```
      if(xmlControl.failedToRead()) {
         control.println("failed to read: "+filename);
      } else {
         // gets the datasets in the xml file
         Iterator it = xmlControl.getObjects(Dataset.class,
              false).iterator();
         while(it.hasNext()) {
            Dataset dataset = (Dataset) it.next();
            if(dataset.getName().equals("magnetization")) {
               magnetization = dataset.getYPoints();
            }
            if(dataset.getName().equals("energy")) {
               energy = dataset.getYPoints();
            }
         }
         numberOfPoints = magnetization.length;
         control.println("Reading: "+filename);
         control.println("Number of points = "+numberOfPoints);
      }
      calculate();
      plotFrame.repaint();
   }

   public void computeCorrelation(int tauMax) {
      plotFrame.clearData();
      double energyAccumulator = 0;
      double magnetizationAccumulator = 0;
      double energySquaredAccumulator = 0;
      double magnetizationSquaredAccumulator = 0;
      for(int t = 0;t<numberOfPoints;t++) {
         energyAccumulator += energy[t];
         magnetizationAccumulator += magnetization[t];
         energySquaredAccumulator += energy[t]*energy[t];
         magnetizationSquaredAccumulator +=
              magnetization[t]*magnetization[t];
      }
      double averageEnergySquared =
           Math.pow(energyAccumulator/numberOfPoints, 2);
      double averageMagnetizationSquared =
           Math.pow(magnetizationAccumulator/numberOfPoints, 2);
      // compute normalization factors
      double normE = (energySquaredAccumulator/numberOfPoints)-
                      averageEnergySquared;
      double normM = (magnetizationSquaredAccumulator/numberOfPoints)-
                      averageMagnetizationSquared;
      for(int tau = 1;tau<=tauMax;tau++) {
         double c_MAccumulator = 0;
         double c_EAccumulator = 0;
         int counter = 0;
         for(int t = 0;t<numberOfPoints-tau;t++) {
            c_MAccumulator += magnetization[t]*magnetization[t+tau];
            c_EAccumulator += energy[t]*energy[t+tau];
            counter++;
         }
         // correlation function defined so that c(0) = 1
         // and c(infinity) -> 0
```



```
            plotFrame.append(0, tau, ((c_MAccumulator/counter)-
                 averageMagnetizationSquared)/normM);
            plotFrame.append(1, tau, ((c_EAccumulator/counter)-
                 averageEnergySquared)/normE);
         }
         plotFrame.setVisible(true);
      }

      public void reset() {
         control.setValue("Maximum time interval, tau", 20);
         readXMLData();
      }

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

Problem 15.14 Correlation times

(a) As a check on `IsingAutoCorrelatorApp`, use the time series for E given in the text to do a hand calculation of $C_E(t)$ in the way that it is computed in the `computeCorrelation` method.

(b) Use class `IsingAutoCorrelatorApp` to compute the equilibrium values of $C_M(t)$ and $C_E(t)$. Save the values of the magnetization and energy only after the system has reached equilibrium. Estimate the correlation times from the energy and the magnetization correlation functions for $L = 8$, and $T = 3$, $T = 2.3$, and $T = 2$. One way to determine τ is to fit $C(t)$ to the exponential form $C(t) \sim e^{-t/\tau}$. Another way is to define the integrated correlation time as

$$\tau = \sum_{t=1} C(t). \tag{15.25}$$

The sum is cut off at the first negative value of $C(t)$. Are the negative values of $C(t)$ physically meaningful? How does the behavior of $C(t)$ change if you average your results over longer runs? How do your estimates for the correlation times compare with your estimates of the relaxation time found in Problem 15.12? Why would the term "decorrelation time" be more appropriate than "correlation time?" Are the correlation times τ_M and τ_E comparable?

(c) To simulate the relaxation to equilibrium as realistically as possible, we have randomly selected the spins to be flipped. However, if we are interested only in equilibrium properties, it might be possible to save computer time by selecting the spins sequentially. Determine if the correlation time is greater, smaller, or approximately the same if the spins are chosen sequentially rather than randomly. If the correlation time is greater, does it still save CPU time to choose spins sequentially? Why is it not desirable to choose spins sequentially in the one-dimensional Ising model? ■

How can we quantify the accuracy of our measurements, for example, the accuracy of the estimated mean energy? As discussed in Chapter 11, the usual measure of the accuracy