CSE 4345 / CSE 5315 - *Computational Methods*

Homework 4- Fall 2011

Due Date: Nov. 27 2011

Problems marked with * are required only for students of CSE 5315 but will be graded for extra credit for students of CSE 4345.

Unconstrained Optimization

1) Optimizing image compression settings for storage Lossy image compression (like JPEG) trades off image quality against compression ratio (the ratio of the original, raw image size versus the compressed image size), effectively producing smaller images by increasing the information loss. Studies of different lossy image compression formats has shown that the behavior in terms of some loss measurements (e.g. Peak Signal to Noise Ratio) behaves (at least for parts of the compression) like an inverse exponential, going from infinite values when there is no noise in the image to zero when no information is left. Consider the following (fictitious) image quality function ($q(c)$) as a function of the compression ratio (c):

$$q(c) = \frac{75}{c^{\frac{1}{4}}}$$

Note that compression ratios smaller than one occur in many lossy compression schemes (including JPEG) and that a compression ratio of 1 does not result in the original image but is still lossy.

Consider an application that wants to store images in this format and has thus to determine what compression ratio to use. To do this, consider the cost of storage to be linear in the size of the resulting file and a penalty for quality loss that is linear with the inverse of the quality (i.e. it is 0 for a lossless image and increases as the quality decreases). The total quality of the stored data will trade off the storage cost and the quality loss by applying a weight of $1.85 * 10^8$ to the penalty function, resulting in the following overall cost function for the stored data:

$$f(c) = s/c + 1.85 * 10^8 * \frac{1}{q(c)}$$

where s is the size of the image.

The goal of the storage program is to determine the optimal compression ratio setting for the algorithm given the raw image size. (Note that this is strictly a constrained optimization problem since there are no negative compression rations. However, since the file size goes to infinity as the compression ratio approaches 0, and the penalty goes towards infinity as the compression ratio goes to infinity, there has to be an optimum in the range of positive values and thus the problem can be treated as an unconstrained optimization problem for iterative optimization approaches when applied only to positive values for the compression ratio).

a) Golden Section Search

   Implement Golden section Search to solve this one-dimensional optimization problem and determine the optimal compression ratios for the following raw image sizes: $614,400$; $3,712,000$; $18,000,000$.

b) Newton's Method

   Implement Newton's Method for this problem and repeat the experiments with the same values as in part a).

2) Multi-dimensional Optimization Consider an extension to the problem of problem $1)$ where all images have to be saved at the same resolution and we can determine this resolution. To make the resolution (and thus of the original size relevant we include an additional penalty function which expresses the loss in image content due to lower image resolution. We will assume that this penalty behaves like $p(s) = e * \frac{s_0 - s}{s} - 1$, where s_0 is the maximum image size (resolution) of the camera. This (with a weight on the new penalty of 1) leads to an extended cost function:

$$f_2(c, s) = s/c + 1.85 * 10^8 * \frac{1}{q(c)} + p(s)$$

a) Multivariate Newton Method

   Derive the formulation for Newton's Method (including the Gradient and the Hessian) for this problem considering that the maximum image size s_0 is $18,000,000$. Implement the Multivariate Newton method for this problem. You can use an existing implementation of a Linear Equation solver for the Newton step calculation.

b)* Conjugate Gradient

   Implement the Conjugate Gradient method for this problem.

Constrained Optimization

3) Optimizing Microcontroller power management using constrained optimization with Barrier functions

In class we already used the power dissipation of a CPU as an example. If we assume that we have a Microcontroller that also has a number of components that act resistively, we can extend this formulation to obtain the following equation describing the power consumption of the Microcontroller circuit:

$$P = C * f * V^2 + \frac{V^2}{R}$$

Assuming that in order to save energy, the system is equipped with a power management component, the goal of smart power management software is to determine the best voltage setting and clock rate such that it minimizes the amount of energy that is consumed but still allows the current processing requirements to be fulfilled and ensures that the supply voltage is sufficient to yield reliable calculations for the current voltage.

Assume the following properties for the Microcontroller:

- The capacitance of the Microcontroller is $C = 10^{-10}$,
- The resistance of the resistive components is $R = 50\Omega$,
- The maximum clock rate for the Microcontroller is $f_{max} = 1.2 * 10^9 Hz$,
- The maximum supply voltage for the Microcontroller is $V_{max} = 5V$,
- The minimum supply voltage for the Microcontroller is $V_{min} = 1.8V$,
- To ensure reliable calculations, the maximum clock rate is proportional to the supply voltage, $f_{max}(V) = 2.4 * 10^8 * V$).

Also, assume that the current processing requirements (i.e. the incoming jobs) can be fulfilled with an average clock rate of $f_r = 10^8$ clock cycles per second and that the power management system has the ability to turn off the Microcontroller for some period of time (i.e. it can set a duty cycle, $0 \leq t_{on} \leq 1$, where the Microcontroller is on and thus the power dissipation is $P = C * f * V^2 + \frac{V^2}{R}$ and for the remainder, $t_{off} = 1 - t_{on}$ the Microcontroller is off therefore does not dissipate any power, leading to an overall energy consumption of $t_{on} * P$ and a Microcontroller cycle requirement of $f * t_{on} \geq f_r$.

Define the power management as a constrained optimization problem where the objective function is the minimization of the power dissipation while ensuring that the processing can be performed reliably.

Implement a barrier function method to solve the constrained optimization problem for the values provided above.