

## Math 426/CISC 410 08F, All sections

R.J. Braun

### Homework 3 Solutions, Hints and Answers

**Problem 1.1.1(c).** The answer is

$$P_6(x) = 1 + \frac{x^2}{2} + \frac{x^4}{24} + \frac{x^6}{720}.$$

**Problem 1.1.2(a).** Note that  $\log$  is denoting the natural log here. Then  $d[\log(1+x)]/dx = (1+x)^{-1}$ ,  $d^2[\log(1+x)]/dx^2 = -(1+x)^{-2}$ , etc.; plugging into the Taylor series formula leads to (1.7).

**Problem 1.3.3.** There are two ways shown here. One is a script file and a separate function file called `samplevar.m`, respectively.

```
% Script: Prob1_3_3s.m
% tries out samplevar and compares with var

x = 1:101;
mys2 = samplevar(x)
theirs2 = var(x)

% randn should have variance of 1
x = randn(21,1);
mys2 = samplevar(x)
theirs2 = var(x)

x = randn(51,1);
mys2 = samplevar(x)
theirs2 = var(x)

x = randn(101,1);
mys2 = samplevar(x)
theirs2 = var(x)

function s2 = samplevar(x)
% compute the variance of a vector
% Input: x, the data
% Output: s2, the variance

n = length(x);
xbar = sum(x)/n;
s2 = sum((x-xbar).^2)/(n-1);
```

The second way is to create a function `testvar` that incorporates `samplevar` into the same file.

```
function testvar
% test out sample var but using a function to do it
% only need to type testvar to the prompt
```

```
x = 1:101;
mys2 = samplevar(x)
theirs2 = var(x)
```

```
% randn should have variance of 1
x = randn(21,1);
mys2 = samplevar(x)
theirs2 = var(x)
```

```
x = randn(51,1);
mys2 = samplevar(x)
theirs2 = var(x)
```

```
x = randn(101,1);
mys2 = samplevar(x)
theirs2 = var(x)
```

```
function s2 = samplevar(x)
% compute the variance of a vector
% Input: x, the data
% Output: s2, the variance
```

```
n = length(x);
xbar = sum(x)/n;
s2 = sum((x-xbar).^2)/(n-1);
```

Sample output from either of these follows. Interestingly, the variance of `randn` isn't too close to unity.

```
>>testvar
mys2 =
    858.5000
theirs2 =
    858.5000
mys2 =
    1.4136
theirs2 =
    1.4136
mys2 =
    0.7125
theirs2 =
    0.7125
```

```

mys2 =
    0.8647
theirs2 =
    0.8647
>>

```

**Problem 1.4.1.** The script below computes the mean and the mean of the absolute values of  $\mathbf{s}$  for each  $N = 10, 20, \dots, 100$  and then plots the results. It's a little bit fancy because an anonymous function is used for `randwalk`, and this function uses the Matlab built-in function `randsrc`. `randsrc` is not in the text, but it generates  $-1$  and  $+1$  with equal probability.

```

% Script: Hw3, prob 1.4.1

NN = 10:10:100;

randwalk = @(x) sum(randsrc(x,1)); % this is the randwalk function!
avg = []; dev = [];

for i=1:length(NN)
    N = NN(i);
    s = zeros(N,1);
    for j=1:10000
        s(j) = randwalk(N);
    end
    avg = [avg mean(s)];
    dev = [dev mean(abs(s))];
end
theory = sqrt(NN);
plot(NN,avg,'-k',NN,dev,'-k',NN,theory,'--k');
xlabel('N')
legend('mean(s)', 'mean(abs(s))', 'sqrt(N)', 'Location', 'East')
figure
plot(NN,dev./theory)
xlabel('N'), ylabel('mean(abs(s))/sqrt(N)')

```

The first plot on the left seems to show that  $\sqrt{N}$  and our computed result have similar shapes. Could they be proportional? The plot of their ratio shows that it is approximately constant, so we the computed result is thus proportional to  $\sqrt{N}$  as suggested in the problem. and what it does in the command window are shown below.

**Problem 1.4.3.** The following script will generate approximations to  $e$ ; it plots and prints results for not very many values of  $N$ , and then makes a plot for many values of  $N$ . This is more than you were asked for.

```

% Script TestESeq.m
% try approximation for exp(1)

```

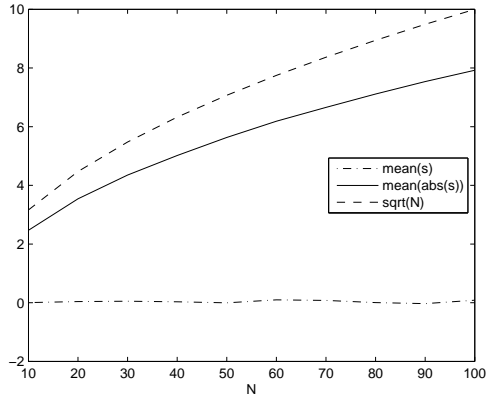


Figure 1: Plots of the mean and the mean of the absolute value of the sum of  $\mathbf{s}$ . The function  $\sqrt{N}$  is shown for comparison.

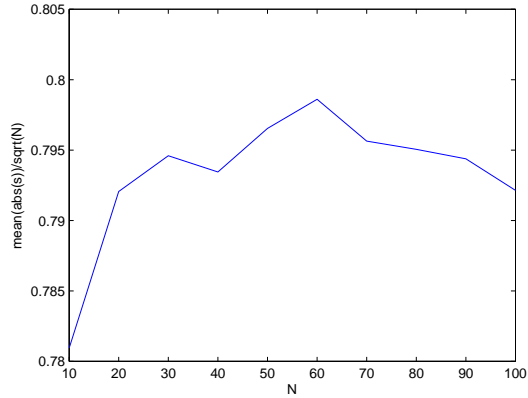


Figure 2: The ratio of computed data to  $\sqrt{N}$  is nearly constant, indicating that there is proportionality between them.

```
%
approx = zeros(1,15);
format short e;
for k = 1:15
    approx(k) = (1+10^(-k))^(10^k);
end
abserror = abs(exp(1)-approx)
loglog(10.^(1:15),abserror);
xlabel('n'); ylabel('|e-(1+1/n)^n|');
title('Error in approximation to e vs. n');
%
% now try curve with vectorization
n = logspace(1,15,1001);
approx2 = (1+1./n).^n;
abserror2 = abs(exp(1)-approx2);
figure
loglog(n,abserror2);
xlabel('n'); ylabel('|e-(1+1/n)^n|');
title('Error in approximation to e vs. n');
```

The resulting error is printed here.

```
>> TestESeq
abserror =
Columns 1 through 6
1.2454e-001 1.3468e-002 1.3579e-003 1.3590e-004 1.3591e-005 1.3594e-006
Columns 7 through 12
1.3433e-007 3.0112e-008 2.2355e-007 2.2478e-007 2.2490e-007 2.4167e-004
Columns 13 through 15
```

2.1718e-003 2.1718e-003 3.1675e-001

>>

The increasing error after  $N = 10^8$  is caused by roundoff error, because the theoretical limit converges to  $e$ . Explaining why this happens is harder; the “solid gold” answer to the problem tries to do this.

One approach is as follows; it is not a proof, but it is consistent with the behavior observed. In the expression  $(1+n^{-1})^n$  with  $n = 10^k$  and  $k = 1, 2, \dots, 15$ , the exponent  $n$  can be represented exactly in binary numbers (it’s an integer), while the term  $n^{-1}$  can’t be. Thus, let’s look at what happens when we put this expression in the computer (at least theoretically). Let

$$\tilde{e}_n = \left(1 + \frac{1}{n} + \epsilon\right)^n;$$

here  $\epsilon$  is the roundoff error from  $1 + (1/n)$  which we estimate to be the unit round  $\mathbf{eps} \approx 10^{-16}$ . Now if use a binomial expansion for this expression, then we find that

$$\tilde{e}_n = 1 + n \left(1 + \frac{1}{n} + \epsilon\right) + \binom{n}{2} \left(1 + \frac{1}{n} + \epsilon\right)^2 + \binom{n}{3} \left(1 + \frac{1}{n} + \epsilon\right)^3 + \dots + \frac{1}{n!} \left(1 + \frac{1}{n} + \epsilon\right)^n.$$

Here the binomial coefficient is

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}.$$

We are considering  $\epsilon$  fixed (given) and  $n$  is increasing. Collecting like terms in powers of  $\epsilon$  yields (not keeping all of them)

$$\tilde{e}_n = \left(1 + 1 + \frac{1}{2} + \frac{1}{6} + \dots + \frac{1}{n!}\right) (1 + \underline{\epsilon n}) + \frac{\epsilon^2 n^2}{2} (2 + \dots) + \dots$$

The first term in parentheses on the right hand side of the equation above converges to  $e$ . Now if  $n = 10^8$ , then  $\epsilon n \approx 10^{-8}$  (underlined in the last equation) and this is the largest term due to roundoff error (anything containing  $\epsilon$ ). This is about the size of the error at the minimum; thus the truncation error from having a finite power can be no larger. If  $n$  is increased, then the term involving  $\epsilon n$  increases and is still the largest contribution compared to the any of the remaining terms with  $\epsilon^k n^k$ ,  $k \geq 2$  and  $n < \epsilon^{-1}$ . This growth of roundoff terms with increasing  $n$  appears to be responsible for the growth of the roundoff error.

**Problem 1.5.1.** (a)  $\kappa = xf(x)/f'(x) = x(1/2)x^{-1/2}/x^{1/2} = 1/2$ . The computation does not have bad relative conditioning for any  $x$ .

(b)  $\kappa = x(1/10)/(x/10) = 1$ . This is not ill-conditioned at any  $x$ .

(c)  $\kappa = x(-\sin(x))/\cos(x) = -x \tan(x)$  this number becomes near any odd multiple of  $\pi/2$ , and so computing  $\cos(x)$  has poor relatively conditioning there.

**Problem 1.5.2.** If only  $b$  varies, then  $\partial x/\partial a = \partial x/\partial c = 0$  and so

$$\left|\frac{\partial x}{\partial b}\right| = \left|\frac{-x}{2ax + b}\right| = \left|\frac{-b \pm \sqrt{b^2 - 4ac}}{2a\sqrt{b^2 - 4ac}}\right|.$$

The last step made use of the formulas for the roots of the quadratic. If we use the relative error, we can use the formula  $\kappa = b(\partial x/\partial b)/x$ , where the names have been changed to reflect the fact that we are interested in  $x(b)$  is playing the role of  $f(x)$ . This gives a relative condition number of

$$\kappa = \left| \frac{b(\partial x/\partial b)}{x} \right| = \left| \frac{-bx}{2ax + b} \frac{1}{x} \right| = \frac{|b|}{\sqrt{b^2 - 4ac}}.$$

**Problem 1.5.3.** Differentiating with respect to  $a_k$  gives

$$0 = a_n n x^{n-1} \frac{\partial x}{\partial a_k} + \dots + x^k + a_k k x^{k-1} \frac{\partial x}{\partial a_k} + \dots + a_1 \frac{\partial x}{\partial a_k};$$

the factors multiplying the partial derivative are  $a_n n x^{n-1} + \dots + a_2 2x + a_1 = p'(x)$ . Then,  $0 = x^k + p'(x) \partial x / \partial a_k$ , and we can solve for the derivative to get our result.