EE257/GP258 **Lab/Homework #9**
Applied Optimization Laboratory Handout #24
Due: Thurs., Mar. 14

Problem 1 – Model fitting methods.

Consider the scattering problem we described in class, where the measured reflectivity as a function of angle has two components:

Surface term, Hagfors' model: $h(\theta, h_{rms}) = \frac{1}{\sqrt{h_{rms}}}\left(\cos^4\theta + \frac{1}{\sqrt{h_{rms}}}\sin^2\theta\right)^{-3/2}$

Volume term: $v(\theta, a) = a\cos\ \theta$

Download the file lab9prob1.dat from the website. This file is a set of measurements of reflectivity as a function of incidence angle. Find the best fitting set of h_{rms} and a to the measurements by several methods.

i. Examine all values of h_{rms} ranging from 0.0001 to 0.1, and a from 0.1 to 10, and find the best fit by computing the error for all cases. Time your code and record the execution time.

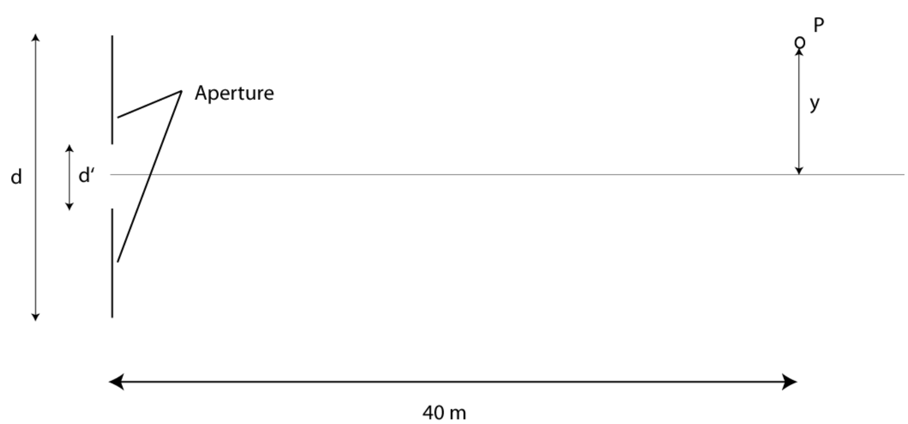
ii. Repeat the fitting procedure but use a gradient following method. Compare your results with your answer to i) and time the code.

iii. Repeat the code one more time but using simulated annealing. Again compare your results and execution times.

iv. Parallelize either of your part ii) or part iii) codes to see how quickly you can evaluate the result.

Problem 2 – Numerical forward models.

Consider the following system model of a near-field antenna measurement:



Here we have an antenna aperture split into two parts. The distance between the outermost aperture points is d, and the length of the 'hole' in the middle of the aperture is d'. The measurement plane is 40 meters from the aperture. The measurement wavelength is 1 cm. Suppose that the distance d is known to be between 1 and 2 meters, and the hole size d' is between 0 and 0.5 m.

Download the data file lab9prob2.dat. This file gives the distance y of each measurement, and the power (magnitude squared) of the antenna signal. The data are noisy. Plot the data to visualize it. Then solve for the best-fitting values d and d' to match the data.

Use both a complete search and either the gradient or simulated annealing methods, and compare the execution times.

Problem 3 – Image registration.

Download the files lab9prob3a.dat and lab9prob3b.dat from the web site. Each of these is a complex (real and imaginary) image of part of the San Jose area. The second image is stretched with respect to the first image. In this problem, we will measure the image distortion and resample the second image to the coordinate system of the first image.

i) Create a byte representation of the magnitude of each image, and combine these using disrgbfile to form a color image where image 1 is the red channel and image 2 is the green and blue channels. If you display this color image, you should be able to see that one image is a shifted and stretched version of the other.

ii) Estimate the shift by cross-correlating a small region (16 x 16 pixels) from the first image against a larger region (48 x 48 pixels) in the second image. We choose the second image to be larger because we do know only approximately how much one image is shifted with respect to the other. The location of the peak of the crosscorrelation give the image "offset" in pixels.

iii) Interpolate the position of the correlation peak to the nearest 1/8 of a pixel.

iv) Repeat steps ii) and iii) at many locations to characterize how the shift varies with location in the image.

v) Fit a line to the peak locations (offsets) to derive a transformation from the coordinates of the second image to the first image.

vi) Resample the second image so that each pixel lies on top of the corresponding pixel in the first image.

vi) Finally, create the same type of color image of the now-registered images as you did in step i) to ensure that you have properly co-registered the images.

Problem 4 – Stochastic gradient descent.

In this problem, we will implement gradient descent and stochastic gradient descent method to find the minimum of a simple objective function.

Let's suppose we want to fit a straight line $y = kx + b$ to a training set of two dimensional data points $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ using least squares. The objective function we want to minimize can be written as:

$$f(k, b) = \sum_{i=1}^{n} (y_i - kx_i - b)^2$$

(1) **[4 points]** In order to minimize the objective function, we need to first obtain the gradient with respect to k and b. Please derive $\nabla_k f(k, b)$ and $\nabla_b f(k, b)$.

(2) **[10 points for codes]** for this part, download data "lab9prob4dat.txt" from course website. Each line in the text file contains a single data point pair (x_i, y_i).

Implement and compare the following techniques on this data set:

(a) Gradient descent

Iterate through the entire dataset and update the parameters as follows:

```
j = 1
while convergence criteria not reached do
     Update k_next = k − η∇_k f(k,b)
     Update b_next = b − η∇_b f(k,b)
     k = k_next
     b = b_next
     j += 1
end while
```

where,
η is the step size of the gradient descent.
The convergence criteria is

$$\Delta_{\%\text{cost}} = \frac{\left|f^{(j)}(k,b) - f^{(j-1)}(k,b)\right| \times 100}{f^{(j-1)}(k,b)} < \varepsilon$$

Where $f^{(j)}(k, b)$ is the value of the objective function at j^{th} iteration.
$\Delta_{\%\text{cost}}$ is computed at the end of each while loop.

Initialize k=0 b=0 and compute $f^{(0)}(k,b)$ with these values.

For this method, use $\eta = 10^{-8}$ and $\varepsilon = 0.1$.

(b) Stochastic gradient descent

Stochastic gradient descent updates the parameters by only considering one training sample at a time. At each iteration, a single sample (x_i, y_i) is randomly selected from the training set and used to update k and b, as follows:

```
Randomly shuffle the training data
i = 1; j = 1;
while convergence criteria not reached do
     Update k_next = k − η∇_k f_i(k,b)
     Update b_next = b − η∇_b f_i(k,b)
     k = k_next
     b = b_next
     i = (i mod n) + 1
     j += 1
end while
```

where,
$f_i(k,b) = (y_i - kx_i - b)^2$
n is the number of data samples.
η is the step size.
The convergence criteria is

$$\Delta_{cost} = .5 * \Delta_{cost} + .5 * \Delta_{\%cost} < \varepsilon$$

$\Delta_{\%cost}$ is defined in part (a). Δ_{cost} and $\Delta_{\%cost}$ are computed at the end of each while loop.

Initialize Δ_{cost}=0, k=0, b=0 and compute $f_0(k,b)$ with these values.

For this method, use $\eta = 6*10^{-6}$ and $\varepsilon = 0.01$.

(3) **[6 points]** for both methods, plot the value of the objective function vs. the number of iterations. Time the entire while loop. Comment on when stochastic gradient descent may have a better performance.