Answers to Chapter One HW

STA 321 Topics in Advanced Statistics

2023-02-12

Contents

Chapter 1: Section 1.1

Problem 1. Sec 1.1 1(a)

Textbook Section 1.1.

1(a). Show that the following equations have at least one solution in the given intervals.

$x\cos x - 2x^2 + 3x - 1 = 0$, $[0.2, 0.3]$ and $[1.2, 1.3]$

Solution: We use the mean value theorem to show the existence of the roots on the given interval. Let $f(x) = x\cos x - 2x^2 + 3x - 1$.

```
## define the following function to evaluate the end points of the intervals.
fun = function(x) x*cos(x) - 2*x^2 + 3*x - 1
## calling the above function
fun(0.2)*fun(0.3)
```

```
## [1] -0.001874581
```

```
fun(1.2)*fun(1.3)
```

```
## [1] -0.02047641
```

Since $f(0.2) \times f(0.3) < 0$ and $f(1.2) \times f(1.3) < 0$, equation $f(x) = 0$ has at least one root in each of the two intervals.

Problem 2. Sec 1.1 4(b)

Textbook Section 1.1.

4(b). Find $\max_{a \le x \le b} |f(x)|$ for the following functions and intervals.

b. $f(x) = (4x - 3)/(x^2 - 2x)$ on $[0.5, 1]$

Solution: We first find the extreme values n the following.

$$f'(x) = \frac{4(x^2 - 2x) - (4x - 3)(2x - 2)}{(x^2 - 2x)^2} = \frac{-4x^2 + 6x - 6}{x^4 - 4x^3 + 4x^2}.$$

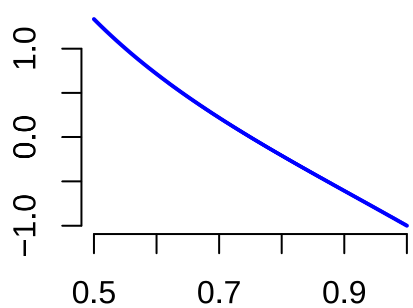
Since $-4x^2 + 6x - 6 = 0$ has no real root. We only need to evaluate the function on the end points of the given interval to determine the maximum of $|f(x)|$. We do in the in the following R code

```
# define the R function
fun1.1.4b = function(x) (4*x - 3)/(x^2 - 2*x)
# Evaluating the function
c(fun1.1.4b(0.5), fun1.1.4b(1))
```

```
## [1]  1.333333 -1.000000
```

Next we plot the function on interval $[0.5, 1]$.

```
xx = seq(0.5, 1, length = 200)
yy = fun1.1.4b(xx)
plot(xx, yy, xlab = "", ylab = "", type = "l", lwd = 2, col = "blue", main = "", bty = "n")
```



Therefore $\max_{a \le x \le b} |f(x)| = 4/3$.

Problem 3 Sec 1.1 11(a)

Textbook Section 1.1.

11(a): Find the third Taylor polynomial $P_3(x)$ for the function $f(x) = (x-1)\ln x$ about $x_0 = 1$.

a. Use $P_3(0.5)$ to approximate $f(0.5)$. Find an upper bound for error $|f(0.5) - P_3(0.5)|$ using the error formula, and compare it to the actual error.

Solution:

$$f(x) = (x-1)^2 - \frac{1}{2}(x-1)^3 + \frac{\frac{2}{\xi^3} + \frac{6}{\xi^4}}{3!}(x-1)^4$$

Where

$$P_3(x) = (x-1)^2 - \frac{1}{2}(x-1)^3 \quad \text{and} \quad R_3(x) = \frac{\frac{2}{\xi^3} + \frac{6}{\xi^4}}{4!}(x-1)^4.$$

Since $\xi \in [0.5, 1]$,

$$|R_3(x) = \frac{\frac{2}{\xi^3} + \frac{6}{\xi^4}}{4!}(x-1)^4 \le \frac{\frac{2}{0.5^3} + \frac{6}{0.5^4}}{4!}(0.5-1)^4 = 0.2916667.$$

The approximated value and the true values are

```r
# Define f(x) and Pn(x)
true.fun = function(x) (x-1)*log(x)
Pn = function(x) (x-1)^2 - 0.5*(x-1)^3
##
true.val = true.fun(0.5)
est.val = Pn(0.5)
est.error = abs(true.val - est.val)
##
kable(c(true.val = true.val, est.val = est.val, est.error = est.error))
```

	x
true.val	0.3465736
est.val	0.3125000
est.error	0.0340736

The true, estimated values and the estimated errors are given in the above table.

Problem 4 Sec 1.1 19

Let $f(x) = e^x$ and $x_0 = 0$. Find the nth Taylor polynomial $P_n(x$ for $f(x)$ about x_0. Find a value of n necessary for $Pn(x)$ to approximate $f(x)$ to within 10^{-6} on $[0, 0.5]$.

Solution. The Taylor expansion of $f(x) = e^x$ has the following form

$$e^x = \sum_{k=0}^{n} \frac{x^k}{k!} + \frac{e^\xi}{(n+1)!}x^{n+1}$$

where $\xi \in (0, x)$. The last term is the remainder which indicates the error when we approximate e^x by $P_n(x) = \sum_{k=0}^{n} \frac{x^k}{k!}$. However, the remainder error on interval $[0, 0.5]$ can be approximated by

$$R_n(x) = \frac{e^\xi}{(n+1)!}x^{n+1} \le \frac{e^{0.5}}{(n+1)!}0.5^{n+1}$$

To find n, we solve the following solving inequality

$$\frac{e^{0.5}}{(n+1)!}0.5^{n+1} < 10^{-10}$$

The algebraic solution to the above inequality is in feasible. We use numerical method to to find the solution.

```
# define the remainder function
Rn = function(x){exp(0.5)*0.5^(x+1) / factorial(x+1)}
# choose a sequence of x to find the smallest value that
# meets the error condition.
x = 1:10
Rn(x)
```

```
##  [1] 2.060902e-01 3.434836e-02 4.293545e-03 4.293545e-04 3.577954e-05
##  [6] 2.555682e-06 1.597301e-07 8.873894e-09 4.436947e-10 2.016794e-11
```

From the above output, the minimum degree of the estimated polynomial is $n = 7$.

Problem 5 Sec 1.1 21

21. The polynomial $P_2(x) = 1 - x^2/2$ is to be used to approximate $f(x) = \cos x$ in $[-1/2, 1/2]$. Find a bound for the maximum error.

Solution: Not that the Taylor expansion of $f(x) = \cos(x)$ has the following general form.

$$\cos(x) = \sum_{k=0}^{\infty} \frac{(-1)^k x^{2k}}{(2k)!}$$

The remainder error term can be written as

$$R_n(x) = \frac{(-1)^k \cos(\xi) x^{2k}}{(2k)!} \quad \text{where} \quad \xi \in [0, x]$$

Since $\cos(x) \le 1$. Therefore,

$$|R_n(x)| = \left|\frac{(-1)^k \cos(\xi) x^{2k}}{(2k)!}\right| \le \frac{x^{2k}}{(2k)!}.$$

When $P_n(x) = 1 - x^2/2$ is used to approximate $f(x) = \cos(x)$, the degree of $P_n(x)$ is $n = 2k = 2$. Therefore, for $x \in [-0.5, 0.5]$, the error bound is given by

$$R_2(x) \le \frac{0.5^{2\times 2}}{(2\times 2)!} = 0.002604167.$$

Chapter 1: Section 1.2

Problem 1 Sec 1.2: 1(a)

1(a): Compute the absolute error and relative error in approximations of p by p^*.

a. $p = \pi, p^* = 22/7$

Solution: We use the following R code to do the calculation.

```
abs.error = abs(pi - 22/7)
rel.error = abs(pi - 22/7)/pi
kable(c(abs.error = abs.error, rel.error = rel.error))
```

	x
abs.error	0.0012645
rel.error	0.0004025

Problem 2 Sec 1.2: 19

19: The two-by-two linear system

$ax + by = e,$

$cx + dy = f,$

where a, b, c, d, e, f are given, can be solved for x and y as follows:

set

$m = c/a$, provided $a \neq 0$;

$d_1 = d - mb$;

$f_1 = f - me$;

$y = f_1/d_1$;

$x = (e - by)/a$.

Solve the following linear systems using four-digit rounding arithmetic.

(a). The system is given by

$1.130x - 6.990y = 14.20$

$1.013x - 6.099y = 14.22$

Solution: The expression of the exact solution to the linear system is given. The error in the solution is due to the rounding error. We use the following R code to do the calculation.

```
# coefficients
a = 1.130
b = -6.990
c = 1.013
d = -6.099
e = 14.20
f = 14.22
####
m  = c/a
d1 = d - m*b
f1 = f - m*e
```

```
y = round(f1/d1,4)              # 4-digit arithmetic requirement
x = round((e - b*y)/a,4)
c(x = x, y = y)
```

```
##       x       y
## 67.6828  8.9101
```

Problem 3 Sec 1.2: 21

21. **(a)** Show that the polynomial nesting technique can also be applied to the evaluation of

$$f(x) = 1.01e^{4x} - 4.62e^{3x} - 3.11e^{2x} + 12.2e^{x} - 1.99.$$

(b). Use three-digit rounding arithmetic, the assumption that $e^{1.53} = 4.62$, and the fact that $e^{nx} = (e^x)^n$ to evaluate $f(1.53)$ as given in part (a).

(c). Redo the calculation in part (b) by first nesting the calculations.

(d). Compare the approximations in parts (b) and (c) to the true three-digit result f (1.53) = -7.61.

Solution: **(a)**. Note that the given $f(x)$ can be re-expressed in the following nested structure.

$$f(x) = (((1.01 \times e^x - 4.62)e^x - 3.11)e^x + 12.2)e^x - 1.99$$

Therefore, we can use nested arithmetic to evaluate the above polynomial.

(b)-(d). This means the direct evaluation by term using 3-digit arithmetic (rounding each term by keeping 3 decimal places). We do the calculation using R in the following. The results are summarized in the output.

```
x = 1.53
E = 4.62
True.val = 1.01*exp(1.53)^4 - 4.62*exp(1.53)^3 - 3.11*exp(1.53)^2 + 12.2*exp(1.53) - 1.99
PartB.val = 1.01*E^4 - 4.62*E^3 - 3.11*E^2 + 12.2*E - 1.99
PartC.val = round((round((round((round(1.01*E,3) - 4.62)*E,3) - 3.11)*E,3) + 12.2)*E,3) - 1.99
##
PartD.Rel.ErrB = abs((True.val - PartB.val)/True.val)
PartD.Rel.ErrC = abs((True.val - PartC.val)/True.val)
c(True.val = True.val, PartB.val = PartB.val, PartC.val = PartC.val, PartD.Rel.ErrB = PartD.Rel.ErrB, Pa
```

```
##       True.val      PartB.val      PartC.val PartD.Rel.ErrB PartD.Rel.ErrC
##    -7.60787145    -7.45124989    -7.46000000     0.02058678     0.01943664
```

The absolute relative error of the nest algorithm is smaller than that of the regular approximation in (b).

Chapter 1: Section 1.3

Problem 1 Sec 1.3: 6(a)

6(a): Find the rates of convergence of the following sequences as $n \to \infty$.

$$\lim_{n\to\infty} \sin(1/n) = 0$$

Solution: We need to use an important result in Calculus: $\lim_{x\to 0}\frac{\sin(x)}{x}=1$. Using this result, we have

$$\lim_{n\to\infty}\frac{\sin(1/n)}{1/n}=1.$$

By definition, $\sin(1/n)=O(1/n)$. That is, the convergence rate of $\sin(1/n)$ as $n\to\infty$ is the as that of $\{1/n\}$.

Problem 2 Sec 1.3: 7(a)

7(a). Find the rates of convergence of the following functions as $h\to 0$. **a**.

$$\lim_{h\to 0}\frac{\sin(h)}{h}=1$$

Solution: We first use the Taylor expansion

$$\frac{\sin(h)}{h}=1-h^2/6+h^4/120-h^6/5040+h^8/362880-\cdots=1+O[h^2]$$

That is the convergence rate of given limit is the same as h^2, $O(h^2)$.

Problem 3 Sec 1.3: 11

11: Construct an algorithm that has as input an integer $n\geq 1$, numbers $x_0,x_1,\cdots,x_n$, and a number x and that produces as output the product $(x-x_0)(x-x_1)\cdots(x-x_n)$.

Solution: We first write the pseudo code for the algorithm.

```
INPUT: vector[x1, x2, ..., xn]
       n
       x
PUTPUT: cumulative product

STEP 1: P = 1     (initial cummulative product)

STEP 2: FOR i < n DO:
        Pi = x - xi
        P = P*Pi
        OUTPUT (P = result)
```

The next example of implementing the algorithm is optional.

Example: Assume that the given vector $V=c(1,3,2,5,7,5,9,4,2,0,7)$. We use n and x as input variables (arguments). The following R function implements the above algorithm.

```r
cumProduct = function(n, x, V){
 if(n > length(V)) stop("n should be smaller than the dimmension of the vector!")
 P = 1
 for(i in 1:n){
   P = P*(x-V[i])
 }
  return(P)
}
```

```r
V = c(1,3,2,5,7,5,9,4,2,0,7)
##
cumProduct( n =8,
            x = 5.5,
            V = V)
```

```
## [1] 77.51953
```

Problem 3 Sec 1.3: 13(a)

a. Suppose that $0 < q < p$ and that $\alpha_n = \alpha + O(n^{-p})$. Show that $\alpha_n = \alpha + O(n^{-q})$.

Solution: Using the definition of **big** O notation and the given condition $\alpha_n = \alpha + O(n^{-p})$, we have

$$|\alpha_n - \alpha| \leq Kn^{-p} < Kn^{-q} \quad \text{since} \quad 0 < q < p.$$

Therefore,

$$\frac{\alpha_n - \alpha}{n^{-q}} \leq K \quad \text{as } n \to \infty.$$

This implies that the rate of convergence is $O(n^{-q})$.