MATH 4073 **Homework 3** **Due Wed, 9/11/13**

Problem 1. Let the sequence $(x_n)_{n=0}^{\infty}$ be defined for all $n \geq 0$ by $x_{n+1} = \frac{7x_n}{8} + \frac{1}{x_n^2}$.

(a) Think of the sequence $(x_n)_{n=0}^{\infty}$ as a functional iteration, $x_{n+1} = g(x_n)$, for an appropriate function g. Write g explicitly. Use one of the theorems stated in Lecture 3 to show that g has a fixed point in the interval $[\frac{3}{2}, 100]$. (In fact, instead of 100, I could have put any arbitrarily large positive number.)

(b) Looking at the explicit expression for $g'(x)$ and thinking about the slope of the graph of $g(x)$ (and comparing it with the slope of the "diagonal") for $x > 0$, give a simple argument showing that there cannot be more than one positive fixed point of g.

(c) Solve the equation $x = g(x)$ explicitly to find the positive fixed point of g. Clearly, this fixed point will be equal to the limit of the sequence $(x_n)_{n=0}^{\infty}$.

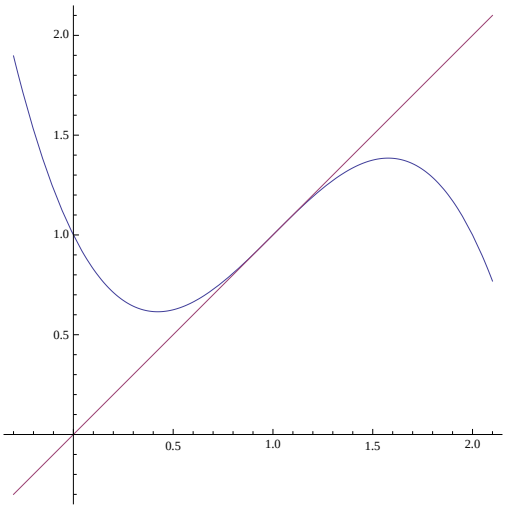
(d) What does the general theory predict about the rate of convergence α and the asymptotic error constant λ?

(e) Now run the MATLAB code `fixedpoint.m` for the functional iteration $x_{n+1} = g(x_n)$ (where g is the function you wrote in part (a)). Start from $x_0 = 3$, and take some small enough tolerance, say, 10^{-12}. From the output, measure empirically the values of α and λ. Do the empirical values match the predictions of the general theory from part (d)?

Problem 2. Consider the function $g(x) = -x^3 + 3x^2 - 2x + 1$. The Mathematica command

```
Plot[ {x, -x^3+3*x^2-2*x+1}, {x,-0.3, 2.1} ]
```

displays the graph of g and the diagonal ($y = x$); the figure is shown below.



(a) Show (by paper and pencil) that $x = 1$ is a fixed point of the function g.

(b) One can easily calculate that $g(0.5) = 0.625$ and $g(1.5) = 1.375$, and can show that g maps the interval $[0.5, 1.5]$ in the interval $[0.625, 1.375]$, which is entirely contained in the interval $[0.5, 1.5]$. Symbolically, one can write

$$g\big([0.5, 1.5]\big) = [0.625, 1.375] \subseteq [0.5, 1.5] \ .$$

Therefore, according to part (a) of the Theorem from Lecture 3 (on page 3.4 of the lecture notes), there is a fixed point of g in the interval $[0.5, 1.5]$.

Explain briefly why you can *not* apply part (b) of the same Theorem to show that the fixed point $x = 1$ is the only fixed point in the interval $[0.5, 1.5]$?

(c) Since you could not use part (b) of the Theorem from Lecture 3 to show the uniqueness of the fixed point $x = 1$ of the function g, try something else: define the function $f(x) := g(x) - x$, show that f is strictly decreasing everywhere except at the point $x = 1$, and use this fact to prove that f cannot have more than one zero.

Hint: Show that the first derivative of f can be written as $-3(x-1)^2$.

(d) Use the MATLAB program `fixedpoint.m` to find the fixed point of g with tolerances 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}, and 10^{-7}. To see the number of iterations the code will perform, set the variable `verbose` to be equal to 1. The stopping criterion this program uses is $|p_n - p_{n-1}| <$ `tol`. Display your results in a table:

Desired tolerance	Value obtained	Number of iterations
10^{-3}	0.901621	9
$\vdots$	$\vdots$	$\vdots$

Look at the computed values of the fixed point. Do they look correct within the desired tolerance?

Hint: To see the output value with more digits, type the Matlab command `format long`. Also, make sure that the parameter `nmax` that you pass to the program (the maximum number of iterations allowed) is large enough.

(e) In your opinion, why did the program need such a large number of iterations before the stopping criterion was met and the program stopped? (And recall that the precision obtained was far from the desired tolerance).

Hint: Think about the values of g'.

Problem 3. Recall that the *multiplicity* of a zero p of the function f is defined as the number m such that

$$f(x) = (x-p)^m \, q(x) \ ,$$

where q is a function satisfying $\lim_{x \to p} q(x) \neq 0$.

Recall also that Newton's method for finding a zero of the function f (or, equivalently, a root of the equation $f(x) = 0$) is based on the iterative procedure $x_{n+1} = g(x_n)$, where x_0 is some starting value, and $g(x) = x - \dfrac{f(x)}{f'(x)}$. We stated in class that, if p is a simple zero of f (i.e., a zero of

multiplicity 1) and the Newton's method converges to p, then the convergence is at least quadratic, i.e., or order $\alpha \geq 2$.

If, however, the zero of f is non-simple, then the Newton's method converges only linearly.

In class we proved that, if p is a fixed point of the function g and $g'(p) \neq 0$, then if the iteration $x_{n+1} = g(x_n)$ converges to p, then the convergence is linear (and $\lambda = |g'(p)|$).

In this problem you will show that, indeed, the Newton's method converges linearly for $m \geq 2$, and will find a modification of Newton's method that works with multiple zeros (but one needs to know the multiplicity of the zero and pass it to the program as one of the arguments).

Let p be a zero of multiplicity $m \geq 2$ of f. Then the Newton's iteration for finding a zero of f has the form

$$\begin{aligned} g(x) &= x - \frac{f(x)}{f'(x)} = x - \frac{(x-p)^m q(x)}{[(x-p)^m q(x)]'} \\ &= x - \frac{(x-p)^m q(x)}{m(x-p)^{m-1} q(x) + (x-p)^m q'(x)} \\ &= x - (x-p)\frac{q(x)}{mq(x) + (x-p)q'(x)} \ , \end{aligned}$$

therefore

$$g'(x) = 1 - \frac{q(x)}{mq(x) + (x-p)q'(x)} - (x-p)\,\frac{\mathrm{d}}{\mathrm{d}x}\left(\frac{q(x)}{mq(x) + (x-p)q'(x)}\right) \ .$$

This implies that

$$g'(p) = 1 - \frac{1}{m} \neq 0 \ ,$$

hence the convergence of Newton's method is only linear.

(a) Let p be a zero of multiplicity $m \geq 2$ of f. Consider the following modification of the Newton's method: $x_{n+1} = g(x_n)$, where

$$g(x) = x - m\frac{f(x)}{f'(x)} \ .$$

Show that in this case $g'(p) = 0$, hence the convergence is faster than linear.

(b) Show that the multiplicity of the root $\frac{\pi}{2}$ of the equation $(x - \frac{\pi}{2})(1 - \sin x) = 0$ is $m = 3$.
Hint: Expand $\sin x$ in a Taylor series around $x_0 = \frac{\pi}{2}$.

(c) The Mathematica code

```
p = N[3, 50000];
m = 3;
f[x_] := (x - Pi/2) * (1-Sin[x]);
For[i = 1, i <= 10, i++,
  { p = p - m*f[p]/f'[p],
    error = Abs[p - Pi/2],
    Print[i, "   ", N[Log[error],10]]
  }
]
```

can be used to find empirically the order of convergence of the method. Here is the output:

```
1     -0.7204139049
2     -3.415376010
3     -11.50140053
4     -35.75947410
5     -108.5336948
6     -326.8563569
7     -981.8243432
8     -2946.728302
9     -8841.440179
10     -26525.57581
```

What is the order of the convergence that you observe? Explain briefly your reasoning. (Note that we are doing the calculations with accuracy or 50000 decimal digits!)

(d) The number $\frac{\pi}{2}$ is a root of the equation

$$\left(x - \frac{\pi}{2}\right)^3 (1 - \sin x) = 0$$

of multiplicity 5. The Mathematica code from part (c) can be modified appropriately find empirically the order of convergence in this case. The output is given below. What do you observe?

```
1     -0.9648058725
2     -4.371283436
3     -14.59097156
4     -45.25003594
5     -137.2272291
6     -413.1588085
7     -1240.953547
8     -3724.337761
9     -11174.49041
10     Indeterminate
```

(e) Modify the MATLAB code `newton.m` to write a code `newton_m.m` that implements the algorithm from part (a) for finding zeros of multiplicity m of the equation $f(x) = 0$. The first line of your code must be

```
function r = newton_m( fun, funder, m, xinit, tol, nmax, verbose)
```

Note that in MATLAB the name of a file must be the same as the name of the code in it. Please attach a printout of your code.

(f) Run your code `newton.m` to find a zero of the equation $(x - \frac{\pi}{2})^3 (1 - \sin x) = 0$ from part (d). Start from some initial value close enough to the exact root $\frac{\pi}{2}$. Attach a printout.

(g) One can estimate roughly the order of convergence m of an iterative method as follows. If after iterating enough number of times the error is significantly smaller than 1 (say, 0.01 or smaller), then the number of correct digits of x_{n+1} is roughly m times the number of correct digits of x_n. You can observe this, in the following quadratically convergent sequence (coming from using Newton's method to find $\sqrt[3]{12}$ as a root of the equation $x^3 - 12 = 0$).

```
1.000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
4.666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666
3.294784580498866213151927437641723356009070294784580498866213151927437641723356009070294784580498866
2.564996283933496822279640898870787662335200127188373868695090191534036878441096931206491314332028214
2.317973627725310965894671183077646390006759422097600376823322683758217378024541415220574391246824662
2.289778566911922285264949690307277524808339353864854224824331811491750945243930217290781459861111391
2.289428538627570193775919893742118572719030429524354378801546408311322845023342839603272971787615841
2.289428485106664986796174761541083407108687502501926456729710419471508178918399015462717660858495387
2.289428485106663735616084423880037791808000926685328369417780029499568633251106146440076011857659331
2.289428485106663735616084423879354017831813841575862144198104552351246075611950964542191974634843603
2.289428485106663735616084423879354017831813841575862144198104348131348598048428300875216322061834091
2.289428485106663735616084423879354017831813841575862144198104348131348598048428300875216322061834091
```

Give a theoretical explanation of this rule of thumb.

Hint: For example, in the number $2.289428485106664986796\ldots$ in the sequence above, you know the result with 8 correct digits after the decimal point: 2.28942848510666; the next one, $2.2894284851066637356160844238800377 9\ldots$, has 28 correct digits after the decimal point, namely, 2.28942848510666373561608442388. If you know a number x_n with k digits of accuracy, what can you say about $\log_{10}|x_n - p|$?