

MA 784, Assignment 1, Due February 26, 2009

Start work on this immediately. The second assignment will be given before this one is due. There are three problems. **Do at least one of them**, which will count toward the six you owe me for the term.

In some of the exercises here, and in the rest of this part of the course, you will be asked to plot or tabulate. When asked to do this, for each iteration tabulate the iteration counter, the norm of  $F$ , and the ratio of  $\|F(x_n)\|/\|F(x_{n-1})\|$  for  $n \geq 1$ . A better alternative is to use the `plot` or `semilogy` commands in MATLAB to plot the norms. When one does this, one can visually inspect the plot to determine superlinear convergence without explicitly computing the ratios.

When the norm is not specified, use the  $l^\infty$  norm. Discussions of results should explain why the numbers and/or plots do or do not match the theory, how you decide which of several alternatives is performing best, and how you implemented things that were not in the codes that I provided. I expect the results to be **typeset**.

### 1. Scalar Equations

Either write a program that solves single nonlinear equations with Newton's method, the chord method, and the secant method (with  $x_{-1} = .99x_0$ ) or use the scalar equation solvers from the course web page for this assignment. Apply these codes to the following function/initial iterate combinations, tabulate or plot iteration statistics, and explain your results.

For each problem estimate the q-order or q-factor (How might you do that?) and, if it's not what the usual theory predicts, explain why. At least two of these problems will produce results that are NOT what the usual theory predicts!

- (a)  $f(x) = \cos(x) - x$ ,  $x_0 = .5$ ,    (b)  $f(x) = \tan^{-1}(x)$ ,  $x_0 = 1$ ,  
(c)  $f(x) = \tan^{-1}(x)$ ,  $x_0 = 10$ ,    (d)  $f(x) = \sin(x)$ ,  $x_0 = 3$ ,  
(e)  $f(x) = \sin(x^2)$ ,  $x_0 = .5$ ,    (f)  $f(x) = x^2 + 1$ ,  $x_0 = 10$

### 2. H-equation Experiments

The solvers `nsold` and `nsoli` live on the web page (look at the menu for nonlinear equations codes) as do a function evaluation for the H-equation and a demo (`heqdemo.m`) that uses `nsold.m` to solve it. The demo requires at least MATLAB 6.0. The demo should make your programming task much easier.

- (a) Do problems 5.7.18 and 5.7.21 in the red book. Use  $H_0 = 0$  as the initial iterate. Try values of  $c = .9, .9999, 1$  and  $N = 100, 400$ . Discuss how the difference in performance between difference and analytic Jacobians varies with  $c$  and  $N$ . What's strange about  $c = 1$ ?
- (b) Use the Newton-GMRES (default) option of the code `nsoli` to solve the same H-equation problems as in the previous problem. How does the performance compare with the dense matrix and direct factorization approaches in the previous problem? How do your conclusions depend on  $c$  and  $N$ ?
- (c) Modify `nsoli.m` and `nsold.m` by setting the forward difference step  $h = 10^{-2}$  (this part of the code is not hard to find if you understand the algorithms). How does this change affect the convergence rate observations that you made in the first two parts of this problem? It's enough to look at  $N = 400$  and  $c = .9$ .

### 3. Proofs and Convergence Theory

- (a) Do exercises 5.7.1, 5.7.11, 5.7.12, and 5.7.15 in the book.  
(b) Do exercises 6.5.2, 6.5.4, and 6.5.7 in the book.