HW2

Textbook §1.3 #4.

```
clear;
format long
i = 1;
%sum = 1;
sum = 1/2+1/3;
err = 1;
while (err >= 10^-3)
    i += 1;
    % t  = (-1)^(i+1)*(1)^(2*i-1)/(2*i-1);
    t1 = (-1)^(i+1)*(1/2)^(2*i-1)/(2*i-1);
    t2 = (-1)^(i+1)*(1/3)^(2*i-1)/(2*i-1);
    % sum += t;
    sum += t1 + t2;
    err = abs(4*sum-pi);
end
i
err
```

The smallest number of terms to make the error less than 10^{-3} and the cprresponding error are

$$n = 4,\ error = 7.420918287377 45e - 04.$$

Remark.

In #3, the restuls are

$$n = 1000,\ error = 9.999997499989 81e - 04.$$

The method of #4. converges faster than the one of #3. since $(1/2)^n$ and $(1/3)^n$ decay in the remainder term.

$$
\begin{aligned}
&\left| \sum_{i=1}^{N} (-1)^{i+1} \frac{1^{2i-1}}{2i-1} - \tan^{-1} 1 \right| \\
&= \left| \sum_{i=1}^{N} (-1)^{i+1} \frac{1^{2i-1}}{2i-1} - \sum_{i=1}^{\infty} (-1)^{i+1} \frac{1^{2i-1}}{2i-1} \right| \\
&= \left| \frac{1}{2(N+1)-1} - \frac{1}{2(N+2)-1} + \dots \right| \\
&\geq \frac{1}{2(N+1)-1} - \frac{1}{2(N+2)-1} \approx \frac{2}{4N^2}
\end{aligned}
$$

$$
\begin{aligned}
&\left| \sum_{i=1}^{N/2} (-1)^{i+1} \left(\frac{(1/2)^{2i-1}}{2i-1} + \frac{(1/3)^{2i-1}}{2i-1} \right) - \left(\tan^{-1} \frac{1}{2} + \tan^{-1} \frac{1}{3} \right) \right| \\
&\leq \frac{(1/2)^{2(N/2+1)-1}}{2(N/2+1)-1} + \frac{(1/3)^{2(N/2+1)-1}}{2(N/2+1)-1}
\end{aligned}
$$

Textbook §1.3 #7(c).

There exist $\xi(h), \eta(h)$ between 0 and h such that

$$\begin{aligned}\left|\frac{\sin h - h\cos h}{h}\right| &= \left|\frac{\left(h + \sin^{(3)}(\xi(h))\frac{h^3}{3!}\right) - h\left(1 + \cos^{(2)}(\eta(h))\frac{h^2}{2!}\right)}{h}\right| \\ &\leq \left(\frac{|\sin^{(3)}(\xi(h))|}{3!} + \frac{|\cos^{(2)}(\eta(h))|}{2!}\right)h^2 \\ &\leq \left(\frac{1}{6} + \frac{1}{2}\right)h^2 \\ &\leq \frac{2}{3}h^2\end{aligned}$$

Therefore,

$$\frac{\sin h - h\cos h}{h} = 0 + O(h^2).$$

Textbook §1.3 #10.

By definition,

$$F_1(x) = L_1 + O(x^\alpha) \text{ as } x \to 0$$

$$F_2(x) = L_2 + O(x^\beta) \text{ as } x \to 0$$

imply that there exist $K1, K2 > 0$ such that

$$|F_1(x) - L_1| \leq K_1 |x^\alpha|$$

$$|F_2(x) - L_2| \leq K_2 |x^\beta|.$$

First,

$$\begin{aligned} |F(x) - c_1 L_1 - c_2 L_2| &\leq |c_1||F_1(x) - L_1| + |c_2||F_2(x) - L_2| \\ &\leq |c_1| K_1 |x^\alpha| + |c_2| K_2 |x^\beta| \\ &\leq (|c_1| K_1 + |c_2| K_2) |x^\gamma| \ (\because \ x \to 0 \ \therefore \ 0 < |x| < 1 \ w.l.o.g) \end{aligned}$$

where $|c_1| K_1 + |c_2| K_2 > 0$ since $K_i > 0$ and $c_i \neq 0$, $i = 1, 2$. Therefore,

$$F(x) = c_1 L_1 + c_2 L_2 + O(x^\gamma) \text{ as } x \to 0.$$

Second,

$$\begin{aligned} |G(x) - L_1 - L_2| &\leq |F_1(c_1 x) - L_1| + |F_2(c_2 x) - L_2| \\ &\leq K_1 |(c_1 x)^\alpha| + K_2 |(c_2 x)^\beta| \\ &\leq (K_1 |c_1|^\alpha + K_2 |c_2|^\beta) |x^\gamma| \ (\because \ x \to 0 \ \therefore \ 0 < |x| < 1 \ w.l.o.g) \end{aligned}$$

where $K_1 |c_1|^\alpha + K_2 |c_2|^\beta > 0$ since $K_i > 0$ and $c_i \neq 0$, $i = 1, 2$. Therefore,

$$G(x) = L_1 + L_2 + O(x^\gamma) \text{ as } x \to 0.$$

Textbook §1.3 #15.

(a) Compute the number of operations in the double sum

$$\sum_{i=1}^{n} \sum_{j=1}^{i} a_i b_j$$

directly. Then

$$\text{number of " } * \text{ "} = 1 + 2 + \ldots + n = \frac{n(n+1)}{2}$$
$$\text{number of " } + \text{ "} = 2 + 3 + \ldots + n = \frac{n(n+1)}{2} - 1 = \frac{(n+2)(n-1)}{2}.$$

(b) Modify the double sum into the form

$$\sum_{i=1}^{n} a_i \left(\sum_{j=1}^{i} b_j \right).$$

Then

$$\text{number of " } * \text{ "} = n$$
$$\text{number of " } + \text{ "} = 2 + 3 + \ldots + n = \frac{n(n+1)}{2} - 1 = \frac{(n+2)(n-1)}{2} \text{ (the same).}$$

HW #2.

(a) Fist solve

$$z^2 = \frac{9}{2}z - 2$$

$$\Rightarrow z = \frac{1}{2}, 4.$$

Then the general solution can be given by

$$x_n = c_1\left(\frac{1}{2}\right)^n + c_2 4^n$$

for some constants c_1, c_2. Second solve

$$\begin{cases} c_1\left(\frac{1}{2}\right) + c_2 4 = \frac{1}{5} \\ c_1\left(\frac{1}{2}\right)^2 + c_2 4^2 = \frac{1}{10} \end{cases}$$

$$\Rightarrow c_1 = \frac{2}{5}, c_2 = 0.$$

Therefore,

$$x_n = \frac{2}{5}\left(\frac{1}{2}\right)^n.$$

Approximate $e_n := x_n^h - x_n^e$ by

$$e_n \approx \frac{9}{2}e_{n-1} - 2e_{n-2}.$$

Then the general form of e_n can also be given by

$$e_n \approx d_1\left(\frac{1}{2}\right)^n + d_2 4^n$$

for some constants d_1, d_2. Solve (note that $\frac{1}{5} = (0.\overline{0011})_2$ and $\frac{1}{10} = (0.0\overline{0011})_2$ are not floating-point numbers)

$$\begin{cases} d_1\left(\frac{1}{2}\right) + d_2 4 = fl(\frac{1}{5}) - \frac{1}{5} = \frac{1}{5}(1+\delta_1) - \frac{1}{5} = \frac{1}{5}\delta_1 \\ d_1\left(\frac{1}{2}\right)^2 + d_2 4^2 = fl(\frac{1}{10}) - \frac{1}{10} = \frac{1}{10}(1+\delta_2) - \frac{1}{10} = \frac{1}{10}\delta_2 \end{cases}$$

$$\Rightarrow d_2 \neq 0$$
$$\Rightarrow e_n \approx d_1\left(\frac{1}{2}\right)^n + d_2 4^n \approx d_2 4^n$$
$$\Rightarrow RelErr \approx \frac{d_2 4^n}{\frac{2}{5}\left(\frac{1}{2}\right)^n} \approx c8^n \text{ for some constant } c.$$

The numerical results indeed shows that the relative error is increased by a factor of 8 after each iteration. That is, this recursive relation is unstable.

```
clear;
format long
n = 40;
x = zeros(n,1);
x(1) = 1/5;
x(2) = 1/10;
for i = 3:n
    x(i) = 9 / 2 * x(i-1) - 2 * x(i-2);
    xstar = 2/5*(1/2)^i;
    RelErr = abs(xstar-x(i)) / xstar;
    fprintf('x(%2.0f) = %20.8d, xstar(%2.0f) = %20.8d, RelErr(%2.0f) = %14.4d \n',...
    i, x(i), i, xstar, i, RelErr);
end
```

n	x_n	x_n^*	RelErr
36	-1170.2857	$5.8207661e-12$	$2.011e+14$
37	-4681.1429	$2.910383e-12$	$1.608e+15$
38	-18724.571	$1.4551915e-12$	12867427506772846
39	-74898.286	$7.2759576e-13$	102939420054182768
40	-299593.14	$3.6379788e-13$	823515360433462144

(b) If $x_1 = x_2 = \frac{1}{5}$, then

$$x_n = \frac{12}{35}\left(\frac{1}{2}\right)^n + \frac{1}{140}4^n.$$

Solve

$$\begin{cases} d_1\left(\frac{1}{2}\right) + d_2 4 = \frac{1}{5}\delta_1 \\ d_1\left(\frac{1}{2}\right)^2 + d_2 4^2 = \frac{1}{5}\delta_1 \end{cases}$$

$$
\begin{aligned}
&\Rightarrow d_1 = \frac{12}{35}\delta_1,\ d_2 = \frac{1}{140}\delta_1 \\
&\Rightarrow e_n \approx \left(\frac{12}{35}\left(\frac{1}{2}\right)^n + \frac{1}{140}4^n\right)\delta_1 \\
&\Rightarrow RelErr \approx \frac{\left(\frac{12}{35}\left(\frac{1}{2}\right)^n + \frac{1}{140}4^n\right)\delta_1}{\frac{12}{35}\left(\frac{1}{2}\right)^n + \frac{1}{140}4^n} = \delta_1.
\end{aligned}
$$

The relative error grows linearly in n. That is, this recursive relation is stable.

n	x_n	x_n^*	RelErr
36	$3.3731189e+19$	$3.3731189e+19$	$1.214e-16$
37	$1.3492476e+20$	$1.3492476e+20$	$1.214e-16$
38	$5.3969903e+20$	$5.3969903e+20$	$1.214e-16$
39	$2.1587961e+21$	$2.1587961e+21$	$1.214e-16$
40	$8.6351844e+21$	$8.6351844e+21$	$1.214e-16$

If $x_1 = 1, x_2 = \frac{1}{2}$, then

$$
x_n = 2\left(\frac{1}{2}\right)^n.
$$

Solve (note that 1 and $\frac{1}{2}$ are floating-point numbers)

$$
\begin{cases}
d_1\left(\frac{1}{2}\right) + d_2 4 = fl(1) - 1 = 0 \\
d_1\left(\frac{1}{2}\right)^2 + d_2 4^2 = fl(\frac{1}{2}) - \frac{1}{2} = 0
\end{cases}
$$

$$
\begin{aligned}
&\Rightarrow d_1 = d_2 = 0 \\
&\Rightarrow e_n \approx 0 \\
&\Rightarrow RelErr \approx 0.
\end{aligned}
$$

n	x_n	x_n^*	RelErr
36	$2.910383e-11$	$2.910383e-11$	0000
37	$1.4551915e-11$	$1.4551915e-11$	0000
38	$7.2759576e-12$	$7.2759576e-12$	0000
39	$3.6379788e-12$	$3.6379788e-12$	0000
40	$1.8189894e-12$	$1.8189894e-12$	0000

The instablility of recurrence formula is mainly due to accumulation of floating point error. since 1, 1/2, 9/2 and 2 are all (very short) finite digit floating point numbers. The floating point error will emerge only for very large N when the mantissa part of a_N is longer than the double precision limit. Therefore it stays stable for $n < N$.

HW #3.

- Read the following functions:
  - `loglog` : Produce a 2-D plot using logarithmic scales for both axes. `loglog(x,y)` and `plot(log10(x),log10(y))` have the same graph.
  - `semilogx` : Produce a 2-D plot using a logarithmic scale for the x-axis. `semilogx(x,y)` and `plot(log10(x),y)` have the same graph.
  - `semilogy` : Produce a 2-D plot using a logarithmic scale for the y-axis. `semilogy(x,y)` and `plot(x,log10(y))` have the same graph.
- For $x_n = n$, $y_n = C_1 n^{-k}$, choose the `loglog` scaling

$$\bar{x}_n = \log_{10} x_n,\ \bar{y}_n = \log_{10} y_n.$$

Then

$$\bar{y}_n = \log_{10} C_1 - k \log_{10} n = \log_{10} C_1 - k\bar{x}_n$$

is a line. Plot $y_n = n^{-1}$ for example.

```
x=1:100;
y=x.^(-1);
%Fig.1-1
loglog(x,y);
%Fig.1-2
%plot(log10(x),log10(y));
```

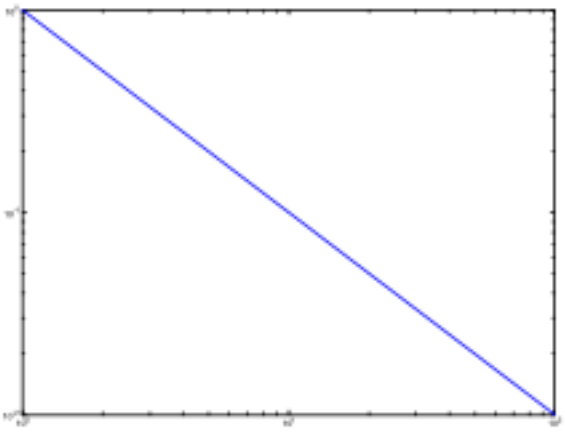


Fig.1-1

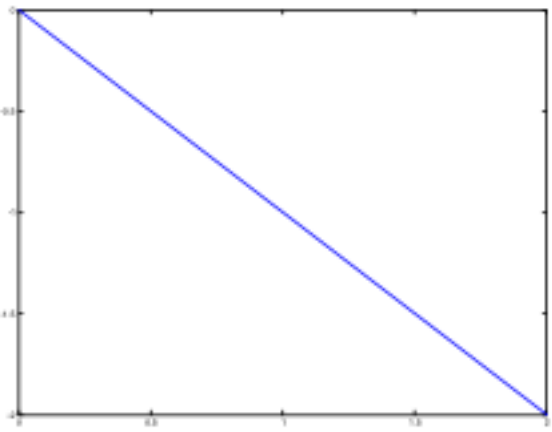


Fig.1-2

- For $x_n = n$, $y_n = C_2 \alpha^n$, choose the `semilogy` scaling

$$\bar{x}_n = x_n,\ \bar{z}_n = \log_{10} z_n.$$

Then

$$\bar{z}_n = \log_{10} C_2 + (\log_{10} \alpha)n = \log_{10} C_2 + (\log_{10} \alpha)\bar{x}_n$$

is a line. Plot $z_n = 0.1^n$ for example.

```
x=1:100;
y=0.1.^x;
%Fig.2-1
semilogy(x,y);
%Fig.2-2
%plot(x,log10(y));
```

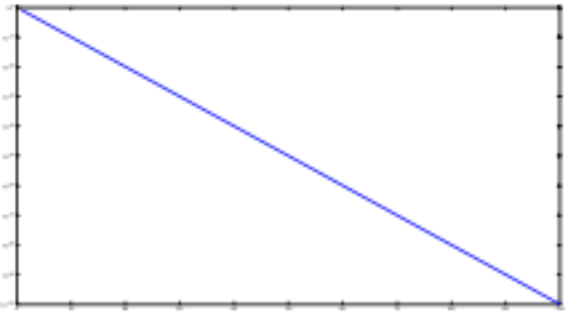


Fig.2-1

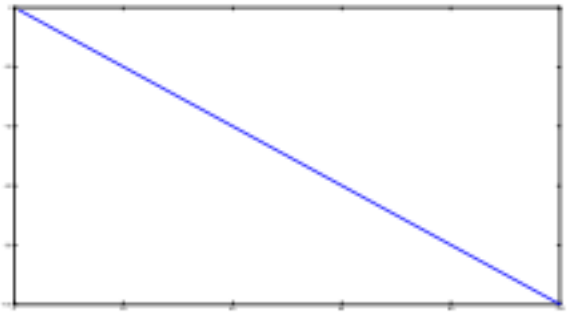


Fig.2-2

- (a)

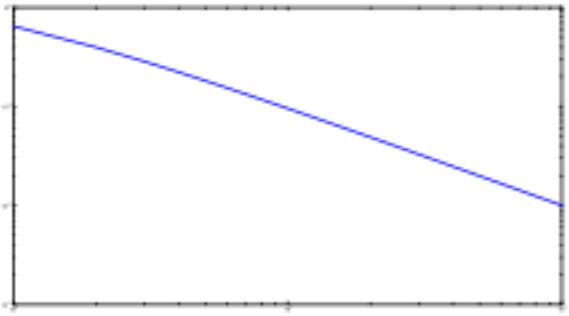
$$\left| \sum_{i=1}^{n} \frac{1}{i^2} - \frac{\pi^2}{6} \right| = \left| \sum_{i=1}^{n} \frac{1}{i^2} - \sum_{i=1}^{\infty} \frac{1}{i^2} \right| = \sum_{i=n+1}^{\infty} \frac{1}{i^2} \leq \int_n^{\infty} \frac{1}{x^2} dx = \frac{1}{n}$$

$$\Rightarrow \sum_{i=1}^{n} \frac{1}{i^2} = \frac{pi^2}{6} + O(\frac{1}{n})$$

```
function p3(N)
x = 1:N;
y = zeros(1,N);
for n = 1:N
    for i = 1:n
        y(n) = y(n) + 1/(i^2);
    end
end

exa = pi^2/6;
y = abs( y - exa );
loglog(x,y);

end
```



- (b) Assume that the leading term of the error is of the form

$$\sum_{i=1}^{n} \frac{1}{i^2} - limit \approx Cn^{-p}.$$

Let

$$S_n = \sum_{i=1}^{n} \frac{1}{i^2}.$$

If the limit is known, then we can use

$$\frac{S_{100} - limit}{S_{200} - limit} \approx \frac{C100^{-p}}{C200^{-p}} = 2^p$$

to obtain p.

If the limit is unknown, then we can use

$$\frac{S_{100} - S_{200}}{S_{200} - S_{400}} \approx \frac{C100^{-p} - C200^{-p}}{C200^{-p} - C400^{-p}} = 2^p$$

to obtain p.

```
function result = p3(n)
sum = 0;
limit = pi^2/6;
for i=1:n
    sum += 1/(i^2);
end
%by using limit
result = sum - limit;

%wihtout using limit
%result = sum;

end
```

By using limit, the result shows that

$$p3(100)/p3(200) = 1.9950 \approx 2^1$$

$$p3(200)/p3(400) = 1.9975 \approx 2^1$$

$$\Rightarrow p = 1.$$

Without using limit, the result shows that

$$(p3(100) - p3(200))/(p3(200) - p3(400)) = 1.9925 \approx 2^1$$

$$\Rightarrow p = 1.$$

Textbook §2.1 #14.

```
function p4(a,b,TOL,N0)
f = @(x) x^2 - 3;

i = 1;
FA = f(a);
p0 = b;
while (i <= N0)
    p = a + (b-a)/2;
    FP = f(p);
    % if ( FP == 0 || (b-a)/2 < TOL)
    % if (abs(p-p0) < TOL)
    % if (abs(p-p0)/abs(p) < TOL)
    if (abs(FP) < TOL)
        fprintf('The approximation solution is %.15f with %f accuracy ...
        after %d iterations.\n', p, TOL, i);
        return;
    end
    i = i +1;
    if (FA * FP > 0)
        a = p;
        FA = FP;
    else
        b = p;
    end
    p0 = p;
end
fprintf('The method failed after N0 iterations, N0=%d\n', N0);
return;

end
```

Run `p4(1,2,10^(-4),100)`.

For the 1st and 2nd stopping conditions, the approximation solution is 1.731994628906250 with 0.000100 accuracy after 14 iterations.

For the 3th and 4th stopping conditions, the approximation solution is 1.732055664062500 with 0.000100 accuracy after 13 iterations.