Sample Solutions of Programming assignment of Chapter 1

Yu-Rong Wang and Cheng-Hsin Hsu

Note that, the solutions are for your reference only. If you have any doubts about the correctness of the answers, please let the instructor and the TA know. More importantly, like other math questions, the homework questions may be solved in various ways. Do not assume the sample solutions here are the only *correct* answers; discuss with others about alternate solutions.

We will not grade your homework assignment, but you are highly encouraged to discuss with us during the Lab hours. The correlation between the homework assignments and quiz/midterm/final questions is high. So you do want to practice more and sooner.

1 Computer Problem

- **1.9**

(a)

```
function cp01_09 % compute exponential function from series
    definition
disp('(b) Terminate summation when next term does not perturb sum.'
    ); disp(' ');
x = [-20; -15; -10; -5; -1; 1; 5; 10; 15; 20];
approxValue = zeros(size(x)); approxValue2 = zeros(size(x));
approxValue3 = zeros(size(x)); trueValue = exp(x);
for i=1:length(x)
  approxValue(i) = approx_exp(x(i));
  approxValue2(i) = approx_exp2(x(i));
  approxValue3(i) = approx_exp3(x(i));
end; absError = approxValue-trueValue; relError = absError./
    trueValue;
disp('(c) Approximate exp(x) by 1 + x + x^2/2 + ....'); disp(' ')
disp(sprintf(' x True value Approx. value %s', ...
' Abs. error Rel. error'))
fprintf(' %5.1f %14.12e %14.12e %13.6e %13.6e\n', ...
[x'; trueValue'; approxValue'; absError'; relError']); disp(' ');
absError2 = approxValue2-trueValue; relError2 = absError2./
    trueValue;
disp('(d) Use exp(-x) = 1/exp(x) for x < 0.'); disp(' ')
fprintf(' x True value Approx. value %s\n', ...
```

```
' Abs. error Rel. error');
fprintf(' %5.1f %14.12e %14.12e %13.6e %13.6e\n', ...
[x'; trueValue'; approxValue2'; absError2'; relError2']); disp(' ')
   ;
absError3 = approxValue3-trueValue; relError3 = absError3./
   trueValue;
disp('(e) Group same-sign terms to compute exp(x) for x < 0.');
   disp(' ')
disp(sprintf(' x True value Approx. value %s', ...
' Abs. error Rel. error'))
fprintf(' %5.1f %14.12e %14.12e %13.6e %13.6e\n', ...
[x'; trueValue'; approxValue3'; absError3'; relError3']); disp(' ')
   ;
disp('This approach merely postpones cancellation until the end, so
   it does not')
disp('produce any improvement over straightforward summation.')
function [approx] = approx_exp(x)
approx = 1; n = 1; nfact = 1; xn = x; term = xn/nfact;
while approx+term ~= approx
  approx = approx+term; n = n+1; nfact = nfact*n; xn = xn*x; term =
      xn/nfact;
end
function [approx] = approx_exp2(x)
if x < 0, approx = 1/approx_exp(-x);
else approx = approx_exp(x); end
function [approx] = approx_exp3(x)
approx = 1; n = 1; nfact = 1; xn = x; term = xn/nfact;
sum_pos = 1; sum_neg = 0;
while approx+term ~= approx
  if term < 0 sum_neg = sum_neg+term;
  else sum_pos = sum_pos+term; end
  approx = sum_pos+sum_neg;
  n = n+1; nfact = nfact*n; xn = xn*x; term = xn/nfact;
end
```

(b) Terminate summation when next term does not perturb sum.

(c) Approximate $exp(x)$ by $1 + x + x^2/2 + ...$

x	True value	Approx. value	Abs. error	Rel. error
-20.0	2.061153622439e-09	6.138259738609e-09	4.077106e-09	1.978070e+00
-15.0	3.059023205018e-07	3.059300523747e-07	2.773187e-11	9.065597e-05
-10.0	4.539992976248e-05	4.539992943412e-05	-3.283639e-13	-7.232695e-09
-5.0	6.737946999085e-03	6.737946999087e-03	1.439820e-15	2.136883e-13
-1.0	3.678794411714e-01	3.678794411714e-01	1.110223e-16	3.017899e-16
1.0	2.718281828459e+00	2.718281828459e+00	0.000000e+00	0.000000e+00
5.0	1.484131591026e+02	1.484131591026e+02	-2.842171e-14	-1.915040e-16
10.0	2.202646579481e+04	2.202646579481e+04	-7.275958e-12	-3.303280e-16
15.0	3.269017372472e+06	3.269017372472e+06	0.000000e+00	0.000000e+00
20.0	4.851651954098e+08	4.851651954098e+08	-1.192093e-07	-2.457087e-16

(d) Use $exp(-x) = 1/exp(x)$ for $x < 0$.

x	True value	Approx. value	Abs. error	Rel. error
-20.0	2.061153622439e-09	2.061153622439e-09	4.135903e-25	2.006596e-16
-15.0	3.059023205018e-07	3.059023205018e-07	0.000000e+00	0.000000e+00
-10.0	4.539992976248e-05	4.539992976248e-05	1.355253e-20	2.985143e-16
-5.0	6.737946999085e-03	6.737946999085e-03	1.734723e-18	2.574558e-16
-1.0	3.678794411714e-01	3.678794411714e-01	-5.551115e-17	-1.508950e-16
1.0	2.718281828459e+00	2.718281828459e+00	0.000000e+00	0.000000e+00
5.0	1.484131591026e+02	1.484131591026e+02	-2.842171e-14	-1.915040e-16
10.0	2.202646579481e+04	2.202646579481e+04	-7.275958e-12	-3.303280e-16
15.0	3.269017372472e+06	3.269017372472e+06	0.000000e+00	0.000000e+00
20.0	4.851651954098e+08	4.851651954098e+08	-1.192093e-07	-2.457087e-16

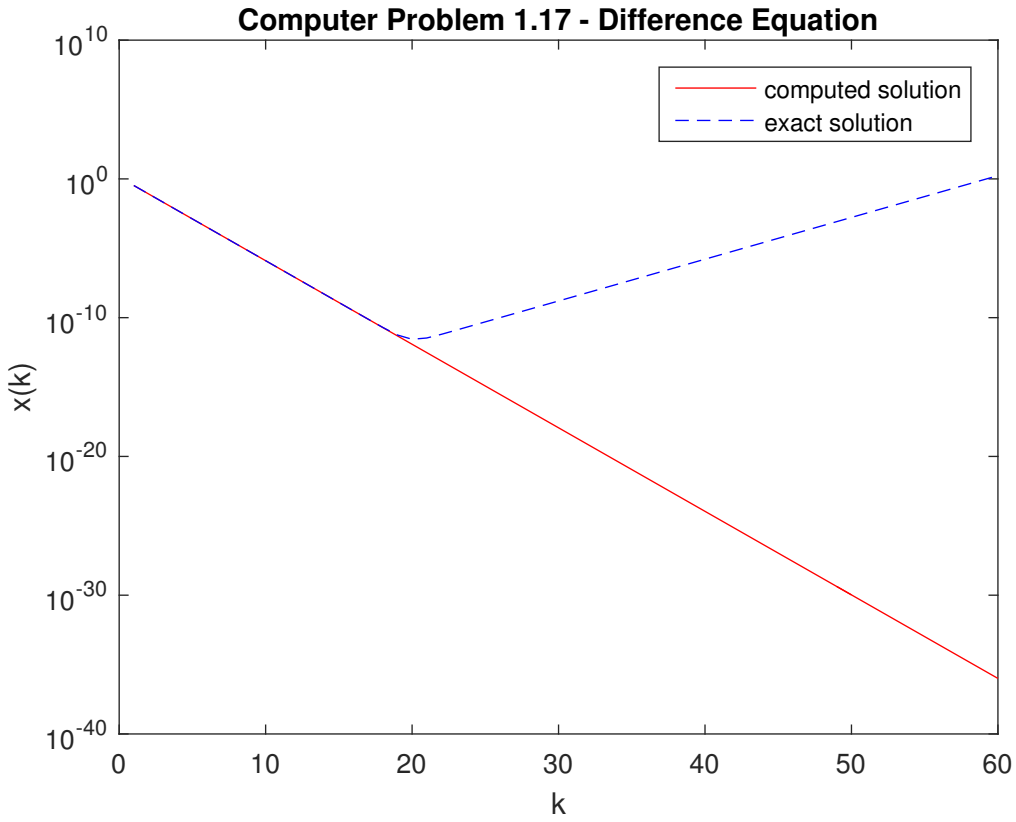
(e) Group same-sign terms to compute $exp(x)$ for $x < 0$.

x	True value	Approx. value	Abs. error	Rel. error
-20.0	2.061153622439e-09	5.960464477539e-08	5.754349e-08	2.791810e+01
-15.0	3.059023205018e-07	3.054738044739e-07	-4.285160e-10	-1.400826e-03
-10.0	4.539992976248e-05	4.539992551145e-05	-4.251038e-12	-9.363534e-08
-5.0	6.737946999085e-03	6.737946999095e-03	9.889659e-15	1.467755e-12
-1.0	3.678794411714e-01	3.678794411714e-01	0.000000e+00	0.000000e+00
1.0	2.718281828459e+00	2.718281828459e+00	0.000000e+00	0.000000e+00
5.0	1.484131591026e+02	1.484131591026e+02	-2.842171e-14	-1.915040e-16
10.0	2.202646579481e+04	2.202646579481e+04	-7.275958e-12	-3.303280e-16
15.0	3.269017372472e+06	3.269017372472e+06	0.000000e+00	0.000000e+00
20.0	4.851651954098e+08	4.851651954098e+08	-1.192093e-07	-2.457087e-16

This approach merely postpones cancellation until the end, so it does not produce any improvement over straightforward summation.

- **1.17**

```
function cp01_17 % solution of difference equation
n = 60; x = zeros(n,1); xtrue = zeros(n,1); x(1) = 1/3; x(2) = 1/12;
for k=3:n
  x(k) = 2.25*x(k-1)-0.5*x(k-2);
end;
xtrue(1:n,1) = (4.^(1-[1:n]'))/3;
semilogy([1:n]',xtrue,'r-',[1:n]',x,'b--')
title('Computer Problem 1.17 - Difference Equation')
xlabel('k'); ylabel('x(k)'); legend('computed solution','exact solution
   ');
disp('Rounding error causes other solution component to dominate
   eventually.');
```



Rounding error causes other solution component to dominate eventually.