HW4

Tuesday, September 26, 2023 2:54 PM

1. In laying water mains, utilities must be concerned with the possibility of freezing. Although soil and weather conditions are complicated, reasonable approximations can be made on the basis of the assumption that soil is uniform in all directions. In that case the temperature in degrees Celsius $T(x,t)$ at a distance x (in meters) below the surface, t seconds after the beginning of a cold snap, approximately satisfies

$$\frac{T(x,t)-T_s}{T_i-T_s} = \operatorname{erf}\left(\frac{x}{2\sqrt{\alpha t}}\right),$$

where T_s is the constant temperature during a cold period, T_i is the initial soil temperature before the cold snap, α is the thermal conductivity (in meters^2 per second), and

$$\operatorname{erf}(t) = \frac{2}{\sqrt{\pi}}\int_0^t \exp(-s^2)ds$$

Assume that $T_i = 20$ [degrees C], $T_s = -15$[degrees C], $\alpha = 0.138 \cdot 10^{-6}$ [meters^2 per second]. *It is convenient to use* **scipy** *to evaluate the erf function.*

For parts (b) and (c), run your experiments with a tolerance of $\epsilon = 10^{-13}$.

(a) We want to determine how deep a water main should be buried so that it will only freeze after 60 *days* exposure at this constant surface temperature.
Formulate the problem as a root finding problem $f(x) = 0$. What is f and what is f'?
Plot the function f on $[0, \bar{x}]$, where $\bar{x}$ is chosen so that $f(\bar{x}) > 0$.

(b) Compute an approximate depth using the Bisection Method with starting values $a_0 =$ 0[meters] and $b_0 = \bar{x}$ [meters].

(c) Compute an approximate depth using Newton's Method with starting value $x_0 = 0.01$[meters]. What happens if you start with $x_0 = \bar{x}$?

a) $$\frac{T(x,t)-T_s}{T_i-T_s} = \operatorname{erf}\left(\frac{x}{2\sqrt{\alpha t}}\right)$$

60 days = 5,184,000 sec = t_f

Want: x_0 s.t.

$$\frac{T(x=x_0, t=t_f)-T_s}{T_i-T_s} = \operatorname{erf}\left(\frac{x_0}{2\sqrt{\alpha t_f}}\right)$$

But, water freezes at $0°C \Rightarrow T(x_0, t_f) = 0$

$$0 = \operatorname{erf}\left(\frac{x_0}{2\sqrt{\alpha t_f}}\right) + \frac{T_s}{T_i - T_s}$$

$$f(x) = \operatorname{erf}\left(\frac{x}{2\sqrt{\alpha t_f}}\right) + \frac{T_s}{T_i - T_s}$$

$$\frac{2}{\sqrt{\pi}} \int_0^{\frac{1}{2\sqrt{\alpha t_f}}x} e^{-s^2}\,ds + \frac{T_s}{}$$

$$= \frac{2}{\sqrt{\pi}} \int_0^{\frac{1}{2\sqrt{\alpha t_f}} x} e^{-s^2}\, ds + \frac{T_s}{T_i - T_s}$$

$$f'(x) = \frac{2}{\sqrt{\pi}} \frac{d}{dx}\left[\int_0^{\frac{1}{2\sqrt{\alpha t_f}} x} e^{-s^2}\, ds \right]$$

let $u = \frac{1}{2\sqrt{\alpha t_f}} x$

$$\frac{du}{dx} = \frac{1}{2\sqrt{\alpha t_f}}$$

$$f'(x) = \frac{2}{\sqrt{\pi}} \frac{d}{du}\left[\int_0^{u} e^{-s^2}\, ds \right] \cdot \frac{du}{dx}$$

$$= \frac{2}{\sqrt{\pi}} e^{-u^2} \cdot \frac{1}{2\sqrt{\alpha t_f}}$$

$$\boxed{= \frac{1}{\sqrt{\pi \alpha t_f}} e^{\frac{-x^2}{4\alpha t_f}}}$$

b) See hw4.py

```
Problem 1b)
Iterations ran: 44, Current approximation: 0.6769618544819593
Error code: 0
```

c)

```
Problem 1c)
Approximate depth with x0 = 0.01: 0.6425178685850313
Error code: 0
Approximate depth with x0 = 1: 0.6421921761374517
Error code: 0
```

Both guesses converge to the root, but with only 1 digit of precision

---

2. Let $f(x)$ denote a function with root α of multiplicity m.

   (a) Write down a formal mathematical definition of what it means for α to be a root of multiplicity m of $f(x)$.

   (b) Show that Newton's method applied to $f(x)$ only converges linearly to the root α.

   (c) Show that the fixed point iteration applied to $g(x) = x - m\frac{f(x)}{f'(x)}$ is second order convergent.

   (d) What does part (c) provide for Newton's method in the case of roots with multiplicity greater than 1?

a)

α is root of f with multiplicity m if:

$$\boxed{\frac{f(x)}{(x-\alpha)^m} = q(x) \quad \text{and} \quad \lim_{x\to\alpha} q(x) \neq 0}$$

b) Let N be the Newton method iteration for f:

$$x_{n+1} = N(x_n), \qquad N(x) = x - \frac{f(x)}{f'(x)}$$

Since $x_{n+1} = N(x_n)$ is a fixed point iteration, it will converge if $0 < |N'(\alpha)| < 1$, where α is the root of α with multiplicity m.

Using $f(x) = (x-\alpha)^m q(x)$

$$\Rightarrow f'(x) = m\,q(x)(x-\alpha)^{m-1} + q'(x)(x-\alpha)^m$$

$$\Rightarrow f''(x) = (x-\alpha)^m\left(\frac{2m\,q'(x)}{x-\alpha} + \frac{(m-1)\,m\,q(x)}{(a-x)^2} + q''(x)\right)$$

$$N(x) = x - \frac{(x-\alpha)^m q(x)}{m q(x)(x-\alpha)^{m-1} + q'(x)(x-\alpha)^{m-1}}$$

$$\Rightarrow N'(x) = \frac{q(x)\left[(m-1)\,m\,q(x) - (x-\alpha)\left(-2m\,q'(x) - (x-\alpha)q''(x)\right)\right]}{\left[m q(x) + (x-\alpha)q'(x)\right]^2}$$

$$N'(\alpha) = \frac{q(\alpha)\left[(m-1)\,m\,q(\alpha)\right]}{m q(\alpha)}$$

$$= \frac{(m-1)m}{m} = \frac{m^2-m}{m} = \frac{m-1}{1}$$

We have $0 < \frac{m-1}{m} < 1$ if $m > 1$.

$\Rightarrow$ Newton's method converges linearly if root α has multiplicity $m > 1$.

c) Let $\tilde{N}(x) = x - m\frac{f(x)}{f'(x)}$

Again using expressions for $f(x)$ and $f'(x)$ from above:

$$\tilde{N}(x) = x - m\frac{(x-\alpha)^m q(x)}{m q(x)(x-\alpha)^{m-1} + q'(x)(x-\alpha)^m}$$

$$= \frac{\alpha m q(x) + x(x-\alpha)q'(x)}{(x-\alpha)q'(x) + m q(x)}$$

$$\Rightarrow \tilde{N}'(x) = \frac{-(x-\alpha)\left[m q(x)\left(-(x-\alpha)q''(x) - 2q'(x)\right) + (m-1)(x-\alpha)q'(x)^2\right]}{\left[(x-\alpha)q'(x) + m q(x)\right]^2}$$

$$\Rightarrow \tilde{N}'(x) = \frac{[\text{illegible}]}{[(x-\alpha)q'(x) + mq(x)]^2}$$

$\tilde{N}'(\alpha) = 0 \Rightarrow$ quadratic or better convergence for $m > 1$

$$\tilde{N}''(x) =$$

$$-\Big[\Big(2(\alpha - x)\Big((x-\alpha)q''(x) + 2mq(x)q'(x) + q'(x)\Big)\Big(mq(x)\Big((a-\alpha)q''(x) - 2q'(x)\Big)$$

$$-(m-1)(\alpha - x)q'(x)^2\Big)\Big) \ldots$$

---

3. Beginning with the definition of order of convergence of a sequence $\{x_k\}_{k=1}^{\infty}$ that converges to α, derive a relationship between the $\log(|x_{k+1} - \alpha|)$ and $\log(|x_k - \alpha|)$. What is the order p in this relationship?

$\{x_k\}_{k=1}^{\infty}$, with $x_k \neq \alpha \ \forall k$ converges to α with order p if $\exists \lambda > 0$ s.t.

$$\lim_{k \to \infty} \frac{|x_{k+1} - \alpha|}{|(x_k - \alpha)^p|} = \lambda > 0$$

Start with:

$$\frac{|x_{k+1} - \alpha|}{|(x_k - \alpha)^p|} = \lambda$$

$$\ln\left(\frac{|x_{k+1} - \alpha|}{|(x_k - \alpha)^p|}\right) = \ln(\lambda)$$

$$\ln(|x_{k+1} - \alpha|) - \ln(|(x_k - \alpha)^p) = \ln(\lambda)$$

$\ln(|x_{k+1}-\alpha|) - p\ln(|x_k-\alpha|) = \ln(\lambda)$

Let $y = \ln(|x_{k+1}-\alpha|)$, $x = \ln(|x_k-\alpha|)$, $b = \ln(\lambda)$

$y - px = b$

$y = px + b$

$\boxed{\text{We have a linear relationship with } p \text{ being the slope.}}$

---

4. There are two ways of improving the convergence of Newton's method when a root has multiplicity greater than 1: Problem 2 c and apply Newton's method to $g(x) = \frac{f(x)}{f'(x)}$.

   In this problem consider finding the root of the function $f(x) = e^{3x} - 27x^6 + 27x^4e^x - 9x^2e^{2x}$ in the interval $(3, 5)$.

   Explore the order of convergence when applying (i) Newton's method, (ii) the modified Newton's method from class and (iii) the modified Newton's method in Problem 2. Which method do you perfer and why?

i)

$f(x) = e^{3x} - 27x^6 + 27x^4e^x - 9x^2e^{2x}$

$= e^{3x} - 9x^2e^{2x} + 27x^4e^x - 27x^6$

$= 1(e^x)^3(-3x^2)^0 + 3(e^x)^2(-3x^2)^1 + 3(e^x)^1(-3x^2)^2 + 1(e^x)^0(-3x^2)^3$

$= \binom{3}{0}(e^x)^3(-3x^2)^0 + \binom{3}{1}(e^x)^2(-3x^2)^1 + \binom{3}{2}(e^x)^1(-3x^2)^2 + \binom{3}{3}(e^x)^0(-3x^2)^3$

Binomial Theorem: $(a+b)^n = \sum_{k=0}^{n} \binom{n}{k} a^{n-k} b^k$

$\Rightarrow$ For $f(x)$, $a = e^x$, $b = -3x^2$, $n = 3$

$\Rightarrow f(x) = (e^x - 3x^2)^3 \Rightarrow f(\alpha) = (e^\alpha - 3\alpha)^3 = 0 \Rightarrow (e^\alpha - 3x^\alpha)^2 = 0$

$\Rightarrow e^\alpha - 3x^\alpha = 0$

$f'(x) = 3(e^x - 3x^2)^2(e^x - 6x)$

$f'(\alpha) = 3(e^\alpha - 3\alpha^2)^2(e^\alpha - 6x) = 0$ bc $(e^\alpha - 3\alpha^2)^2 = 0$

$f''(x) = 3(e^x - 3x^2)(90x^2 - 3e^x(x^2 + 8x + 2) + 3e^{2x})$

$f''(\alpha) = 3(e^\alpha - 3\alpha^2)(90\alpha^2 - 3e^\alpha(\alpha^2 + 8\alpha + 2) + 3e^{2\alpha})$

$= 0$ bc $(e^\alpha - 3\alpha^2) = 0$

$f'''(\alpha) \neq 0$ (verified by graphing)

All of this implies that $m = 3$

For Newton's method $x_{n+1} = N(x_n)$ where $N(x) = x - \dfrac{f(x)}{f'(x)}$,

we will have linear convergence since $m = 3 > 1$, shown in problem 2b).

ii)

Let $\mu(x) = \dfrac{f(x)}{f'(x)} \Rightarrow \mu'(x) = 1 - \dfrac{f(x)f''(x)}{(f'(x))^2}$

$N(x) = x - \dfrac{\mu(x)}{\mu'(x)}$

From hw4.py:

```
Problem 4ii)
Iteration: 0, Approximation: 4.0
Iteration: 1, Approximation: 3.672035696891022
Iteration: 2, Approximation: 3.7295805255614876
Iteration: 3, Approximation: 3.733067726997577
Iteration: 4, Approximation: 3.733079028514993
Iteration: 5, Approximation: 3.733079028632814
Iteration: 6, Approximation: 3.7330790286328144
```

The number of shared digits between each iteration doubles, indicating quadratic convergence.

iii) Modified Newton's method:

$$x_{n+1} = \hat{N}(x_n) = x_n - m\frac{f(x_n)}{f'(x_n)}$$

From hw4.py:

```
Problem 4iii)
Iteration: 0, Approximation: 4.0
Iteration: 1, Approximation: 3.7843611451673693
Iteration: 2, Approximation: 3.7353793750795443
Iteration: 3, Approximation: 3.7330838978740966
Iteration: 4, Approximation: 3.733079028654685
Iteration: 5, Approximation: 3.7330790286328144
Iteration: 6, Approximation: 3.733079028632814
```

The number of shared digits between each iteration doubles, indicating quadratic convergence.

I prefer using $\mu(x)$, as the multiplicity of the root isn't needed but it still converges quadratically.

5. Use Newton and Secant method to approximate the largest root of

$$f(x) = x^6 - x - 1.$$

Start Newton's method with $x_0 = 2$. Start Secant method with $x_0 = 2$ and $x_1 = 1$.

(a) Create at table of the error for each step in the iteration. Does the error decrease as you expect?

(b) Plot $|x_{k+1} - \alpha|$ vs $|x_k - \alpha|$ on log-log axes where α is the exact root for both methods. What are the slopes of the lines that result from this plot? How does this relate to the order?

a) The error decreases as expected

```
Iteration: 0:
Newton Error: 0.6528, Secant Error: 0.34719999999999995
Iteration: 1:
Newton Error: 0.3334282722513089, Secant Error: 0.6528
Iteration: 2:
Newton Error: 0.08353898823906247, Secant Error: 0.33107096774193545
Iteration: 3:
Newton Error: 0.09222904389056352, Secant Error: 0.31652524586882747
Iteration: 4:
Newton Error: 0.1856615672266868, Secant Error: 0.17151105570959868
Iteration: 5:
Newton Error: 0.2108467258294946, Secant Error: 0.2235209346285807
Iteration: 6:
Newton Error: 0.21246947165637087, Secant Error: 0.21352891879474667
Iteration: 7:
Newton Error: 0.21247586149977882, Secant Error: 0.21244731817359686
Iteration: 8:
Newton Error: 0.21247586159848053, Secant Error: 0.21247593435099255
Iteration: 9:
Newton Error: 0.21247586159848053, Secant Error: 0.21247586160350007
```

b) Couldn't get graph code to work $\therefore$ slopes should be 2 for Newton since it is quadratically convergent and 1 for Secant since it is linearly convergent.