Computational Optimization
Laboratory 1 - 1/18/08 - Golden Section Search
Due in class Friday 1/25/08

In this lab we experiment with a Matlab implementation of the golden section search.

1. Open your home directory of your RCS account as your H: drive. Create a new folder called compopt in this directory.

2. Download the files f.m and golden.m found under the Labs section of the course web page

   ```
   www.rpi.edu/~bennek/class/compopt
   ```

   into your compopt directory.

3. Start up Matlab. Change directory to your directory by typing

   ```
   cd H:\compopt
   ```

   Check to see that the files f.m and golden.m by typing

   ```
   dir
   ```

4. Edit the file f.m using the Matlab editor by typing

   ```
   edit f.m
   ```

5. Matlab is an interpreted language that has a syntax similar to C or Pascal. The best way to learn it is to just use it. It is smart in that it knows all about vectors and matrices. Functions are kept in separate files called M-files. The name of the function with a ".m" extension is the name of the M-file. The file f.m is an example of how you program a simple function, in this case $f(x) = x^2$. When you edit the file you should see the following contents:

   ```
   function fx=f(x)
   fx=x^2;
   ```

   The first line says here is a function of x which returns a single value fx. The second lines says the value returned is x^2. To evaluate f at the point 4, just type **f(4)** at the Matlab prompt in the command window.

6. The fplot command can be used to plot your function. Type help fplot to see details about fplot. Let's plot f between -2 and 2.

   ```
   fplot(@f,[-2,2]);
   ```

7. Now construct your own function. Save the file f.m under a new name, say g.m. Edit the file g.m to evaluate the function $x^8 - 4 * x + 3$. Try evaluating g.m for $x = 0$ and $x = 0.00001$. Compare the answers produced by Matlab with the analytical solutions. To see more digits of accuracy type

   ```
   format long
   ```

   Evaluate g(0) and g(.00001) again. Compare your answers.

8. Now let's look at how golden section search is implemented in Matlab. Edit the file golden.m. The percent indicates a comment that is ignored by Matlab. The comment lines before the function is defined are used to tell what the function is and how to call it. Type **help golden** in the Matlab window. Note that Matlab displays the comment lines before (and only before) the function is defined. It's a good idea to make your code self-documenting and always fill in these lines. If you have any questions about how any Matlab function works just type help and the function name. Try typing **help function**.

   After the comments, the definition of the function starts with the word function. You send into golden the name of the objective function, the left end point of the interval, the right end point of the interval, and a stopping tolerance. It returns the approximate minimum x value xmin, minimum objective value fxmin, and the end points of the last interval. The line 'function [xmin,fxmin,newa,newb]=golden(fun,a,b,tol)' actually defines the function. If you want more details on how Matlab works, checkout the Matlab primer available on my homepage under Useful Info.

9. To run golden section search on the function f on the interval $[-1, 0.1]$ with tolerance 0.01 enter the following at the Matlab prompt:

   ```
   [xmin,fxmin,newa,newb]=golden(@f,-1,0.1,0.01)
   ```

   The @ sign is used to indicate that the handle for the function f should be passed to golden. A function handle allows a function to be called indirectly. Within golden the variable func will now refer to the function handle of f. The command func(x) will then evaluation the function f at x.

   The results will be printed out for you. Note that we should evaluate the quality of the results in several ways. We can look at the at the difference between the optimal value for x and the calculated optimal value of x. We can look at the true optimal value of $f(x)$ and the actual optimal value. We can examine the first order necessary optimality conditions. Is $f'(xmin) = 0$? We can also examine the computational cost of the algorithm. Golden.m reports the number of function evaluations performed.

   You can plot your results by doing the following:

   ```
   fplot(@f,[-1,0.1]]);
   hold
   plot(xmin,f(xmin),'r^');
   ```

   The fplot command plots the f on the indicated interval. The hold command, holds the plot so we can add more stuff. The plot command, plots the point (xmin, f(xmin), in red with a triangle. See help plot for more information about the plot command and how to change it.

10. Now you try it. Minimize x^2 on the interval $[-1, 1]$ with tolerance 1e-6. You may find it handy to use the arrow keys to scroll back to the last time the golden function was used and just edit the input parameters. How many iterations did it take? How accurate was the solution in terms of the minimum value, minimum function value, and gradient?

11. Now construct a record of your results to hand in using the diary command. In Matlab, type **diary 'lab1.res'**. Now repeat the command you gave Matlab in the last item to minimize x^2 on the interval $[-1, 1]$ with tolerance 1e-6. When you are done type **diary off**. The Matlab output will be in the file lab1.res. You can print the file using the Matlab editor. In general, we don't want to print huge diary files, so feel free to edit the output to just the salient parts.

12. *The following two problems are the only part of the lab that you need to turn in.* Use golden section search to minimize the function $x^8 - 4 * x + 3$ over the interval 0 to 3 with tolerance 1e-4. What is the true minimum of this function? What is the derivative of the function at the point returned by golden.m? How close is the objective value of the point found to the optimal objective? Explain in words the quality of your results e.g. how accurate of a solution did you obtain?.

13. Now try the following: minimize $h(x) = 0.5 * x^2 - 10 * sin(x)$. Check the results for the intervals $[-10, 10]$ and $[-10, 5]$ with tolerance 1e-4. Plot the function and the solutions that were found in matlab. How do the results compare? How do you account theoretically for the discrepancies? Are the solutions found optimal in some sense?