;

CS514 Fall '00

Numerical Analysis

Solution of Homework 1

1. Selected questions from text in Chapter 1

Problem 10:

(a) Cancellation error occurs if $|x|$ is small. To aviod cancellation, one can use

$$f(x) = \frac{(1+x^2)-1}{\sqrt{1+x^2}+1} = \frac{x^2}{\sqrt{1+x^2}+1}$$

which requires only benign arithmetic operations.

(b)

$$(\text{cond } f)(x) = |\frac{xf'(x)}{f(x)}| = 1 + \frac{1}{\sqrt{1+x^2}} \leq 2, \forall x \in \Re.$$

Therefore, f is *well-conditioned*.

(c) This shows a well-condition problem is solved by an *ill-conditioned* algorithm due to the occurrence of cancellation error.

Problem 11:

(i) Let $p_1 = x, \cdots, p_k = fl(p_{k-1}x), \cdots, p_n = fl(p_{n-1}x)$. Then, $p_2 = x^2(1+\epsilon_2)$, $p_3 = x(x^2(1+\epsilon_2))(1+\epsilon_3) = x^3(1+\epsilon_2)(1+\epsilon_3)$, $\cdots$, $p_n = x^n(1+\epsilon_2)\cdots(1+\epsilon_n)$, where $\epsilon_k <$eps. Hence,

$$|\frac{p_n - x^n}{x^n}| = |(1+\epsilon_2)\cdots(1+\epsilon_n) - 1| \leq (n-1)\text{eps}.$$

(ii) $fl(x^n) = e^{n(\ln x(1+\epsilon_1))(1+\epsilon_2)}(1+\epsilon_3), |\epsilon_i| \leq$ eps. Thus,

$$fl(x^n) \approx e^{n \ln x(1+\epsilon_1+\epsilon_2)}(1+\epsilon_3) = e^{n\ln x}e^{(\epsilon_1+\epsilon_2)n\ln x}(1+\epsilon_3)$$
$$\approx x^n(1+(\epsilon_1+\epsilon_2)n\ln x + \epsilon_3),$$
$$|\frac{fl(x^n)-x^n}{x^n}| \approx |(\epsilon_1+\epsilon_2)n\ln x + \epsilon_3| \leq (2n|\ln x| + 1)\text{eps}.$$

Then, (i) is always better than (ii) if $|\ln x| > \frac{1}{2}$ and when $e^{-\frac{1}{2}} < x < e^{\frac{1}{2}}$, it is true if $n \leq \frac{2}{1-2|lnx|}$.

Problem 24: The functions are $\Re \to \Re$. The condition number, $(\text{cond } f)(x) = |\frac{xf'(x)}{f(x)}|$.

(a) $(\text{cond } f)(x) = |\frac{1}{\ln x}|$, $x > 0$. When $x \to 1$, $(\text{cond } f)(x) \to \infty$. Thus, it is *ill-conditioned* when x is near 1.

(b) $(\text{cond } f)(x) = |x \tan x|$, $|x| < \frac{\pi}{2}$. When $|x| \to \frac{\pi}{2}$, $|x \tan x| \to \infty$. Thus, it is *ill-conditioned* when $|x|$ approaches $\frac{\pi}{2}$.

(c) (cond $f)(x) = |\frac{x}{\sin^{-1} x\sqrt{1-x^2}}|$, $|x| < 1$. When $x \to 1$, (cond $f)(x) \to \infty$. Thus, it is *ill-conditioned* when $|x|$ is near 1.

(d) (cond $f)(x) = |\frac{x}{(1+x^2)\sin^{-1}(\frac{x}{\sqrt{1+x^2}})}| < 1$. It is always *well conditioned.*

Problem 25:

(a) (cond $f)(x) = |\frac{1}{n}| \leq 1$, where $x > 0$ and $n > 0$. f is *well conditioned* for all x.

(b) (cond $f)(x) = |\frac{x}{\sqrt{x^2-1}}|$, $x > 1$. When $x \to 1$, (cond $f)(x) \to \infty$. Thus, it is *ill-conditioned* when x approaches 1 and *well conditioned* as $x \to \infty$.

(c) Let $\vec{x} = [x_1, x_2]$.

First, consider each components, x_1 and x_2.

$$\text{(cond } f)(x_1) = \frac{x_1^2}{x_1^2+x_2^2} < 1$$
$$\text{(cond } f)(x_2) = \frac{x_2^2}{x_1^2+x_2^2} < 1$$

Thus, f is *well conditioned* for any x_1 and x_2.

Second, use the *global* definition of the condition number.

$$\text{(cond } f)(\vec{x}) = \frac{\|\vec{x}\|_2 \|f'(\vec{x})\|_2}{|f(\vec{x})|} = 1.$$

The norm used here is Euclidean Norm. Similar result for the condition number can be obtained with other norms.

(d) First, consider each components, x_1 and x_2.

$$\text{(cond } f)(x_1) = |\frac{x_1}{x_1+x_2}|$$
$$\text{(cond } f)(x_2) = |\frac{x_2}{x_1+x_2}|$$

f will be *ill conditioned* if $|x_1 + X_2|$ is very small but $|x_1|$ and $|x_2|$ are not. This is due to the cancellation error.

Second, use the *global* definition of the condition number.

$$\text{(cond } f)(\vec{x}) = \frac{\|\vec{x}\|_* \|f'(\vec{x})\|_*}{|f(\vec{x})|}$$
$$= \frac{\|\vec{x}\|_* \|[1,1]\|_*}{|x_1+x_2|}.$$

The norm can be any norm.

Problem 31: $m_1 = \max_\mu \sum_\nu |a_{\nu\mu}|$.

($\|A\|_1 \leq m_1$) Let $x \neq 0$,

$\|Ax\|_1 = \sum_\nu |\sum_\mu a_{\nu\mu} x_{mu}| \leq \sum_\nu \sum_\mu |a_{\nu\mu}||x_{mu}|$(triangle ineuqality)

$= \sum_\mu |x_{mu}| \sum_\nu |a_{\nu\mu}| \leq \|x\|_1 m_1$.

So, $\frac{\|Ax\|_1}{\|x\|_1} \leq m_1$.

Hence, $\max_{x\neq 0} \frac{\|Ax\|_1}{\|x\|_1} \leq m_1$.

Therefore, $\|A\|_1 \leq m_1$.

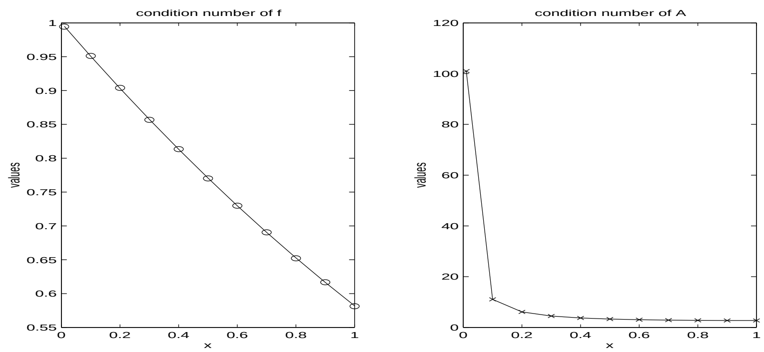

Figure 1: The plots for two condition numbers

($\|A\|_1 \geq m_1$) Let p with $\sum_\nu |a_{\nu p}| = \max_\mu \sum_\nu |a_{\nu\mu}|$.

Consider $y \neq 0$, $y_j = \begin{cases} 1 & j = p \\ 0 & j \neq p \end{cases}$.

Then $\|y\|_1 = 1$.

Now, $\|Ay\|_1 = \sum_\nu |\sum_\mu a_{\nu\mu} x_{mu}| = \sum_{nu} |a_{\nu p}| = \max_\mu \sum_\nu |a_{\nu\mu}|$

$= \|y\|_1 \max_\mu \sum_\nu |a_{\nu\mu}|$.

Hence, $\|A\|_1 \geq \frac{\|Ay\|_1}{\|y\|_1} = \max_\mu \sum_\nu |a_{\nu\mu}| = m_1$.

Therefore, $\|A\|_1 \geq m_1$.

From above, we conclude $\|A\|_1 = m_1$.

Problem 41

(a) $f(x) = 1 - e^{-x}$, for $0 \leq x \leq 1$. Then, $f'(x) = e^{-x}$. So, if $x = 0$, $f(0) = 0$ and (cond f)(x)=$f'(0) = 1$. If $x \neq 0$, (cond f)$(x) = \frac{x}{e^x - 1} = \frac{x}{x + \frac{x^2}{2!} + \cdots} \leq 1$.

(b) $f_A(x) = [1 - e^{-x}(1 + \epsilon_1)](1 + \epsilon_2)$, $|\epsilon_i| <$eps, $i = 1, 2$.

Then, $f_A(x) = 1 - e^{-x} - \epsilon_1 e^{-x} + \epsilon_2(1 - e^{-x})$.

Set $f_A(x) = f(x_A)$, then $x_A = x - \epsilon_1 + \epsilon_2(e^x - 1)$.

Note: during the calculation, we ignore $O(\text{eps}^2)$.

Therefore, $|x - x_A| = |\epsilon_1 - \epsilon_2(e^x - 1)| \leq \text{eps} + (e^x - 1)\text{eps} = e^x \text{eps}$,

$\frac{|x - x_A|}{|x|} \leq \frac{e^x}{x}\text{eps}$,

(cond A)$(x) = \frac{e^x}{x}$.

(c) Figure 1 shows the plots for two condition numbers. f is uniformly well conditioned on [0,1]. But, the algorithm is *ill conditioned* when x is small due to cancellation error.

2. **(a)** Show that the following three schemes can be used to recursively generate the se-

quence $\{\frac{1}{2^n}\}_{n=0}^{\infty}$.

(1) $r_n = (\frac{1}{2})r_{n-1}$, for $n = 1, 2, \cdots$.

sol: This is trivial.

(2) $p_n = (\frac{3}{2})p_{n-1} - (\frac{1}{2})p_{n-2}$, for $n = 2, 3, \cdots$.

sol: Let $p_n = A\frac{1}{2^n} + B$. Then, consider

$p_n = \frac{3}{2}p_{n-1} - \frac{1}{2}p_{n-2}$

$p_n = \frac{3}{2}(A\frac{1}{2^{n-1}} + B) - \frac{1}{2}(A\frac{1}{2^{n-2}} + B)$

$p_n = A(\frac{1}{2^n}) + B$

Set $A = 1$ and $B = 0$, the proof is done.

(3) $q_n = (\frac{5}{2})q_{n-1} - q_{n-2}$, for $n = 2, 3, \cdots$.

sol: omitted since the proof is similar as(2).

(b) Use MATLAB to generate the first ten numerical approximations to the sequence $\{x_n\} = \{\frac{1}{2^n}\}$ using the schemes in (a):

For (1) $r_0 = 0.994$,

For (2) $p_0 = 1$ and $p_1 = 0.497$,

For (3) $q_0 = 1$ and $q_1 = 0.497$.

Produce the numerical results to two tables: one for approximation values and the other for errors. The table formats are as:

Table 1. For approximation values

```
----------------------------------------------
   n |     x     |     r    |    p    |    q
----------------------------------------------
   1 |           |          |         |
 ... |    ...    |    ...   |   ...   |   ...
```

Table 2. For errors, $|x_n - r_n|, |x_n - p_n|$, and $|x_n - q_n|$

```
----------------------------------------------
   n |     x-r      |     x-p      |    x-q
----------------------------------------------
   1 |              |              |
 ... |     ...      |     ...      |    ...
```

Answer: The tables are as followings:

Table 1.

n	x	r	p	q
1	1.0000000000	0.9940000000	1.0000000000	1.0000000000
2	0.5000000000	0.4970000000	0.4970000000	0.4970000000
3	0.2500000000	0.2485000000	0.2455000000	0.2425000000
4	0.1250000000	0.1242500000	0.1197500000	0.1092500000
5	0.0625000000	0.0621250000	0.0568750000	0.0306250000
6	0.0312500000	0.0310625000	0.0254375000	-0.0326875000
7	0.0156250000	0.0155312500	0.0097187500	-0.1123437500
8	0.0078125000	0.0077656250	0.0018593750	-0.2481718750
9	0.0039062500	0.0038828125	-0.0020703125	-0.5080859375
10	0.0019531250	0.0019414062	-0.0040351562	-1.0220429688
11	0.0009765625	0.0009707031	-0.0050175781	-2.0470214844

Table 2.

n	x-r	x-p	x-q
1	0.0060000000	0.0000000000	0.0000000000
2	0.0030000000	0.0030000000	0.0030000000
3	0.0015000000	0.0045000000	0.0075000000
4	0.0007500000	0.0052500000	0.0157500000
5	0.0003750000	0.0056250000	0.0318750000
6	0.0001875000	0.0058125000	0.0639375000
7	0.0000937500	0.0059062500	0.1279687500
8	0.0000468750	0.0059531250	0.2559843750
9	0.0000234375	0.0059765625	0.5119921875
10	0.0000117188	0.0059882812	1.0239960938
11	0.0000058594	0.0059941406	2.0479980469

(c) Use MATLAB to plot the errors of the three schemes and indicate which scheme is stable or unstable.

Answer: The plots are given in Figure 2. Scheme(3) is more unstable than the other two. Scheme(1) is most stable.

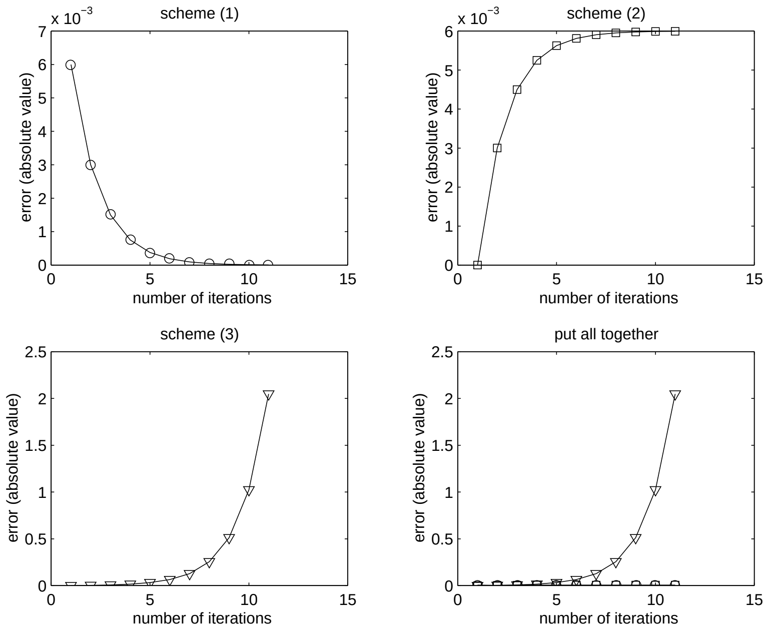

Figure 2: The plots for three schemes

3. **(a)** Consider the evaluation of $I_n = \int_0^1 x^n e^{x-1} dx$, for some $n > 1$. Note that $I_1 = \frac{1}{e} \approx 0.3678794$. Please show that I_n can be evaluated recursively by

$$I_n = 1 - nI_{n-1}.$$

Answer: Use intergration by parts, $\int f'g = fg - \int fg'$ to show. (Let $f' = x^{x-1}dx$ and $g = x^n$.)

(b) Use MATLAB to evaluate I_{12}, output the results to a table,

```
---------------
  n  |   In
  1  |
 ... |   ...
```

plot the result, and discuss its condition (ill-condition or well-condition).

Answer: The table is as:

```
  n |        In
----|----------------
  1 |  0.3678794000
  2 |  0.2642412000
  3 |  0.2072764000
```

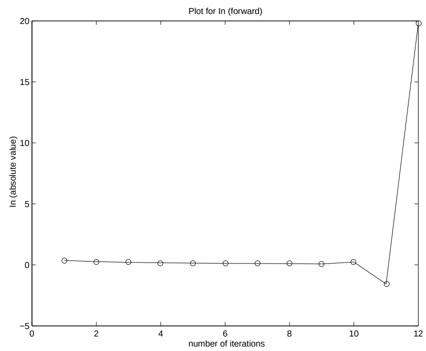

Figure 3: The plots for first method

```
  4 |   0.1708944000
  5 |   0.1455280000
  6 |   0.1268320000
  7 |   0.1121760000
  8 |   0.1025920000
  9 |   0.0766720000
 10 |   0.2332799999
 11 |  -1.5660799991
 12 |  19.7929599890
```

The plot is as Figure 3. It shows that it is *ill conditioned.*

(c) Above method seems ill-conditioned, how to improve it? Also, write a MATLAB program to output the results in a table (i.e. record each iteration result to the table) and plot it. Discuss why the new method is better.

Answer: Use backward analysis instead. Let

$$I_{n-1} = \frac{1-I_n}{n}$$

Since $I_n = \int_0^1 x^n e^{x-1} dx \le \int_0^1 x^n dx = \frac{1}{n-1}$ and $I_{23} \le \frac{1}{24} \approx 0.0437\cdots$, we may start from $I_{23} = 0$. One may select a different start point. The result table is as:

```
  n |         In
----|----------------
 23 |   0.0000000000
 22 |   0.0434782609
 21 |   0.0434782609
 20 |   0.0455486542
```

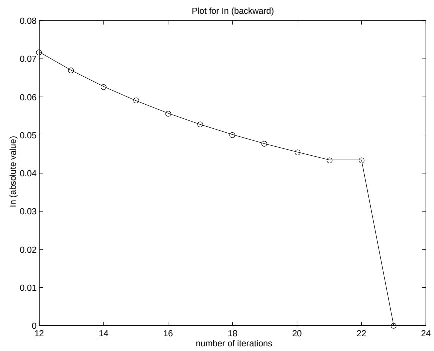

Figure 4: The plots for new method

```
19 |  0.0477225673
18 |  0.0501198649
17 |  0.0527711186
16 |  0.0557193460
15 |  0.0590175409
14 |  0.0627321639
13 |  0.0669477026
12 |  0.0717732536
```

The plot in Figure 4 shows it is *well conditioned*.