MATH 4073 Homework 6 Due Wed, 10/13/10

Problem 1. The MATLAB code `muller.m` (available on the class web-site) is a simple implementation of Müller's method for finding zeros described in Section 2.6 of the book. Don't forge to type `format long` in MATLAB, to see the results with high precision.

(a) Run `muller.m` in verbose mode to find a (complex) root of the equation

$$x^5 - 3x^4 + 4x^3 - 4x^2 + 3x - 1 = 0 \ ,$$

starting from the following initial points: $p_0 = 0.0$, $p_1 = -1.0$, $p_2 = -1.5$; use tolerance 10^{-10} and maximum number of steps 100. Do the iterates converge, and to what value? How many steps were needed to achieve the desired accuracy?

(b)
- Run `muller.m` in verbose mode to find a (complex) root of the equation

$$x^5 - 3x^4 + 4x^3 - 4x^2 + 3x - 1 = 0 \ ,$$

starting with $p_0 = 0.7$, $p_1 = 0.8$, $p_2 = 1.1$; use same tolerance and maximum number of steps as in part (a). Do the iterates converge, and to what value? How many steps did the algorithm perform until it stopped? At the moment of stopping, does it look like the desired accuracy was really achieved?

- Run `muller.m` in verbose mode to find a (complex) root of the equation

$$x^5 - 3x^4 + 4x^3 - 4x^2 + 3x - 1.1 = 0 \ ,$$

(*note that this equation is different from the one considered before!*). Start with $p_0 = 0.7$, $p_1 = 0.8$, $p_2 = 1.1$; with the same tolerance and maximum number of steps as before. Do the iterates converge, and to what value? How many steps did the algorithm perform until it stopped? At the moment of stopping, does it look like the desired accuracy was really achieved?

- Run `muller.m` in verbose mode to find a (complex) root of the equation

$$x^5 - 3x^4 + 4x^3 - 4x^2 + 3x - 0.9 = 0 \ ,$$

(*note that this equation is different from the ones considered before!*). Start with $p_0 = 0.7$, $p_1 = 0.8$, $p_2 = 1.1$; with the same tolerance and maximum number of steps as before. Do the iterates converge, and to what value? How many steps did the algorithm perform until it stopped? At the moment of stopping, does it look like the desired accuracy was really achieved?

- What was the reason for Müller's method to behave so differently for the three equations considered in part (b)? Give mathematical arguments in support of your claim. You can confirm your guess by finding the roots using the MATLAB command `roots`. (Incidentally, even `roots` will have trouble with the root that

`muller.m` had trouble with, but to see this, you have to use `format long`.)

Hint: Looking at the graphs of the left-hand sides of the equations can also be very helpful. In Mathematica, for example, the command

```
Plot[x - Cos[x] + 2, {x, -4, 4}]
```

will plot the graph of the function $f(x) = x - \cos x + 2$, for $x \in [-4, 4]$. (Note that Mathematica does not necessarily put the horizontal axis at $y = 0$.)

Problem 2. As you know, one way to approximate a function f of one variable is to replace it by its tangent line at some point of interest, or by the "best fitting" parabola at this point (these approximations correspond to using the first- or second-order Taylor polynomial of the function f at this point). This type of approximation, however, works very well only near this point, and can be very inaccurate over an entire *interval*.

One way to approximate a function f (of one variable) on an entire interval is the following. Choose some class of functions $\mathcal{H}$, say all linear functions. Then look for a function h from this class $\mathcal{H}$ for which the "distance" between f and h is the smallest possible. The "distance" – which is usually called "error" – can be defined in many different ways. If we want to approximate f by a function $h \in \mathcal{H}$ on the interval $[a, b]$, and we want $|f(x) - h(x)|$ to be small for all $x \in [a, b]$, then an appropriate definition for the "error" would be

$$\max_{x \in [a,b]} |f(x) - h(x)| \ .$$

Another choice is to minimize

$$\int_a^b |f(x) - h(x)| \, dx \ .$$

In this problem you will apply this idea to find the best approximation of the function $f(x) = x^2$ by a linear function, $h_{\mu,\nu}(x) := \mu x + \nu$, over the interval $[0, 1]$ if the "error" is given by the integral

$$E_{a,b,f}(\mu, \nu) := \int_a^b [f(x) - h_{\mu,\nu}(x)]^2 \, dx \ . \tag{1}$$

In other words, you have to choose the values of the constants μ and ν that minimize the error $E_{a,b,f}(\mu, \nu)$ given by (1).

Hint: You may find the following formula useful:

$$(\alpha + \beta + \gamma)^2 = \alpha^2 + \beta^2 + \gamma^2 + 2\alpha\beta + 2\alpha\gamma + 2\beta\gamma \ .$$

Problem 3. In this problem you will study in detail the error in the linear interpolation of the function $f(x) = \sin(\pi x)$ on the interval $x \in [x_0, x_1] := [0, \frac{1}{4}]$.

(a) Find the first order Lagrange polynomial $P_1(x)$ of $f(x) = \sin(\pi x)$ that passes through the points $(x_0, f(x_0))$ and $(x_1, f(x_1))$. Find the coefficients of $P_1(x)$ exactly (i.e., leave numbers like $\sqrt{2}$ in this form).

(b) Let

$$E := \max_{x \in [0, \frac{1}{4}]} |f(x) - P_1(x)|$$

be the *true* error of the first order Lagrange interpolation. Show that the exact value of E is $\frac{1}{\pi}\sqrt{\pi^2 - 8} - \frac{2\sqrt{2}}{\pi} \arccos \frac{2\sqrt{2}}{\pi}$. Find the numerical value of E.

Hint: You first have to find the value x^* of the argument that maximizes the expression $|f(x) - P_1(x)|$, and then show that $|f(x^*) - P_1(x^*)|$ is equal to the expression above.

(c) Find the error bound given by Theorem 3.3 in Section 3.1 of the book. Note that you do not know the value of $\xi(x)$ in this bound, so you will have to take the maximum of the absolute value of the derivative in this bound over the whole interval $[x_0, x_1]$. Separately, you will have to find the maximum value of the product of $(x - x_j)$ terms. Find the exact value of this bound, and then compute its numerical value. Compare with the exact value of the error found in part (b).

(d) Now compute the first-degree Taylor polynomial of $\sin(\pi x)$ around $x = 0$.

(e) Let $G := \max_{x \in [0, \frac{1}{4}]} |f(x) - T_1(x)|$ be the error of the approximation of $f(x) = \sin(\pi x)$ by its first-degree Taylor polynomial around 0. Find the exact value of G. How does it compare with the true E error in first-order Lagrange interpolation?

Hint: This is very easy and does not require any differentiation – think about the shape of the function $\sin(\pi x)$ and the geometric meaning of $T_1(x)$.