Math 701 Programming Project 1

Newton's Method

We will work on the programming part of this project in class; however, your final report should be prepared independently. Where appropriate include full program listings and output.

1. Consider Newton's method for solving $f(x) = 0$ where $f(x) = x^2 - 2$ using the starting point $x_0 = 1$.

(i) Let $e_n = x_n - \sqrt{2}$ and create a table with three columns showing n, x_n and e_n for $n = 0, 1, \ldots, 8$.

(ii) A sign of quadratic convergence is that the number of significant digits double at each iteration. Does that happen in this case?

(iii) Comment on how rounding error effects the numerical convergence of Newton's method.

(iv) Write $|e_{n+1}| = M_n |e_n|^2$ and compute M_n for $n = 1, 2, 3$, and 4. In this case is M_n bigger or less than 1?

(v) Use multi-precision arithmetic with at least 10 000 digits precision to determine the asymptotic value of M_n when n is large. Can you also find this value analytically?

2. Consider the secant method for solving $f(x) = 0$ where $f(x) = x^2 - 2$ using the starting points $x_0 = 0$ and $x_1 = 1$.

(i) Let $e_n = x_n - \sqrt{2}$ and create a table with three columns showing n, x_n and e_n for $n = 0, 1, \ldots, 8$.

(ii) A sign of quadratic convergence is that the number of significant digits double at each iteration. Does that happen in this case?

(iii) According to Wikipedia `https://en.wikipedia.org/wiki/Secant_method` the order of convergence of the secant method is

$$\alpha = \frac{1 + \sqrt{5}}{2} \approx 1.618,$$

which is less than quadratic. Write $|e_{n+1}| = M_n |e_n|^\alpha$ and compute M_n for $n = 1, 2, \ldots, 7$. In this case is M_n bigger or less than 1?

(iv) Prove that the order of convergence of the secant method is α. If you look the proof up, please cite your references and rewrite the proof in your own words.

3. Consider the fixed point iteration for solving $f(x) = 0$ given by $x_{n+1} = h(x_n)$ where

$$h(x) = x - \frac{f(x)f'(x)}{[f'(x)]^2 - f(x)f''(x)}.$$

(i) Show that $f(x) = 0$ implies $h(x) = x$. Conversely show that if $h(x) = x$ then either $f(x) = 0$ or $f'(x) = 0$.

(ii) Compute $h'(x)$ and show that

$$h'(x) = \begin{cases} 0 & \text{when } f(x) = 0 \text{ and } f'(x) \neq 0 \\ 2 & \text{when } f(x) \neq 0,\ f'(x) = 0 \text{ and } f''(x) \neq 0. \end{cases}$$

Conclude that the fixed points of h for which $f(x) = 0$ are stable, but the fixed points for which $f'(x) = 0$ are not.

(iii) Use this fixed point iteration with $x_0 = 1$ to solve $f(x) = 0$ where $f(x) = x^2 - 2$. Compare the performance of this method with your results for Newton's method.

(iv) Suppose $f(x) = (x - \xi)^m q(x)$ where $\lim_{x \to \xi} q(x) \neq 0$. Let $g(x) = x - f(x)/f'(x)$ as in Newton's method and show that

$$\lim_{x \to \xi} g'(x) = \frac{m-1}{m} \qquad \text{and} \qquad \lim_{x \to \xi} h'(x) = 0.$$

Conclude that even when $f'(x) = 0$ this method, unlike Newton's method, has an accelerating rate of convergence as x_n approaches the solution $x = \xi$ to $f(x) = 0$.

4. The function

$$f(x) = 2\cos(5x) + 2\cos(4x) + 6\cos(3x) + 4\cos(2x) + 10\cos(x) + 3$$

has two roots on the interval $[0, 3]$; one root is near 1 and the other near 2.

(i) Use Newton's method $x_{n+1} = g(x_n)$ with $x_0 = 1$ and also with $x_0 = 2$ to approximate these two roots. Use the fact that the exact roots are $\pi/3$ and $2\pi/3$ to compute the error e_n at each iteration for $n = 0, 1, \ldots, 18$.

(ii) Use the method $x_{n+1} = h(x_n)$ with $x_0 = 1$ and again also with $x_0 = 2$ to approximate these two roots. Again use the fact that the exact roots are $\pi/3$ and $2\pi/3$ to compute the error e_n at each iteration for $n = 0, 1, \ldots, 18$.

(iii) Comment on the rate of convergence and the effects of rounding error in the above two computations.

5. Define

$$x_{n+1} = \frac{x_n + 2x_{n-1} + x_{n-2}}{4} - \frac{F_n F_n'}{[F_n']^2 - F_n F_n''}$$

where

$$F_n = f\Big(\frac{x_n + 2x_{n-1} + x_{n-2}}{4}\Big),$$
$$F_n' = \frac{2}{x_n - x_{n-2}}\Big\{f\Big(\frac{x_n + x_{n-1}}{2}\Big) - f\Big(\frac{x_{n-1} + x_{n-2}}{2}\Big)\Big\},$$
$$F_n'' = \frac{2}{x_n - x_{n-2}}\Big(\frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}} - \frac{f(x_{n-1}) - f(x_{n-2})}{x_{n-1} - x_{n-2}}\Big)$$

to create a secant-method-like approximation of the method given by h that doesn't involve f' and f''. Study this method both numerically and analytically. Test this method for the functions $f(x) = x^2 - 2$ and

$$f(x) = 2\cos(5x) + 2\cos(4x) + 6\cos(3x) + 4\cos(2x) + 10\cos(x) + 3.$$

How does this method compare to the usual secant method?

6. Consider the simpification to the above method given by

$$x_{n+1} = x_n - \frac{F_n F_n'}{[F_n']^2 - F_n F_n''}.$$

Study this simplification both numerically and analytically. Does it work as well as the previous method? Why or why not? Consider adding a relaxation parameter $\alpha \in (0,1)$ to obtain

$$x_{n+1} = x_n - \frac{\alpha F_n F_n'}{[F_n']^2 - F_n F_n''}.$$

Can you find a value of α which makes the scheme work better? Is there an optimal value for α?