Homework 1

Johan Jensen

ABSTRACT

This homework has three parts.

1. Theoretical questions and computations related to digital representation of numbers.
2. Analyzing digital elevation data from the Mount St. Helens area. You will apply histogram equalization to enhance the image.
3. Analyzing seismic reflection data. You will apply an amplitude gain correction to enhance the image.

PREREQUISITES

Completing the computational part of this homework assignment requires

- `Madagascar` software environment available from `http://www.ahay.org/`
- LaTeX environment with `SEGTeX` available from `http://www.ahay.org/wiki/SEGTeX`

To do the assignment on your personal computer, you need to install the required environments. Please ask for help if you don't know where to start.

The homework code is available from the `Madagascar` repository by running

```
svn co http://svn.code.sf.net/p/rsf/code/trunk/book/geo391/hw1
```

DIGITAL REPRESENTATION OF NUMBERS

You can either write your answers to theoretical questions on paper or edit them in the file `hw1/paper.tex`. Please show all the mathematical derivations that you perform.

1. UT's "burnt orange" color is expressed by code `#BF5700`, where each pair of symbols (`BF`, `57`, and `00`) refers to a hexadecimal (base 16) representation of the red, green, and blue components. Convert these numbers to an octal (base 8) and a decimal (base 10) representations.

2. The C program listed below, when compiled and run from the command line, takes a string from the user and prints out the string characters. Modify the program to output ASCII integer codes for each character in the string. What is the ASCII code for the special new line character "`\n`"?

string.c

```c
#include <stdio.h> /* for printf and scanf */

int main(void)
{
    char *s, string[101];

    printf("Input a string:\n");
    scanf("%100s",string);

    /* loop over characters */
    for (s=string; *s != '\0'; s++)
        printf("%c\n",*s);
}
```

3. Write a program that prints out "little" if the computer is little-endian and "big" if the computer is big-endian. Test it on your computer.

4. In the IEEE double-precision floating-point standard, 64 bits (binary digits) are used to represent a real number: 1 bit for the sign, 11 bits for the exponent, and 52 bits for the mantissa. A double-precision normalized non-zero number x can be written in this standard as

$$x = \pm(1.d_1 d_2 \cdots d_{52})_2 \times 2^{n-1023}$$

with $1 \leq n \leq 2046$, and $0 \leq d_k \leq 1$ for $k = 1, 2, \ldots, 52$. What is the largest number that can be expressed in this system?

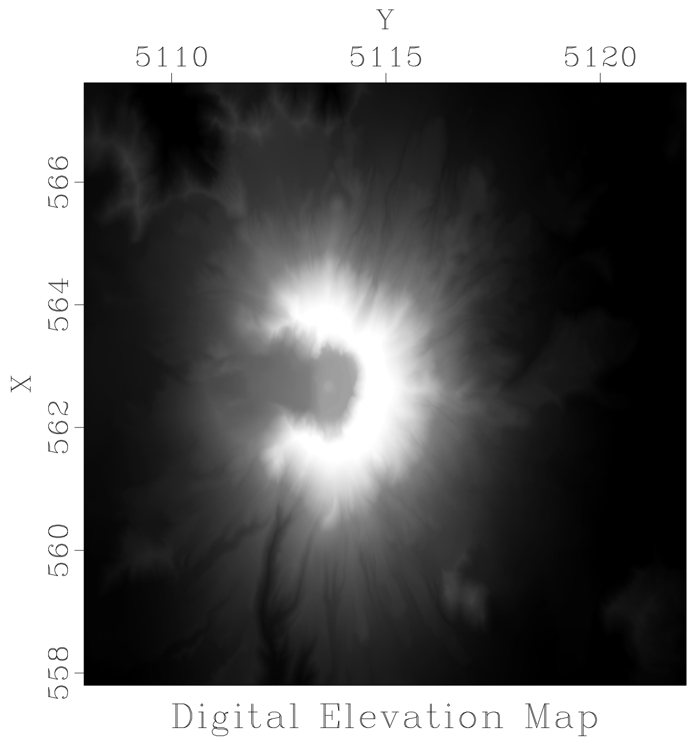

Figure 1: Digital elevation map of Mount St. Helens area.

HISTOGRAM EQUALIZATION

Figure 1 shows a digital elevation map of the Mount St. Helens area in Washington. Start by reproducing this figure on your screen.

1. Change directory to `hw1/dem`
2. Run

   ```
   scons byte.view
   ```

3. Examine the file `byte.rsf` which refers to the byte (unsigned character) numbers which get displayed on the screen.
   (a) Open `byte.rsf` with a text editor to check its contents.
   (b) Run

   ```
   sfin byte.rsf
   ```

   to check the data size and format.
   (c) Run

   ```
   sfattr < byte.rsf
   ```

   to check data attributes. What is the maximum and minimum value? What is the mean value? For an explanation of different attributes, run `sfattr` without input.

Each image has a certain distribution of values (a histogram). The histogram for the elevation map values is shown in Figure 2. When different values in a histogram are not uniformly distributed, the image can have a low contrast. One way of improving the contrast is *histogram equalization*.

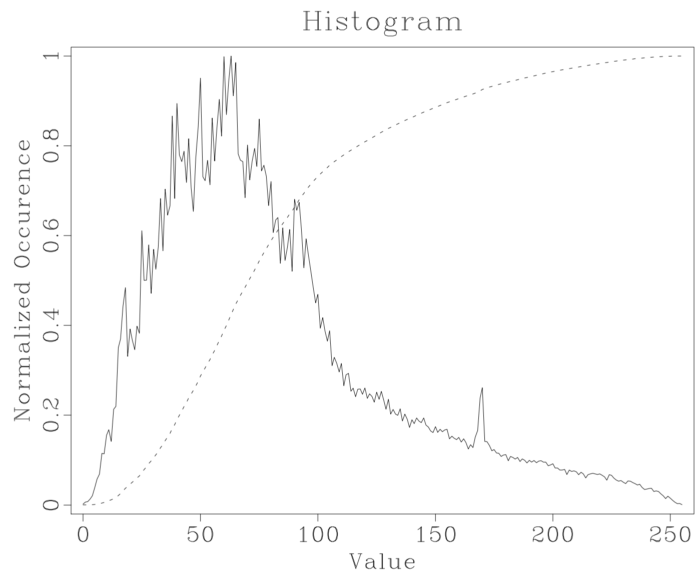

Figure 2: Normalized histogram (solid line) and cumulative histogram (dashed line) of the digital elevation data.

Let $f(x, y)$ be the original image. The equalized image will be $F(x, y)$. Let $h(f)$ be the histogram (probability distribution) of the original image values. Let $H(F)$ be the histogram of the modified image. The mapping of probabilities suggests

$$H(F)\, dF = h(f)\, df \tag{1}$$

or, if we want the modified histogram to be uniform,

$$\frac{dF}{df} = C\, h(f) \tag{2}$$

where C is a constant. Solving equation 2, we obtain the mapping

$$F(f) = f_0 + C \int_{f_0}^{f} h(\phi)\, d\phi \ , \tag{3}$$

where f_0 is the minimum value of f.

The algorithm for histogram equalization consists of the following three steps:

1. Taking an input image $f(x, y)$, compute its histogram $h(f)$.
2. Compute the cumulative histogram $F(f)$ according to equation (3). Choose an appropriate C so that the range of F is the same as the range of f.
3. Map every pixel $f(x, y)$ to the corresponding $F(x, y)$.

Your task:

1. Among the `Madagascar` programs, find a program that implements histogram equalization. **Hint:** you may find the `sfdoc` utility useful.
2. Edit the `SConstruct` file to add histogram equalization. Create a new figure and compare it with Figure 1.
3. Check the effect of equalization by recomputing the histogram in Figure 2 with equalized data. Run

   `scons hist.view`

   to display the figure on your screen.

dem/SConstruct

```
from rsf.proj import *

# Download data
txt = 'st-helens_after.txt'
Fetch(txt,'data',
      server='https://raw.githubusercontent.com',
      top='agile-geoscience/notebooks/master')
Flow('data.asc',txt,'/usr/bin/tail -n +6')

# Convert to RSF format
Flow('data','data.asc',
     '''
     echo in=$SOURCE data_format=ascii_float
     label=Elevation unit=m
     n1=979  o1=557.805  d1=0.010030675 label1=X
     n2=1400 o2=5107.965 d2=0.010035740 label2=Y |
     dd form=native |
     clip2 lower=0 | lapfill grad=y niter=200
     ''')

# Convert to byte form
Flow('byte','data',
     '''
     dd form=native |
     byte pclip=100 allpos=y bias=2231
     ''')

# Display
Result('byte',
```

```
        '''
        grey yreverse=n screenratio=1
        title="Digital Elevation Map"
        ''')

# Histogram
Flow('hist','byte',
     '''
     dd type=float |
     histogram n1=256 o1=0 d1=1 |
     dd type=float
     ''')
Plot('hist',
     'graph label1=Value label2=Occurence title=Histogram')

# Cumulative histogram
Flow('cumu','hist','causint')

Result('hist','hist cumu',
        '''
        cat axis=2 ${SOURCES[1]} | scale axis=1 |
        graph label1=Value label2="Normalized Occurence"
        title=Histogram dash=0,1
        ''')

# ADD HISTOGRAM EQUALIZATION

End()
```

TIME-POWER AMPLITUDE-GAIN CORRECTION

Raw seismic reflection data come in the form of shot gathers $S(x,t)$, where x is the offset (horizontal distance from the receiver to the source) and t is recording time. Raw data are inconvenient for analysis because of rapid amplitude decay of seismic waves. The decay can be compensated by multiplying the data by a gain function. A commonly used function is a power of time. The gain-compensated gather is

$$S_\alpha(x,t) = t^\alpha \, S(x,t) \; . \tag{4}$$

The advantage of the time-power gain is its simplicity and the ability to reverse it by multiplying the data by $t^{-\alpha}$. What value of α should one use? Claerbout (1985) argues in favor of $\alpha = 2$: one factor of t comes from geometrical spreading and the other from scattering attenuation. Your task is to develop an algorithm for finding a better value of α for a given dataset.

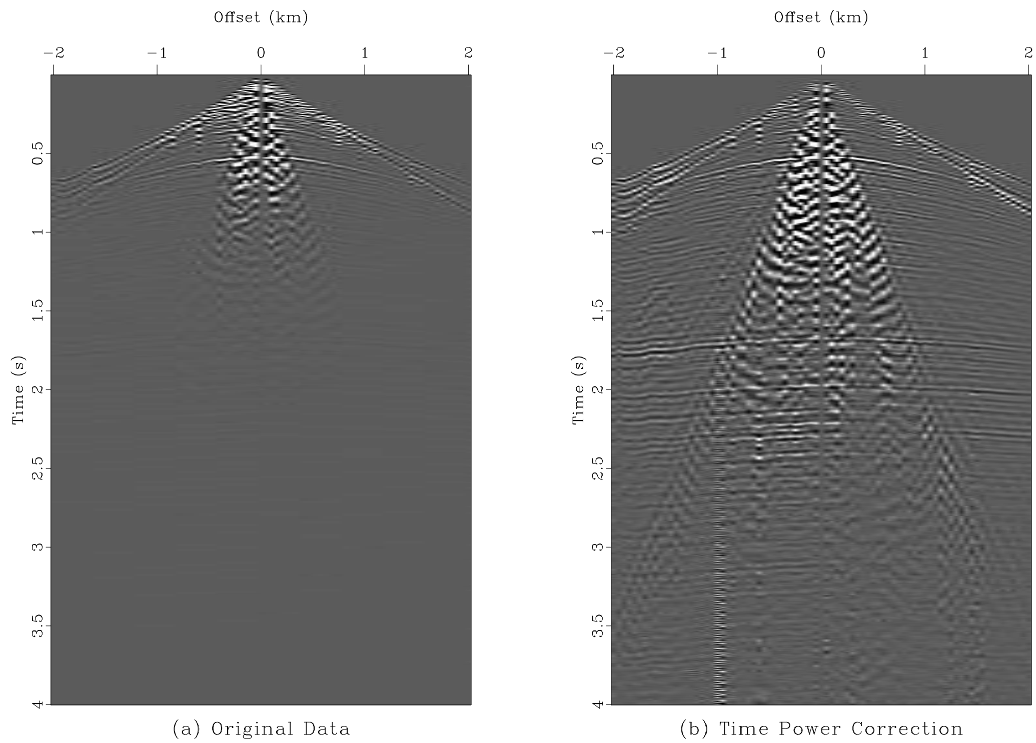

Figure 3: Seismic shot record before and after time-power gain correction.

Figure 3 shows a seismic shot record before and after applying the time-power gain (4) with $\alpha = 2$. Start by reproducing this figure on your screen.

1. Change directory to `hw1/tpow`

2. Run

   ```
   scons tpow.view
   ```

3. Edit the `SConstruct` file. Find where the value of α is specified in this file and try changing it to a different value. Run `scons tpow.view` again to check the result.

4. How can we detect if the distribution of amplitudes after the gain correction is uniform? Suggest a measure (an objective function) that would take $S_\alpha(x,t)$ and produce one number that measures uniformity.

5. By modifying the program `objective.c`, compute your objective function for different values of α and display it in a figure. Does the function appear to have a unique minimum or maximum?

tpow/objective.c

```
#include <rsf.h>

int main(int argc, char* argv[])
{
    int it, nt, ix, nx, ia, na;
    float *trace, *ofunc;
    float a, a0, da, t, t0, dt, s;
    sf_file in, out;

    /* initialization */
    sf_init(argc,argv);
    in = sf_input("in");
    out = sf_output("out");

    /* get trace parameters */
    if (!sf_histint(in,"n1",&nt)) sf_error("Need n1=");
    if (!sf_histfloat(in,"d1",&dt)) dt=1.;
    if (!sf_histfloat(in,"o1",&t0)) t0=0.;

    /* get number of traces */
    nx = sf_leftsize(in,1);

    if (!sf_getint("na",&na)) na=1;
    /* number of alpha values */
    if (!sf_getfloat("da",&da)) da=0.;
    /* increment in alpha */
    if (!sf_getfloat("a0",&a0)) a0=0.;
    /* first value of alpha */

    /* change output data dimensions */
    sf_putint(out,"n1",na);
    sf_putint(out,"n2",1);
    sf_putfloat(out,"d1",da);
    sf_putfloat(out,"o1",a0);

    trace = sf_floatalloc(nt);
    ofunc = sf_floatalloc(na);

    /* initialize */
    for (ia=0; ia < na; ia++) {
        ofunc[ia] = 0.;
    }

```

```
    /* loop over traces */
    for (ix=0; ix < nx; ix++) {

        /* read data */
        sf_floatread(trace,nt,in);

        /* loop over alpha */
        for (ia=0; ia < na; ia++) {
            a = a0+ia*da;

            /* loop over time samples */
            for (it=0; it < nt; it++) {
                t = t0+it*dt;

                /* apply gain t^alpha */
                s = trace[it]*powf(t,a);

                /* !!! MODIFY THE NEXT LINE !!! */
                ofunc[ia] += s*s;
            }
        }
    }

    /* write output */
    sf_floatwrite(ofunc,na,out);

    exit(0);
}
```

6. Suggest an algorithm for finding an optimal value of α by minimizing or maximizing the objective function. Your algorithm should be able to find the optimal value without scanning all possible values. **Hint:** if the objective function is $f(\alpha) = F[S_\alpha(x,t)]$ and

$$f(\alpha) \approx f(\alpha_0) + f'(\alpha_0)\,(\alpha - \alpha_0) + \frac{f''(\alpha_0)}{2}\,(\alpha - \alpha_0)^2 \tag{5}$$

then what is the optimal α?

7. **EXTRA CREDIT** for implementing your algorithm for an automatic estimation of α and testing it on the shot gather from Figure 3.

tpow/SConstruct

```
from rsf.proj import *

# Download data
Fetch('wz.25.H','wz')

# Convert and window
Flow('data','wz.25.H',
     '''
     dd form=native | window min2=-2 max2=2 |
     put label1=Time label2=Offset unit1=s unit2=km
     ''')

# Display
Plot('data','grey title="(a) Original Data"')
Plot('tpow','data',
     'pow pow1=2 | grey title="(b) Time Power Correction" ')

Result('tpow','data tpow','SideBySideAniso')

# Compute objective function
prog = Program('objective.c')
Flow('ofunc','data %s' % prog[0],
     './${SOURCES[1]} na=21 da=0.1 a0=1')

Result('ofunc',
       '''
       scale axis=1 |
       graph title="Objective Function"
       label1=alpha label2= unit1= unit2=
       ''')

End()
```

COMPLETING THE ASSIGNMENT

1. Change directory to `hw1`.
2. Edit the file `paper.tex` in your favorite editor and change the first line to have your name instead of Jensen's.
3. Run

   ```
   sftour scons lock
   ```

   to update all figures.
4. Run

   ```
   sftour scons -c
   ```

   to remove intermediate files.
5. Run

   ```
   scons pdf
   ```

   to create the final document.
6. Submit your result (file `paper.pdf`) on paper or by e-mail.

REFERENCES

Claerbout, J. F., 1985, Imaging the Earth's interior: Blackwell Scientific Publications.