Name: Rui Qiu *Student ID:* 999292509

1	10
2	66
3	5
T	81

STA 410/2102 — First Test — 2015-10-26

For all questions, show enough of your work to indicate how you obtained your answer. No books, notes, or calculators are allowed. You have 50 minutes to write this test.

Question 1: [15 Marks] For some function f (which we can compute), we wish to find a solution to $f(x) = 0$, with an absolute accuracy of 10^{-10} or better. (That is, we wish to find an x such that $|x - x^*| \le 10^{-10}$, where x^* is some true solution to $f(x) = 0$.)

We know that f is continuous, and that $f(10) = 12$ and $f(110) = -2$. We plan to use bisection to solve $f(x) = 0$, starting with the interval from 10 to 110. How many iterations of bisection (each of which evaluates f once, not counting the values at the initial endpoints) will be needed to guarantee that we will obtain a solution with the required accuracy? Explain your answer.

Note: $2^{10} = 1024 \approx 10^3$.

Solution: Since the convergence rate of bisection method is 1. That means each iteration halves the interval and,

$|\varepsilon^{(t+1)}| = c|(\varepsilon^{(t)})|^{\beta}$ where c is a constant, $\beta = 1$.

and $c = \frac{1}{2}$ because each interval shrinks ~~at~~ to $\frac{1}{2}$ of its size before.

The starting interval is $110 - 10 = 100 = 10^2$

$(10^2)\cdot(\frac{1}{2})^n = \boxed{2^{-40} \approx 10^{-3}}$ $\quad 10^{-10}$

$(\frac{1}{2})^n = 10$

$2^{-n} = 10^{-5}$

$-\log_2 10^{-5} = 5\log_2 10 \approx 5 \times 3 = 15$ iterations.

$\frac{100}{2\times 10^{-10}} = 10^{12}/2$

$\doteq 2^{40}/2$

$= 2^{39}$

39 iters

10/15

Question 2: [70 Marks Total] Suppose we model positive real data values, $y_1, \ldots, y_n$, as being independently generated from a gamma distribution, whose density function is

$$f(y) \;=\; \frac{b^a}{\Gamma(a)}\, y^{a-1} \exp(-by)$$

where a and b are positive real model parameters. We wish to find the maximum likelihood estimates for a and b based on the data $y_1, \ldots, y_n$.

Note that $\Gamma(a)$ is the "gamma function". The log of the gamma function is computed by the R function `lgamma`. The derivative of the log of the gamma function is computed by R's `digamma` function, and the second derivative by `trigamma`. All these functions may take a vector as their argument, and return the vector of corresponding function values.

a) [10 Marks] Write an R function below that computes the log likelihood for parameter values a and b given a data vector y. The beginning of the function definition is already shown below, to which you need to add the body of the function. Do not use R's built-in `dgamma` function.

```
log_lik <- function (y,a,b)  {
    sum <- 0
    for (i in 1:n) { sum <- sum + a*log(b) - logt lgamm(a) + (a-1) * log(y[i])-b*y[i]}
    sum
}
```

b) [6 Marks] Write an R function below that computes the derivative of the log likelihood with respect to the a parameter:

```
log_lik_deriv_a <- function (y,a,b)  {
    sum <- 0
    for (i in 1:n) {sum <- sum + log(b) + digamma(a) // gamma(a) + log(y[i])}
    sum
}
```

c) [6 Marks] Write an R function below that computes the derivative of the log likelihood with respect to the b parameter:

```
log_lik_deriv_b <- function (y,a,b)  {
    sum <- 0
    for (i in 1:n) {sum <- sum + a/b - y[i]}
    sum
}
```

d) [6 Marks] Write an R function below that computes the second derivative of the log likelihood with respect to the a parameter:

```
log_lik_deriv2_a <- function (y,a,b)  {
    - n *((trigamma(a) * gamma(a) - digamma(a) * digamma(a)) / (gamma(a))^2)
}
```

e) [6 Marks] Write an R function below that computes the second derivative of the log likelihood with respect to the b parameter:

```
log_lik_deriv2_b <- function (y,a,b)  {
    n * (-a * b ^(-2))
}
```

Suppose we decide to find the maximum likelihood estimates for a and b by using an alternating maximization procedure (also known as non-linear Gauss-Seidel iteration). That is, we alternately maximize the log likelihood with respect to a, with b fixed, then with respect to b, with a fixed, then with respect to a again, etc. We decide to use univariate Newton iteration to maximize with respect to a or with respect to b.

f) [15 Marks] Write an R function below that tries to find a zero of a univariate function `f1`, whose derivative is the function `f2`, starting from the initial point `x`, using `m` iterations of Newton iteration. The beginning of the function is already present; you should write the body of the function. Your function should use only basic R facilities, not R's `nlm` function.

```
newton_iteration <- function (f1,f2,x,m)
    count <- 0
    while (count < m) {
        x <- x - f1(x)/f2(x)
        count <- count +1
    }
    x
}
```

✓

g) [21 Marks] Write an R function that uses the `newton_iteration` function (with `m` set to 10) and the functions for computing the log likelihood and its derivatives to find the maximum likelihood estimate for a and b given a data vector y. Your function should use alternating maximization as described above. The value returned should be a vector of estimates for a and b. The beginning of the functions is already present, with the argument `n` being the number of iterations of alternating maximization to do, and the arguments `a` and `b` being the initial values to use to start the iterations.

```
mle <- function (y,a,b,n) {
    count <- 0
    while (count< n) {
    a<-newton_iteration (log_like_deriv_a, log_like_deriv2_a, a ,10)
    b<-newton_iteration (log_like_deriv_b, log_like_deriv2_b, b, 10)
    count<- count +1
    }
    c(a,b)
}
```

-4

should be
function (a) log_like_deriv_a(y,a,b),
function (a) log_like_deriv2_a(y,a,b),

(b)
(b)

Question 3: [15 Marks] We wish to compute an approximation to the integral of some function, f, of d variables, over the range $(0,1)^d$. The function f is continuous and infinitely differentiable.

For example, if $d = 3$, we wish to compute an approximation to

$$I = \int_0^1 \int_0^1 \int_0^1 f(x,y,z)\, dz\, dy\, dx$$

We approximate I by nested estimates of univariate integrals. So for the $d = 3$ example, we define

$$h(x,y) = \int_0^1 f(x,y,z)\, dz$$

and

$$g(x) = \int_0^1 h(x,y)\, dy$$

so that

$$I = \int_0^1 g(x)\, dx$$

Suppose we approximate all the integrals we need to compute using the Trapezoid Rule. For example, we approximate the integral of $g(x)$ above that gives I using the Trapezoid Rule. Similarly, for every evaluation of $g(x)$ at some point x, we approximate the integral of $h(x,y)$ above that gives $g(x)$ using the Trapezoidal Rule, and similarly for the integral of $f(x,y,z)$ that gives $h(x,y)$. Suppose that for all uses of the Trapezoidal Rule we divide the interval $(0,1)$ into the same number of sub-intervals (and hence the number of points where we evaluate the integrand is this number plus one).

Find the rate at which the error declines as the total number, n, of points at which f is evaluated increases, explaining how you obtained your answer, and how the rate depends on d.

$$\sum_{i=0}^{n-1} \int_0^1 \frac{f(a+ih)+f(a+(i+1)h)}{2} = \int_0^1 \frac{f(a)}{2} + f(a+h)+\cdots+\frac{f(a+(n-1)h)}{2}$$

So for any d, the rate should be $\boxed{n^{-2^d}}$ when using trapezoidal rule.

Solution: For a given n # of points evaluated by Trapezoidal Rule, The rate of error declines is n^{-2}. ✓

4/12

~~So for $g(x)$ on $(0,1)$, the error declines at n^{-2}~~ rate

~~• Then for x-y dimension, we have x fixed, y declines at n^{-2} rate relatively to x~~

~~but we know x declines at n^{-2} rate, so y declines at $(n^{-2})^2 = n^{-4}$~~.

~~Similarly, z has a rate n^{-2} relatively to (x,y), so z's rate should be $(n^{-4})^2 = n^{-8}$~~

error of

• $h(x,y)$ declines at rate of n^{-2} like a normal trapezoidal rule does.

• since error of $g(x)$ should decline as n^{-2} times the decline rate of h, so $g(x)$ declines at n^{-4} rate.

• similarly, error of the whole should decline at $(n^{-4})^2 = n^{-8}$.

Suppose that instead of the Trapezoid Rule we use Simpson's Rule for all integrations. Then how fast will the error declines with n?

Solution: Since the rate of error declines in Simpson's Rule is n^{-4}. ✓

Similarly, the rate for all integrations using Simpson's rule is n^{-4^d}

1/3

~~Sps we approximate all the integrals we need to compute using the Trapezoid Rule. For example, we approximate the integral of $g(x)$ above that gives I using the Trapezoid Rule. Similarly, for every evaluation of $g(x)$ at some pt x, approx the integral of $h(x,y)$ above that gives $g(x)$ using the TR & for integral of $f(x,y,z)$ that gives $h(x,y)$. Sps . .~~

If we evaluate the integrand at m points for each use of TR, then we will evaluate the function f at m^d points.

So we need to use $m = n^{\frac{1}{d}}$

The univariate TR with m pts has error proportional to m^{-2} (large m), which is $n^{-2/d}$

The error at each stage of integration will tend to add (we can't count on being so lucky that they cancel out). So the error in I will decline in proportion to $n^{-2/d}$.

$\Rightarrow$ m^{-4}, $n^{-4/d}$ (Simpson's Rule)