```
   void damping() {
      for(int i = 0;i<size;i++) {
         for(int j = 0;j<size;j++) {
            // w used to index cell near boundary subject to damping
            for(int w = 0;w<4;w++) {
               for(int comp = 0;comp<3;comp++) {
                  E[comp][w][i][j] -= dampingCoef*E[comp][w][i][j];
                  E[comp][size-w-1][i][j] -=
                        dampingCoef*E[comp][size-w-1][i][j];
                  E[comp][i][w][j] -= dampingCoef*E[comp][i][w][j];
                  E[comp][i][size-w-1][j] -=
                        dampingCoef*E[comp][i][size-w-1][j];
                  E[comp][i][j][w] -= dampingCoef*E[comp][i][j][w];
                  E[comp][i][j][size-w-1] -=
                        dampingCoef*E[comp][i][j][size-w-1];
                  B[comp][w][i][j] -= dampingCoef*B[comp][w][i][j];
                  B[comp][size-w-1][i][j] -=
                        dampingCoef*B[comp][size-w-1][i][j];
                  B[comp][i][w][j] -= dampingCoef*B[comp][i][w][j];
                  B[comp][i][size-w-1][j] -=
                        dampingCoef*B[comp][i][size-w-1][j];
                  B[comp][i][j][w] -= dampingCoef*B[comp][i][j][w];
                  B[comp][i][j][size-w-1] -=
                        dampingCoef*B[comp][i][j][size-w-1];
               }
            }
         }
      }
   }
}
```

Listing 10.9 The `MaxwellApp` program computes and displays the electric field by solving Maxwell's equations.

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

public class MaxwellApp extends AbstractSimulation {
   Vector2DFrame frame = new Vector2DFrame("x", "y",
        "EField in xy plane");
   int size;
   Maxwell maxwell;
   // x and y components of E for middle plane in z direction
   double[][][] Exy;

   public MaxwellApp() {
      frame.setZRange(false, 0, 1.0);
   }

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



```
   public void initialize() {
      size = control.getInt("size");
      Exy = new double[2][size][size];
      maxwell = new Maxwell(size);
      frame.setAll(Exy);
      frame.setPreferredMinMax(0, Maxwell.dl*size, Maxwell.dl*size, 0);
      plotField();
   }

   protected void doStep() {
      maxwell.doStep();
      plotField();
      frame.setMessage("t="+decimalFormat.format(maxwell.t));
   }

   void plotField() {
      double[][][][] E = maxwell.E; // electric field
      int mid = size/2;
      for(int i = 0;i<size;i++) {
         for(int j = 0;j<size;j++) {
            Exy[0][i][j] = E[i][j][mid][0]; // Ex
            Exy[1][i][j] = E[i][j][mid][1]; // Ey
         }
      }
      frame.setAll(Exy);
   }

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

The `MaxwellApp` program shows the electric field in the xy-plane. The xy-components of the electric field are represented by arrows whose length is fixed and whose color indicates the field magnitude at each position where the field is defined.

Problem 10.23 Fields from a current loop

(a) A steady current in the middle of the xy-plane is turned on at $t = 0$ and left on for one time unit. Before running the program, predict what you expect to see. Compare your expectations with the results of the simulation. Use $\Delta t = 0.03$, $\Delta l = 0.1$, and take the number of cubes in each direction to be `n(1)` = `n(2)` = `n(3)` = 8.

(b) Add a plot of the magnetic field. Where should the viewing plane be placed to produce the best visualization? How should the plane be oriented? Predict what you expect to see before you run the simulation.

(c) Verify the stability requirement (10.58) by running your program with $\Delta t = 0.1$ and $\Delta l = 0.1$. Then try $\Delta t = 0.05$ and $\Delta l = \Delta t\sqrt{3}$. What happens to the results in part (a) if the stability requirement is not satisfied?

(d) Modify the current density in part (a) so that $\mathbf{j}$ oscillates sinusoidally. What happens to the electric and magnetic field vectors?