MATH 426 - Assignment 6 Part I

June 17, 2008

1 Anonymous Functions and Function Handles

First write a function `simp(f, a, b, n)` which uses Simpson's rule to approximate the definite integral of $f(x)$ over the interval $[a, b]$ using n subintervals (n even). The implementation of the function will be very similar to that of `trap.m` we saw in the lecture. Next, write a script file in which which you will define the functions
$f(x) = \sin(\sqrt{x}) \quad x \in [0, \frac{\pi}{2}]$
$g(x) = e^{\cos(x)} \quad x \in [0, \pi]$
as anonymous functions and then use `simp` to compute the definite integral of each function over the given interval. Use $n = 4, 8, 16, 32, ..., 512$ and print a table as follows:

```
  n   I(f)         I(g)
------------------------------
  2   1.05997075   3.71030536
  4   1.09734946   3.97173128
  8   1.11043794   3.97746305
 16   1.11503637   3.97746326
 32   1.11665656   3.97746326
 .
 .
 .
```

2 Subfunctions - Functions Handles

Write a function `deriv(f, x0, h)` that uses the following finite difference formula

$$Df(x_0) = \frac{f(x_0 + h) - f(x_0 - h)}{2h}$$

to approximate the derivative $f'(x_0)$ of $f(x)$ at $x = x_0$. Then, write another function `testDeriv` with header

```
function testDeriv
```

which computes the derivative of $f(x) = \cos(\frac{x}{2}) + \sin(2x)$ over the interval $[-\pi, \pi]$, and plots the function $f(x)$ and its derivative in the same plot (use a suitably small value of h).

You will need to define the function $f(x)$ as a subfunction of `testDeriv`.

3 Another Experiment With Function Handles

Write a function

```
x=bisection(f, a, b, tol)
```

which finds a zero of $f(x)$ in the interval $[a, b]$, using the bisection algorithm discussed in class.

Then, find the zero of the following functions near the given points:
$f(x) = \cos(\frac{x}{2}) + \sin(2x)$ near $x = 2$
$g(x) = x - 5\ln(x)$ near $x = 1$

In writing the driver routine that calls `bisection` you are free to define the functions $f(x)$ and $g(x)$ as anonymous functions or subfunctions (in which case the driver routine will itself be a function).