving charge, we recall that the electric field is related to the time rate of change of the magnetic flux. Hence, we expect that the total electric field at the observation point $\mathbf{R}$ has a contribution due to the magnetic field created by the motion of the charge. We know that the magnetic field due to a moving charge is given by

$$\mathbf{B} = \frac{1}{c}\frac{q\mathbf{v} \times \mathbf{r}}{r^3}. \tag{10.35}$$

If we define the vector potential $\mathbf{A}$ as

$$\mathbf{A} = \frac{q}{r}\frac{\mathbf{v}}{c}, \tag{10.36}$$

we can express $\mathbf{B}$ in terms of $\mathbf{A}$ as

$$\mathbf{B} = \nabla \times \mathbf{A}. \tag{10.37}$$

As we did for the scalar potential V, we argue that the correct formula for $\mathbf{A}$ is

$$\mathbf{A}(\mathbf{R}, t) = q\frac{\mathbf{v}_{\rm ret}/c}{r_{\rm ret}(1 - \hat{\mathbf{r}}_{\rm ret} \cdot \mathbf{v}_{\rm ret}/c)}. \tag{10.38}$$

Equations (10.33) and (10.38) are known as the Liénard–Wiechert form of the potentials.

The contribution to the electric field $\mathbf{E}$ from V and $\mathbf{A}$ is given by

$$\mathbf{E} = -\nabla V - \frac{1}{c}\frac{\partial \mathbf{A}}{\partial t}. \tag{10.39}$$

The derivatives in (10.39) are with respect to the observation coordinates. The difficulty associated with calculating these derivatives is that the potentials depend on $t_{\rm ret}$, which in turn depends on $\mathbf{R}$, $\mathbf{r}$, and t. The result can be expressed as

$$\mathbf{E}(\mathbf{R}, t) = \frac{q r_{\rm ret}}{(\mathbf{r}_{\rm ret} \cdot \mathbf{u}_{\rm ret})^3}[\mathbf{u}_{\rm ret}(c^2 - v_{\rm ret}^2) + \mathbf{r}_{\rm ret} \times (\mathbf{u}_{\rm ret} \times \mathbf{a}_{\rm ret})], \tag{10.40}$$

where

$$\mathbf{u}_{\rm ret} \equiv c\hat{\mathbf{r}}_{\rm ret} - \mathbf{v}_{\rm ret}. \tag{10.41}$$

The acceleration of the particle is given by $\mathbf{a}_{\rm ret} = d\mathbf{v}(t)/dt|_{t=t_{\rm ret}}$. We can also show using (10.37) that the magnetic field $\mathbf{B}$ is given by

$$\mathbf{B} = \hat{\mathbf{r}}_{\rm ret} \times \mathbf{E}. \tag{10.42}$$

The above discussion is not rigorous but leads to the correct expressions for **E** and **B**. We suggest that you accept (10.40) and (10.42) in the same spirit as you accepted Coulomb's law and the Biot–Savart law. All of classical electrodynamics can be reduced to (10.40) and (10.42) if we assume that the sources of all fields are charges, and all electric currents are due to the motion of charged particles. Note that (10.40) and (10.42) are consistent with the special theory of relativity and reduce to known results in the limit of stationary charges and steady currents.

Although (10.40) and (10.42) are deceptively simple (we do not even have to solve any differential equations), it is difficult to calculate the fields analytically even if the position of a charged particle is an analytic function of time. The difficulty is that we must find the retarded time $t_{\rm ret}$ from (10.32) for each observation position **R** and time t. For example, consider a charged particle whose motion is sinusoidal, that is, $x(t_{\rm ret}) = A\cos\omega t_{\rm ret}$. To calculate the fields at the position $\mathbf{R} = (X, Y, Z)$ at time t, we need to solve the following transcendental equation for $t_{\rm ret}$:

$$t_{\rm ret} = t - \frac{r_{\rm ret}}{c} = t - \frac{1}{c}\sqrt{(X - A\cos^2\omega t_{\rm ret})^2 + Y^2 + Z^2}. \tag{10.43}$$

The solution of (10.43) can be expressed as a root finding problem for which we need to find the zero of the function $f(t_{\rm ret})$:

$$f(t_{\rm ret}) = t - t_{\rm ret} - \frac{r_{\rm ret}}{c}. \tag{10.44}$$

There are various ways of finding the solution for the retarded time. For example, if the motion of the charges is given by an analytic expression, we can employ *Newton's method* or the *bisection method*. Because we will store the path of the charged particle, we use a simple method that looks for a change in the sign of the function $f(t_{\rm ret})$ along the path. First find a value t_a such that $f(t_a) > 0$ and another value t_b such that $f(t_b) < 0$. Because $f(t_{\rm ret})$ is continuous, there is a value of $t_{\rm ret}$ in the interval $t_a < t_{\rm ret} < t_b$ such that $f(t_{\rm ret}) = 0$. This technique is used in the `RadiatingCharge` class shown in Listing 10.6. Note that the particle's path is a sinusoidal oscillation specified in method `evaluate`. The name `evaluate` is used because `RadiatingCharge` implements the `Function` interface, which requires an `evaluate` method. What is the maximum velocity for a particle that moves according to this function?

Listing 10.6 The `RadiatingCharge` class computes the radiating electric and magnetic fields using Liénard–Wiechert potentials.

```
package org.opensourcephysics.sip.ch10;
import java.awt.Graphics;
import org.opensourcephysics.display.*;
import org.opensourcephysics.numerics.*;

public class RadiatingCharge implements Drawable, Function {
   Circle circle = new Circle(0, 0, 5);
   double t = 0;                           // time
   double dt = 0.5;                        // time step
   // current number of points in storage
   int numPts = 0;
   double[][] path = new double[3][1024]; // storage for t,x,y
   double[] r = new double[2];
```



```
   double[] v = new double[2];
   double[] u = new double[2];
   double[] a = new double[2];
   // maximum velocity for charge in units where c = 1
   double vmax;

   public RadiatingCharge() {
      resetPath();
   }

   private void resizePath() {
      int length = path[0].length;
      if(length>32768) { // drop half the points
         System.arraycopy(path[0], length/2, path[0], 0, length/2);
         System.arraycopy(path[1], length/2, path[1], 0, length/2);
         System.arraycopy(path[2], length/2, path[2], 0, length/2);
         numPts = length/2;
         return;
      }
      double[][] newPath = new double[3][2*length]; // new path
      System.arraycopy(path[0], 0, newPath[0], 0, length);
      System.arraycopy(path[1], 0, newPath[1], 0, length);
      System.arraycopy(path[2], 0, newPath[2], 0, length);
      path = newPath;
   }

   void step() {
      t += dt;
      if(numPts>=path[0].length) {
         resizePath();
      }
      path[0][numPts] = t;
      path[1][numPts] = evaluate(t); // x position of charge
      path[2][numPts] = 0;
      numPts++;
   }

   void resetPath() {
      numPts = 0;
      t = 0;
      path = new double[3][1024];     // storage for t,x,y
      path[0][numPts] = t;
      path[1][numPts] = evaluate(t); // x position of charge
      path[2][numPts] = 0;
      numPts++; // initial position has been added
   }

   void electrostaticField(double x, double y, double[] field) {
      double dx = x-path[1][0];
      double dy = y-path[2][0];
      double r2 = dx*dx+dy*dy;
      double r3 = r2*Math.sqrt(r2);
      double ex = dx/r3;
      double ey = dy/r3;
      field[0] = ex;
      field[1] = ey;
```

```
      field[2] = 0; // magnetic field
   }

   double dsSquared(int i, double t, double x, double y) {
      double dt = t-path[0][i];
      double dx = x-path[1][i];
      double dy = y-path[2][i];
      return dx*dx+dy*dy-dt*dt;
   }

   void calculateRetardedField(double x, double y, double[] field) {
      int first = 0;
      int last = numPts-1;
      double ds_first = dsSquared(first, t, x, y);
      if(ds_first>=0) { // field has not yet propagated to the location
         electrostaticField(x, y, field);
         return;
      }
      while((ds_first<0)&&(last-first)>1) {
         int i = first+(last-first)/2; // bisect the interval
         double ds = dsSquared(i, t, x, y);
         if(ds<=0) {
            ds_first = ds;
            first = i;
         } else {
            last = i;
         }
      }
      double t_ret = path[0][first]; // time where ds changes sign
      r[0] = x-evaluate(t_ret); // evaluate x at retarded time
      r[1] = y;                 // evaluate y at retarded time
      // derivative of x at retarded time
      v[0] = Derivative.centered(this, t_ret, dt);
      v[1] = 0;                 // derivative of y at retarded time
      // acceleration of x at retarded time
      a[0] = Derivative.second(this, t_ret, dt);
      a[1] = 0;                 // acceleration of y at retarded time
      double rMag = Vector2DMath.mag2D(r); // magnitdue of r
      u[0] = r[0]/rMag-v[0];
      u[1] = r[1]/rMag-v[1];
      double r_dot_u = Vector2DMath.dot2D(r, u);
      double k = rMag/r_dot_u/r_dot_u/r_dot_u;
      // u cross a is perpendicular to plane of motion
      double u_cross_a = Vector2DMath.cross2D(u, a);
      double[] temp = {r[0], r[1]};
      temp = Vector2DMath.crossZ(temp, u_cross_a); // r cross u
      // (c*c - v*v) where c = 1
      double c2v2 = 1-Vector2DMath.dot2D(v, v);
      double ex = k*(u[0]*c2v2+temp[0]);
      double ey = k*(u[1]*c2v2+temp[1]);
      field[0] = ex;
      field[1] = ey;
      field[2] = k*Vector2DMath.cross2D(temp, r)/rMag;
   }

   public void draw(DrawingPanel panel, Graphics g) {
```



```
      circle.setX(evaluate(t));
      circle.draw(panel, g); // draw the charged particle on the screen
   }

   public double evaluate(double t) {
      return 5*Math.cos(t*vmax/5.0);
   }
}
```

The `RadiatingCharge` class computes the electric field due to an oscillating charge using the Liénard–Wiechert potentials. We choose units such that the speed of light $c = 1$. As the charge moves, it stores its ith data point in a two-dimensional array `path[3][i]` containing the time, its x-position, and its y-position. To find the retarded time at the position (x, y), we use the `dsSquared` method to compute the square of the space-time interval between the given location and points along the path. The square of the space-time separation is defined as

$$\Delta s^2 = \Delta x^2 + \Delta y^2 - c^2\Delta t^2, \tag{10.45}$$

where $\Delta x = x - x_{\rm path}$, $\Delta y = y - y_{\rm path}$, and $\Delta t = t - t_{\rm path}$. The last point on the path contains the current position of the charge so Δs^2 must be positive because Δt is zero (unless the charge is at the observation point (x, y) in which case Δs^2 is zero and the field is infinite due to the $1/r^2$ dependence). The `calcRetardedField` method evaluates Δs^2 at the first point in the trajectory to determine if it is negative. We assume the charge was stationary for $t < 0$ and compute the electrostatic field if Δs^2 is positive at the trajectory's first point where $t = 0$. If Δs^2 is negative at the trajectory's first point, we repeatedly bisect the path into smaller and smaller segments while checking to see if Δs^2 remains negative at the beginning of the segment and positive at the end. In this way we can find the retarded time when we have a path segment bounded by two data points. Note that the `RadiatingCharge` class uses the `Vector2DMath` class to perform the necessary vector arithmetic. This helper class is not listed but is available in ch10 code package.

The `RadiatingEFieldApp` program is shown in Listing 10.7. It displays the electric field in the xy-plane using a `Vector2DFrame`. The `calculateFields` method computes the retarded field at every grid point. The simulation's `doStep` method invokes this method after it moves the charge.

Listing 10.7 The `RadiatingEFieldApp` program computes the radiating electric and magnetic fields using Liénard–Wiechert potentials.

```
package org.opensourcephysics.sip.ch10;
import org.opensourcephysics.controls.*;
import org.opensourcephysics.frames.Vector2DFrame;

public class RadiatingEFieldApp extends AbstractSimulation {
   Vector2DFrame frame = new Vector2DFrame("x", "y", "Electric field");
   RadiatingCharge charge = new RadiatingCharge();
   int gridSize;      // linear dimension of grid used to compute fields
   double[][][] Exy; // x and y components of electric field
   double xmin = -20, xmax = 20, ymin = -20, ymax = 20;

   public RadiatingEFieldApp() {
      frame.setPreferredMinMax(xmin, xmax, ymin, ymax);
```

```
      frame.setZRange(false, 0, 0.2);
      frame.addDrawable(charge);
   }

   public void initialize() {
      gridSize = control.getInt("size");
      Exy = new double[2][gridSize][gridSize];
      // maximum speed of charge
      charge.vmax = control.getDouble("vmax");
      charge.dt = control.getDouble("dt");
      frame.setAll(Exy);
      initArrays();
   }

   private void initArrays() {
      charge.resetPath();
      calculateFields();
   }

   private void calculateFields() {
      double[] fields = new double[3]; // Ex, Ey, Bz
      for(int i = 0;i<gridSize;i++) {
         for(int j = 0;j<gridSize;j++) {
            // x location where we calculate the field
            double x = frame.indexToX(i);
            // y location where we calculate the field
            double y = frame.indexToY(j);
            // return the retarded time
            charge.calculateRetardedField(x, y, fields);
            Exy[0][i][j] = fields[0]; // Ex
            Exy[1][i][j] = fields[1]; // Ey
         }
      }
      frame.setAll(Exy);
   }

   public void reset() {
      control.setValue("size", 31);
      control.setValue("dt", 0.5);
      control.setValue("vmax", 0.9);
      initialize();
   }

   protected void doStep() {
      charge.step();
      calculateFields();
   }

   public static void main(String[] args) {
      SimulationControl.createApp(new RadiatingEFieldApp());
   }
}
```


Problem 10.19 Field lines from an accelerating charge

(a) Read the code for `RadiatingEFieldApp` carefully to understand the correspondence between the program and the analytic results, (10.40) and (10.42), discussed in the text.

(b) Describe qualitatively the nature of the electric and magnetic fields from an oscillating point charge. How does the electric field differ from that of a static charge at the origin? What happens as the speed increases? The physics breaks down if the maximum speed is greater than c. Does the algorithm break down? Explain.

(c) Modify `RadiatingEFieldApp` to show the z-component of the magnetic field in the xy-plane using a `Scalar2DFrame`.

(d) Modify the program to observe a charge moving with uniform circular motion about the origin. What happens as the speed of the charge approaches the speed of light? ■

Problem 10.20 Spatial dependence of the radiating fields

(a) As waves propagate from an accelerating point source, the total power that passes through a spherical surface of radius R remains constant. Because the surface area is proportional to R^2, the power per unit area or intensity is proportional to $1/R^2$. Also, because the intensity is proportional to E^2, we expect that $E \propto 1/R$ far from the source. Modify the program to verify this result for a charge that is oscillating along the x-axis according to $x(t) = 0.2\cos t$. Plot $|E|$ as a function of the observation time t for a fixed position, such as $\mathbf{R} = (10, 10, 0)$. The field should oscillate in time. Find the amplitude of this oscillation. Next double the distance of the observation point from the origin. How does the amplitude depend on R?

(b) Repeat part (a) for several directions and distances. Generate a polar diagram showing the amplitude as a function of angle in the xy-plane. Is the radiation greatest along the line in which the charge oscillates? ■

Problem 10.21 Fields from a charge moving at constant velocity

(a) Use `RadiationApp` to calculate $\mathbf{E}$ due to a charged particle moving at constant velocity toward the origin, for example, $x(t_{\rm ret}) = 1 - 2t_{\rm ret}$. Take a snapshot at $t = 0.5$ and compare the field lines with those you expect from a stationary charge.

(b) Modify `RadiationApp` so that $x(t_{\rm ret}) = 1 - 2t_{\rm ret}$ for $t_{\rm ret} < 0.5$ and $x(t_{\rm ret}) = 0$ for $t_{\rm ret} > 0.5$. Describe the field lines for $t > 0.5$. Does the particle accelerate at any time? Is there any radiation? ■

Problem 10.22 Frequency dependence of an oscillating charge

(a) The radiated power at any point in space is proportional to E^2. Plot $|E|$ versus time at a fixed observation point (for example, $X = 10, Y = Z = 0$) and calculate the frequency dependence of the amplitude of $|E|$ due to a charge oscillating at the frequency ω. It is shown in standard textbooks that the power associated with radiation from an oscillating dipole is proportional to ω^4. How does the ω-dependence that you measured compare to that for dipole radiation? Repeat for a much bigger value of R and explain any differences.

(b) Repeat part (a) for a charge moving in a circle. Are there any qualitative differences? ■

*10.8 ■ MAXWELL'S EQUATIONS

In Section *10.7 we found that accelerating charges produce electric and magnetic fields that depend on position and time. We now investigate the direct relation between changes in **E** and **B** given by the differential form of Maxwell's equations:

$$\frac{\partial \mathbf{B}}{\partial t} = -\frac{1}{c}\nabla \times \mathbf{E} \tag{10.46}$$

$$\frac{\partial \mathbf{E}}{\partial t} = c\nabla \times \mathbf{B} - 4\pi \mathbf{j}, \tag{10.47}$$

where **j** is the electric current density. We can regard (10.46) and (10.47) as the basis of electrodynamics. In addition to (10.46) and (10.47), we need the relation between **j** and the charge density ρ that expresses the conservation of charge:

$$\frac{\partial \rho}{\partial t} = -\nabla \cdot \mathbf{j}. \tag{10.48}$$

A complete description of electrodynamics requires (10.46), (10.47), (10.48), and the initial values of all currents and fields.

For completeness, we obtain the Maxwell's equations that involve $\nabla \cdot \mathbf{B}$ and $\nabla \cdot \mathbf{E}$ by taking the divergence of (10.46) and (10.47), substituting (10.48) for $\nabla \cdot \mathbf{j}$, and then integrating over time. If the initial fields are zero, we obtain (using the relation $\nabla \cdot (\nabla \times \mathbf{a}) = 0$ for any vector **a**)

$$\nabla \cdot \mathbf{E} = 4\pi\rho \tag{10.49}$$

$$\nabla \cdot \mathbf{B} = 0. \tag{10.50}$$

If we introduce the electric and magnetic potentials, it is possible to convert the first-order equations (10.46) and (10.47) to second-order differential equations. However, the familiar first-order equations are better suited for numerical analysis. To solve (10.46) and (10.47) numerically, we need to interpret the curl and divergence of a vector. As its name implies, the curl of a vector measures how much the vector twists around a point. A coordinate free definition of the curl of an arbitrary vector **W** is

$$(\nabla \times \mathbf{W}) \cdot \hat{\mathbf{S}} = \lim_{S \to 0} \frac{1}{S} \oint_C \mathbf{W} \cdot d\mathbf{l}, \tag{10.51}$$

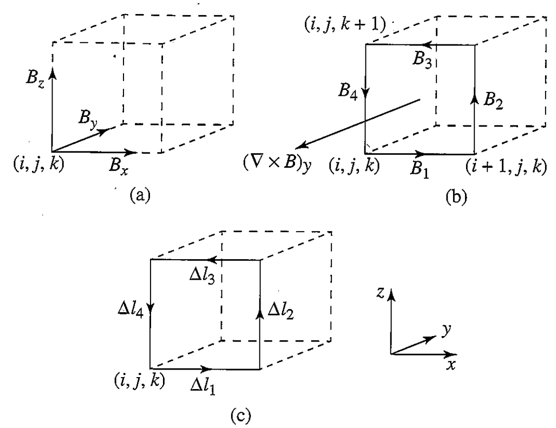

Figure 10.6 Calculation of the curl of **B** defined on the edges of a cube. (a) The edge vector **B** associated with cube (i, j, k). (b) The components B_i along the edges of the front face of the cube. $B_1 = B_x(i, j, k)$, $B_2 = B_z(i+1, j, k)$, $B_3 = -B_x(i, j, k+1)$, and $B_4 = -B_z(i, j, k)$. (c) The vector components $\Delta \mathbf{l}_i$ on the edges of the front face. (The y-component of $\nabla \times \mathbf{B}$ defined on the face points in the negative y direction.)

where **S** is the area of any surface bordered by the closed curve C, and $\hat{\mathbf{S}}$ is a unit vector normal to the surface S.

Equation (10.51) gives the component of $\nabla \times \mathbf{W}$ in the direction of $\hat{\mathbf{S}}$ and suggests a way of computing the curl numerically. We divide space into cubes of linear dimension Δl. The rectangular components of **W** can be defined either on the edges or on the faces of the cubes. We compute the curl using both definitions. We first consider a vector **B** that is defined on the edges of the cubes so that the curl of **B** is defined on the faces. (We use the notation **B** because we will find that it is convenient to define the magnetic field in this way.) Associated with each cube is one edge vector and one face vector. We label the cube by the coordinates corresponding to its lower left front corner; the three components of **B** associated with this cube are shown in Figure 10.6a. The other edges of the cube are associated with B vectors defined at neighboring cubes.

The discrete version of (10.51) for the component of $\nabla \times \mathbf{B}$ defined on the front face of the cube (i, j, k) is

$$(\nabla \times \mathbf{B}) \cdot \hat{\mathbf{S}} = \frac{1}{(\Delta l)^2} \sum_{i=1}^{4} B_i \Delta l_i, \tag{10.52}$$

where $S = (\Delta l)^2$, and B_i and l_i are shown in Figures 10.6b and 10.6c, respectively. Note that two of the B_i are associated with neighboring cubes.

The components of a vector can also be defined on the faces of the cubes. We call this vector **E** because it will be convenient to define the electric field in this way. In Figure 10.7a we show the components of **E** associated with the cube (i, j, k). Because **E** is normal to a cube face, the components of $\nabla \times \mathbf{E}$ lie on the edges. The components E_i and l_i are shown in Figures 10.7b and 10.7c, respectively. The form of the discrete version of $\nabla \times \mathbf{E}$