AMSC/CMCS 466 – Introduction to Numerical Analysis I

Fall Term 2003

Instructor: Georg Dolzmann

Homework set #4

Problem 1: a) Write a simple MATLAB program

```
system(f,g,fx,fy,gx,gy,x0,y0,tol,miter)
```

to solve the system of two nonlinear equations $f(x,y) = g(x,y) = 0$ by Newton's method. The input parameters are the functions f and g and their partial derivatives, the starting point (x_0, y_0), an absolute error tolerance `tol`, and the maximum number of iterations `miter`. For the solution of the resulting linear system use the MATLAB command \.

b) The following system has four zeros in the domain $(-4,4) \times (-4,4)$,

$$f(x,y) = x^2 + xy^3 - 9 = 0, \quad g(x,y) = 3x^2y - y^3 - 4 = 0.$$

Use the command `contour` to plot the level sets of both f and g in the same picture to determine reasonable initial guesses. Try the following MATLAB code:

```
[X,Y] = meshgrid(-4:.1:4, -4:.1:4);
figure
hold on
contour(X,Y,X.^2+X.*Y.^3-9, [0 0], 'r');
contour(X,Y,3*X.^2.*Y-Y.^3-4, [0 0], 'b');
```

c) Use `system` to approximate the four zeros with `tol`=10^{-10}. Determine the number of iterations and the rate of convergence.

d) Define the function $h(x,y) = f^2(x,y) + g^2(x,y)$. The zeros of the nonlinear system correspond to the minima of h. Use the MATLAB function `fmins` to approximate the four zeros. Use `help fmins` and `help foptions` to find out more about `fmins` and its options. In particular, use `options(1)=1` and `options(3)=`$1.e-10$.

Problem 2: [Atkinson 2.33] Use Newton's method to calculate the real roots of the following polynomials as accurately as possible. Estimate the multiplicity of each root, and then if necessary, try an alternative way of improving your calculated values.

a) $x^4 - 3.2x^3 + .96x^2 + 4.608x - 3.456$;

b) $x^5 + .9x^4 - 1.62x^3 - 1.458x^2 + .6561x + .59049$.

Use `ezplot` to plot the functions in order to get estimates for the values of the roots. You can use the commands

```
syms x
diff(x^4-3.2*x^3+.96*x^2+4.608*x-3.456,'x')
```

to find the derivative of f. Based on the assumption that

$$\frac{e_{n+1}}{e_n^p} = C,$$

find a formula for the *experimental order of convergence* $p =$`eoc`. Your formula will involve three successive estimates for the absolute error. Modify your code for the Newton scheme to allow a correction for roots with higher multiplicity. A possible code could involve a definition of the form

```
function [root, ier] = nmr(f,df,q,x0,eps,itmax,info)
```

where `q` is an estimate for the multiplicity of the root and `info` controls how much information is printed during the iterations. Use

```
disp(sprintf('x%i=%f, error = %f, eoc = %f\n',itnum,x1,enp1,eoc));
```

to print in each step the value x_{n+1}, an estimate for the absolute error e_{n+1}, and the experimental order of convergence (when defined). Use `eps=`10^{-6}. Make sure that all output parameters are defined before you exit the subroutine. Otherwise MATLAB will produce an error.

Problem 3: [Atkinson 2.28] There is another modification of Newton's method, similar to the secant method, but using a different approximation to the derivative $f'(x_n)$. Define

$$x_{n+1} = x_n - \frac{f(x_n)}{D(x_n)}, \quad D(x_n) = \frac{f\big(x_n + f(x_n)\big) - f(x_n)}{f(x_n)} \quad n \geq 0.$$

This one-point method is called *Steffenson's method.* Assuming $f'(\alpha) \neq 0$, show that this is a second-order method. *Hint:* Write the iteration as $x_{n+1} = g(x_n)$. Use $f(x) = (x - \alpha)h(x)$ (why?) with $h(\alpha) \neq 0$, and then compute the formula for $f(x)$ in terms of $h(x)$. Having done so, apply Theorem 2.8.

Problem 4: Consider the following recursion,

$$x_{n+1} = 2x_n - 3 + \frac{2}{x_n}.$$

Determine the fixed points (by hand) and examine whether the iteration converges locally. If this is the case, determine the rate of convergence.