# 2021 Spring MAS 365: Homework 2

posted on Mar. 18; due by Mar. 25

1. [10 points] Let $\{p_n\}_{n=0}^{\infty}$ be a sequence generated by the Secant method. It can be shown that if $\{p_n\}_{n=0}^{\infty}$ converges to $p$, the solution to $f(x) = 0$, then a constant $C$ exists with $|e_{n+1}| \approx C|e_n|\,|e_{n-1}|$ for sufficently large value of $n$, where $e_n := p_n - p$ for all $n$. Assuming that $\{p_n\}$ converges to $p$ of order $\alpha$, show that $\alpha = \frac{1+\sqrt{5}}{2} \approx 1.62$.

   **Solution:** For sufficently large values of $n$, we have
   $$|e_{n+1}| \approx \lambda |e_n|^{\alpha}.$$
   Then, we have
   $$\lambda |e_n|^{\alpha} \approx |e_{n+1}| \approx C|e_n|\,|e_{n-1}| \approx C|e_n|\,\frac{1}{\lambda^{1/\alpha}}|e_n|^{1/\alpha}.$$
   Here, we need $\alpha = 1 + \frac{1}{\alpha}$ for the equation to satisfy for all $n$, and this reduces to $\alpha = \frac{1+\sqrt{5}}{2}$.

2. [5 points each]

   (a) Show that the sequence $p_n = 10^{-2^n}$ converges quadratically to 0.

   (b) Show that the sequence $p_n = 10^{-n^k}$ does not converge to 0 quadratically for any $k > 1$.

   **Solution:**

   (a)
   $$\lim_{n\to\infty} \frac{|p_{n+1} - 0|}{|p_n - 0|} = \lim_{n\to\infty} \frac{10^{-2^{n+1}}}{(10^{-2^n})^2} = 1.$$

   (b)
   $$\lim_{n\to\infty} \frac{|p_{n+1} - 0|}{|p_n - 0|}^2 = \lim_{n\to\infty} \frac{10^{-(n+1)^k}}{(10^{-n^k})^2} = \lim_{n\to\infty} 10^{2n^k - (n+1)^k} = \lim_{n\to\infty} 10^{n^k\left(2 - \left(\frac{n+1}{n}\right)^k\right)} = \infty$$

3. [5 points each] Let $P_n(x)$ be the $n$th Taylor polynomial for $f(x) = e^{-x}$ about $x_0 = 0$.

   (a) For fixed $x$, show that $p_n = P_n(x)$ satisfies
   $$\lim_{n\to\infty} \frac{p_{n+1} - p}{p_n - p} < 1,$$
   where $p$ is a limit of $\{p_n\}_{n=0}^{\infty}$.

   (b) Let $x = 1$, and use Aitken's $\Delta^2$ method to generate the sequence $\hat{p}_0, \ldots, \hat{p}_5$. Report five-digit rounding values. [Hint: Use MATLAB command `format long` for more digits.]

   **Solution:**

(a) By Taylor's Theorem and $f^{(k)}(x) = (-1)^k e^{-x}$, we have

$$p_n = \sum_{k=0}^{n} \frac{f^{(k)}(x_0)}{k!}(x - x_0)^k = \sum_{k=0}^{n} \frac{(-1)^k}{k!} x^k$$

that converges to a limit $p$ for any fixed $x$, and

$$p_n - p = -\frac{f^{(n+1)}(\xi(x))}{(n+1)!}(x - x_0)^{n+1} = -\frac{(-1)^{n+1}e^{-\xi_n(x)}}{(n+1)!} x^{n+1},$$

where $\xi_n(x)$ is between 0 and $x$. Then,

$$\lim_{n\to\infty} \frac{p_{n+1} - p}{p_n - p} = \lim_{n\to\infty} \frac{-\frac{(-1)^{n+2}e^{\xi_{n+1}(x)}}{(n+2)!}x^{n+2}}{-\frac{(-1)^{n+1}e^{\xi_n(x)}}{(n+1)!}x^{n+1}} = \lim_{n\to\infty} -\frac{e^{\xi_{n+1}(x)-\xi_n(x)}x}{n+2} = 0 < 1.$$

(b) Note that $p = e^{-1} \approx 0.36788$

| $n$ | $p_n$ | $\hat{p}_n$ |
|---|---|---|
| 0 | 1.0000 | 0.33333 |
| 1 | 0 | 0.37500 |
| 2 | 0.50000 | 0.36667 |
| 3 | 0.33333 | 0.36806 |
| 4 | 0.37500 | 0.36786 |
| 5 | 0.36667 | 0.36788 |
| 6 | 0.36806 | |
| 7 | 0.36786 | |

4. [10 points] Use Gaussian elimination with backward substitution and two-digit rounding arithmetic to solve the following linear system. Do not reorder the equations.

$$\begin{aligned} E_1 &: -x_1 + 4x_2 + x_3 = 8, \\ E_2 &: \frac{5}{3}x_1 + \frac{2}{3}x_2 + \frac{2}{3}x_3 = 1, \\ E_3 &: 2x_1 + x_2 + 4x_3 = 11. \end{aligned}$$

**Solution:** The augmented matrix is

$$\left[\begin{array}{ccc|c} -1.0 & 4.0 & 1.0 & 8.0 \\ 1.7 & 0.67 & 0.67 & 1.0 \\ 2.0 & 1.0 & 4.0 & 11 \end{array}\right].$$

Performing the operations

$$(E_2 + 1.7E_1) \to (E_2), \quad (E_3 + 2.0E_1) \to (E_3)$$

gives

$$\left[\begin{array}{ccc|c} -1.0 & 4.0 & 1.0 & 8.0 \\ 0.0 & 7.5 & 2.4 & 15 \\ 0.0 & 9.0 & 6.0 & 27 \end{array}\right].$$

Further performing the operation

$$(E_3 - 1.2E_1) \to (E_2),$$

gives

$$\left[\begin{array}{ccc|c} -1.0 & 4.0 & 1.0 & 8.0 \\ 0.0 & 7.5 & 2.4 & 15 \\ 0.0 & 0.0 & 3.1 & 9.0 \end{array}\right].$$

Finally, the back substitution gives

$$\begin{aligned} x_3 &= \frac{9.0}{3.1} \approx 2.9, \\ x_2 &= \frac{15 - 2.4 \times 2.9}{7.5} \approx 1.1, \\ x_1 &= \frac{8.0 - 4.0 \times 1.1 - 1.0 \times 2.9}{-1.0} \approx -0.70. \end{aligned}$$

5. [10 points each]

   (1) Implement Steffensen's method via MATLAB grader.

   (2) Implement Müller's method via MATLAB grader. (See its pseudocode in the textbook.)

   (3) Implement Newton's method with Horner's method via MATLAB grader.

**Solution:**

(1)
```
function sol = steffensen(p0, N, eps)
    p = p0;
    for n=1:N
        p1 = sqrt(exp(p)/3);
        p2 = sqrt(exp(p1)/3);
        pold = p;
        p = p - (p1 - p)^2/(p2 - 2*p1 + p);
        if (abs(p - pold)/abs(p)) < eps
            break;
        end
    end
    sol = [p; n];
end
```

(2)
```
function sol = muller(p0, p1, p2, N, eps)
    f = @(p) p^4 + 2.4*p^3 - 12.95*p^2 - 34.608*p + 91.296;
    fp0 = f(p0);
    fp1 = f(p1);
    fp2 = f(p2);
    for n=3:N
        h1 = p1 - p0;
        h2 = p2 - p1;
        del1 = (fp1 - fp0)/h1;
        del2 = (fp2 - fp1)/h2;
        d = (del2 - del1)/(h2 + h1);
```

```
        b = del2 + h2*d;
        D = sqrt(b^2 - 4*fp2*d);

        if (abs(b + D) > abs(b - D))
            p = p2 - 2*fp2/(b + D);
        else
            p = p2 - 2*fp2/(b - D);
        end
        if (abs(p - p2)/abs(p)) < eps
            break;
        end
        p0 = p1; fp0 = fp1;
        p1 = p2; fp1 = fp2;
        p2 = p; fp2 = f(p);
    end
    sol = [p; n];
end
```

(3)
```
function sol = newton_horner(p0, N, eps)
    a = [1; 0; -4; -3; 5];
    b = zeros(5,1);
    c = zeros(4,1);
    p = p0;
    for n=1:N
        % horner
        b(1) = a(1);
        c(1) = b(1);
        for k=2:4
            b(k) = a(k) + b(k-1)*p;
            c(k) = b(k) + c(k-1)*p;
        end
        b(5) = a(5) + b(4)*p;
        pold = p;
        p = p - b(5)/c(4);
        if (abs(p - pold)/abs(p)) < eps
            break;
        end
    end
    sol = [p; n; b(1:4)];
end
```