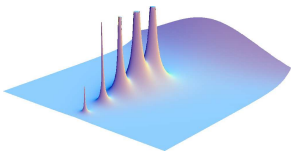

Physics 170

Exercise A

Due 31 January 2005

On 30 January 1996 an earthquake was recorded by the instruments in the HMC seismology lab between 18:00 and 24:00 UT (universal time). The goal of this exercise is for you to download the data file holding the recorded amplitude of motion as a function of time, load the relevant portion into a graphics package, and produce a publication-ready plot of the data.

Numerical values are typically stored internally in a binary (base 2) representation. While easy for the computer to read, binary is not easy for humans to read. We are accustomed to seeing 101 and thinking that it is one more than one hundred, not 5. In many circumstances, therefore, one reads numerical values written in decimal form (e.g., 3.14159) into a program, which must convert the values to a binary internal representation. When it is time for the program to output other numerical quantities, and if these are for direct human consumption or for export to an unknown program, they are typically displayed in standard, decimal representation. Hence, the internal binary representation must be converted to decimal.

There are several reasons why it is inappropriate to store data in a file using a decimal representation. First, the decimal representation almost always takes more space (bytes). Second, the base conversion procedure is computationally expensive. Third, the decimal representation takes a variable number of bytes. Therefore, you cannot simply jump to the hundredth value in the file with a simple byte-offset computation and a repositioning of the file pointer; you must scan all the way from the beginning. Therefore, programs that generate significant quantities of numerical data invariably store the data in binary form, usually with a header block to identify various aspects of the data, including textual comments, descriptions of the number and kind of values in the file, and other descriptive information. Table 1 shows how the decimal number 512 (which is 2^9 or 0x20 in hexadecimal) stored in a two-byte variable would be positioned in memory.

A binary data file named quake.binary has been placed on the course web page. This file contains header information, followed by a succession of 16-bit (2-byte) binary values that represent raw data numbers taken on January 30, 1996 in the HMC seismology lab between 18:00 and 24:00 UT. The organization of the file is shown in Table 2.

The data have been recorded at a rate of 10 samples per second, so the main body of the six-hour file should contain at total of 216,000 samples (432,000 bytes).

Table 1: How to write the number 512 in computer memory. In a big-endian machine (most), the most-significant byte is first (at the lower-numbered memory location), and the least-significant byte is last. In little-endian machines (Intel), the opposite is true. For more information, see http://saeta.physics.hmc.edu/courses/p170/code/endian.php

memory position	big-endian		little-endian	
	hexadecimal	binary	hexadecimal	binary
0x0000	02	00000010	00	00000000
0x0001	00	00000000	02	00000010

Table 2: Structure of the binary file quake.bin.

Byte offset	Contents
0 – 159	textual header information (interesting, but irrelevant for present purposes)
160 – 161	two-byte integer continaing the sample rate, in samples per second
162 – 165	four-byte integer containing the number of 2-byte data points in the rest of the file
166 – 432165	consecutive 2-byte data points
432166 – 432383	padding

An earthquake occurred on the date in question shortly after the 34th minute of data in this file. Your job is to isolate the data between minutes 34 and 37 and to prepare an appropriately scaled and labeled plot of the raw amplitude as a function of time. Please save your data in a text file for possible later use in the course.

This assignment effectively has two parts: converting the data from binary to textual representation and preparing the plot. Hints on writing a computer program in C, C++, or Java, including how to access binary files, may be found on the course webpage. Be sure to read the section on **endians**.

I recommend using Kaleidagraph, Origin, Igor, or similar multipurpose data plotting and analysis package to prepare the final plot. Instructions for their use are available on the department web kiosk.

Warning: Please be sure to have successfully downloaded the data file to your own directory before the weekend, so that we can help you troubleshoot problems.

Some points on writing up your solution

- Please check http://saeta.physics.hmc.edu/courses/p170/documentation.php for general guidelines on preparing exercise and problem solutions.
- In particular, make sure that the comment at the top of your program pro-

vides a general description of the file format and the context for the program, including any inputs the program expects and a description of the output it provides.

- Which do you find clearer? Why?

```
inData.skip( 2 * sampRate * 60 * startMinutes );
inData.skip( 40800 );
```

Proper graph format

In older days, scientists would prepare a rough sketch of a graph and give the data to a graphics department to prepare the final graph for submission to a publisher. Graphics department knew how to set the appropriate aspect ratio, to use the appropriate sized fonts, adjust tick marks, and so forth. The personal computer has democratized the production of scientific graphics, shifting the burden from graphics departments to scientists. The result is that many graphs are less useful and informative than they ought to be.

As you prepare your plot of the earthquake data, please respect the following guidelines:

- Use a 4:3 aspect ratio for the plot area. That is, make the y-axis 75% as long as the x-axis. this is the standard format for graphs in publications. You should use this by default, but feel free to depart from this format should the data require it.
- Mirror axes. That is, include upper and right axes which mirror the bottom and left axes. This makes it easy to lay a ruler across the graph and extract data. Label axes informatively.
- When you have as many data points as in this case, do not show discrete symbols. Use a line style, instead. With a more modest number of points, it is preferable to show discrete points of appropriate size.
- Use an appropriate font size. As a guideline, imagine shrinking the graph down to fit in a 3-inch column in a printed article. Text should be at least 9 points at this scale.
- Use a sensible unit for the time axis. I think minutes are somewhat better here, but seconds are acceptable.
- Put an appropriate number of minior ticks between the major, labelled ones.