Computer Experiments

(5)

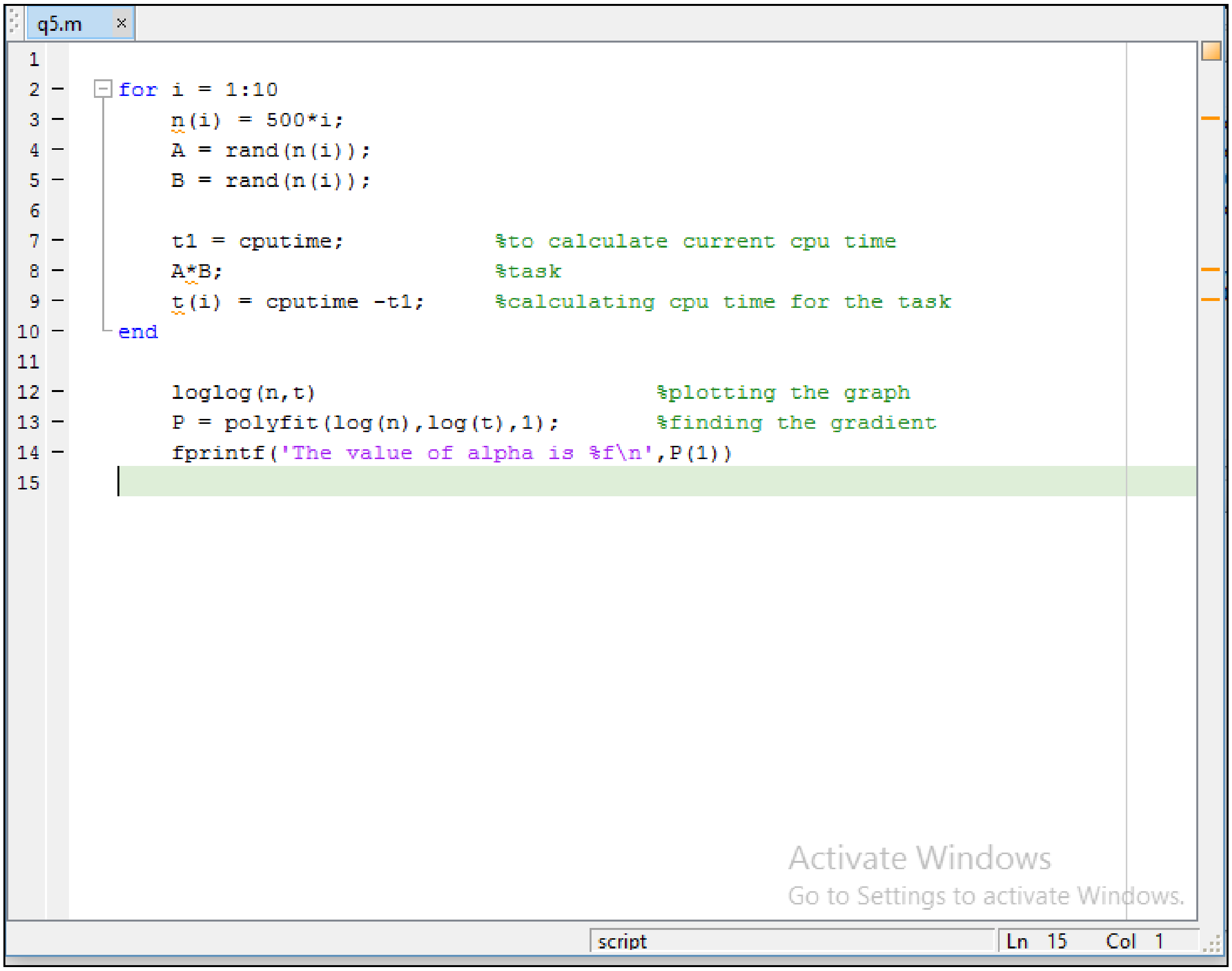


Figure 1: Script screenshot of q5

Code for question (5)

```
for i = 1:10
    n(i) = 500*i;
    A = rand(n(i));
    B = rand(n(i));

    t1 = cputime;           %to calculate current cpu time
    A*B;                    %task
    t(i) = cputime -t1;     %calculating cpu time for the task
end

    loglog(n,t)                         %plotting the graph
    P = polyfit(log(n),log(t),1);       %finding the gradient
    fprintf('The value of alpha is %f\n',P(1))
```

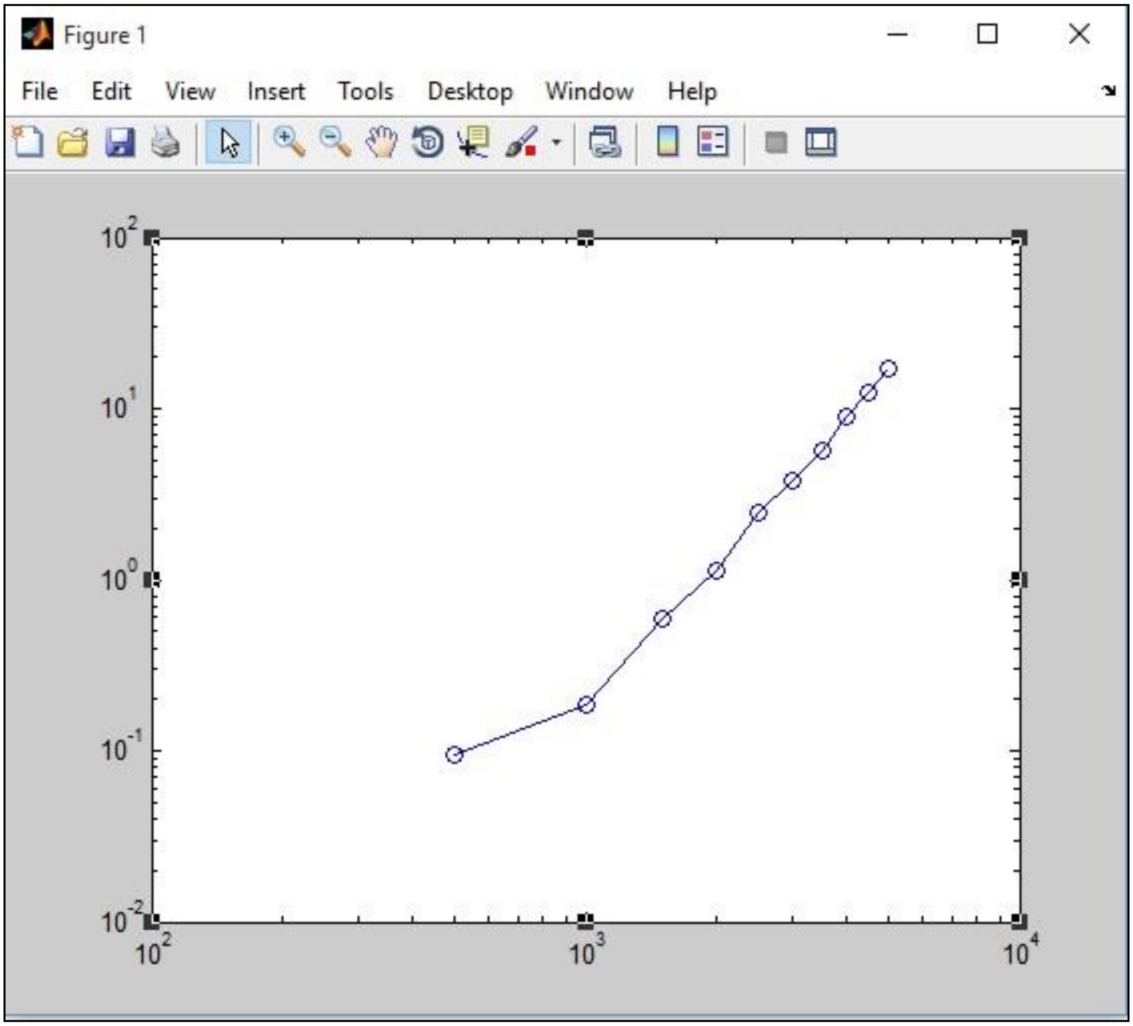


Figure 2: The variation of log(n) against log(t)

Calculations

$$t = Cn^{\alpha}$$
$$\log(t) = \log(C) + \log(n^{\alpha})$$
$$\log(t) = k + \alpha\log(n)$$

α can be estimate by 'polyfit' function

According to the theoretical problem value of alpha is 3

Output

- The value of alpha is 2.475217 (according to the polyfit of graph)

6) (a)

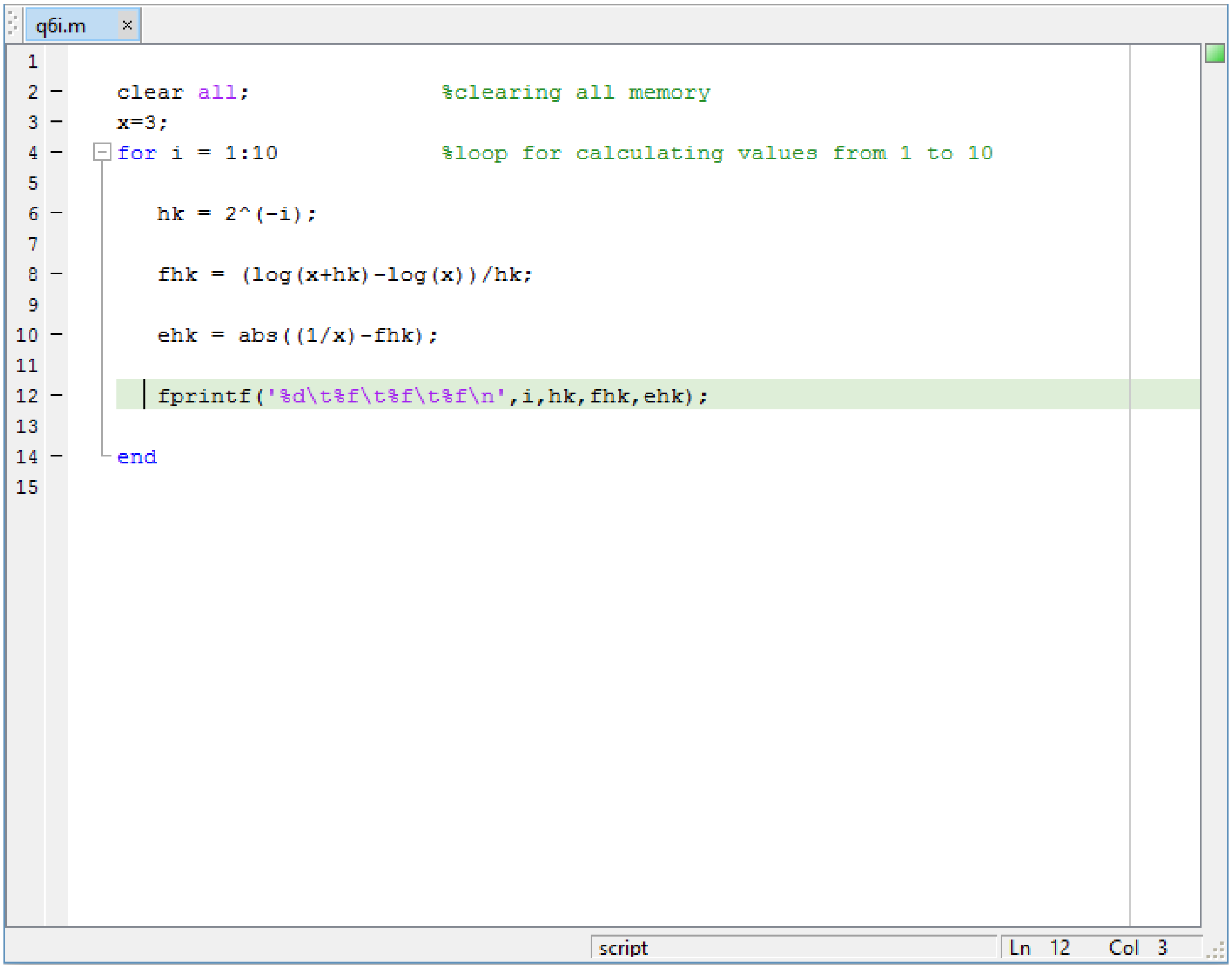


Figure 3: Script screenshot for q6i

Code for question (6) (a)

```
clear all;                  %clearing all memory
x=3;
for i = 1:10                %loop for calculating values from 1 to 10

   hk = 2^(-i);
   fhk = (log(x+hk)-log(x))/hk;
   ehk = abs((1/x)-fhk);
   fprintf('%d\t%f\t%f\t%f\n',i,hk,fhk,ehk);

end
```

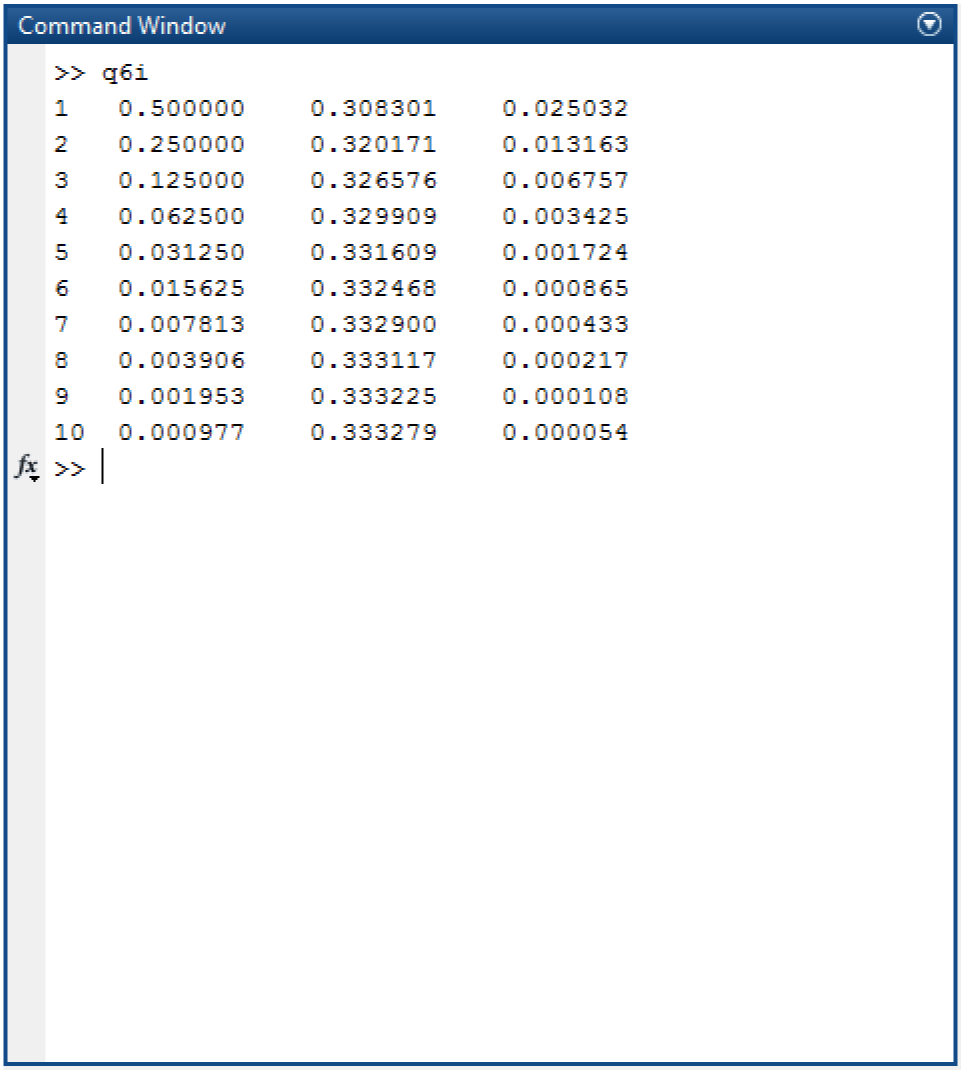


Figure 4: The resulting output screenshot for q6i

k	h_k	$f'_{hk}(x)$	E_{hk}
1	0.500000	0.308301	0.025032
2	0.250000	0.320171	0.013163
3	0.125000	0.326576	0.006757
4	0.062500	0.329909	0.003425
5	0.031250	0.331609	0.001724
6	0.015625	0.332468	0.000865
7	0.007813	0.332900	0.000433
8	0.003906	0.333117	0.000217
9	0.001953	0.333225	0.000108
10	0.000977	0.333279	0.000054

Table 1: The resulting output for q6i

(b)

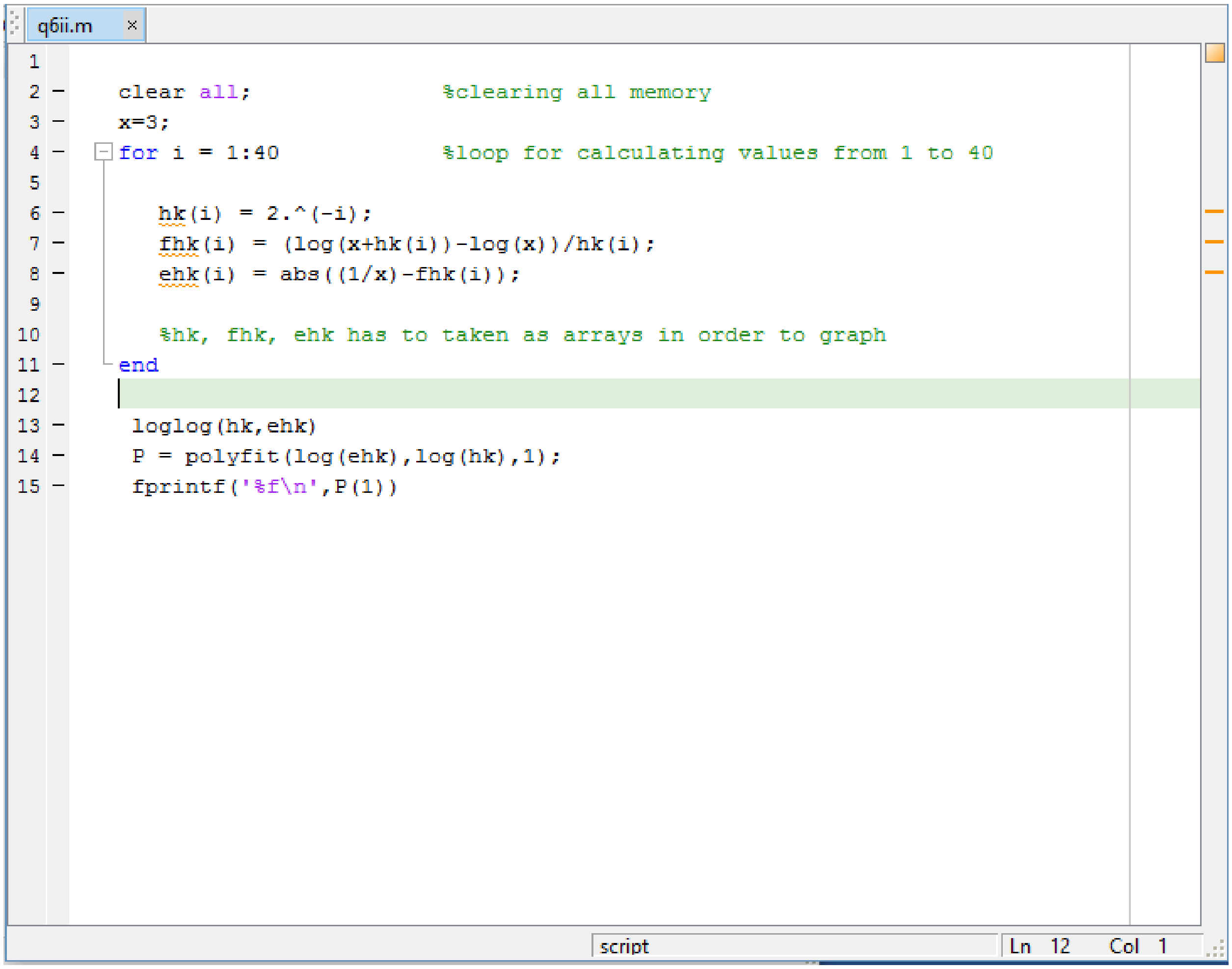


Figure 5: Script screenshot for q6ii

Code for question (6) (b)

```
clear all;                %clearing all memory
x=3;
for i = 1:40              %loop for calculating values from 1 to 40

   hk(i) = 2.^(-i);
   fhk(i) = (log(x+hk(i))-log(x))/hk(i);
   ehk(i) = abs((1/x)-fhk(i));

   %hk, fhk, ehk has to taken as arrays in order to graph
end

 loglog(hk,ehk)
 P = polyfit(log(ehk),log(hk),1);
 fprintf('%f\n',P(1))
```

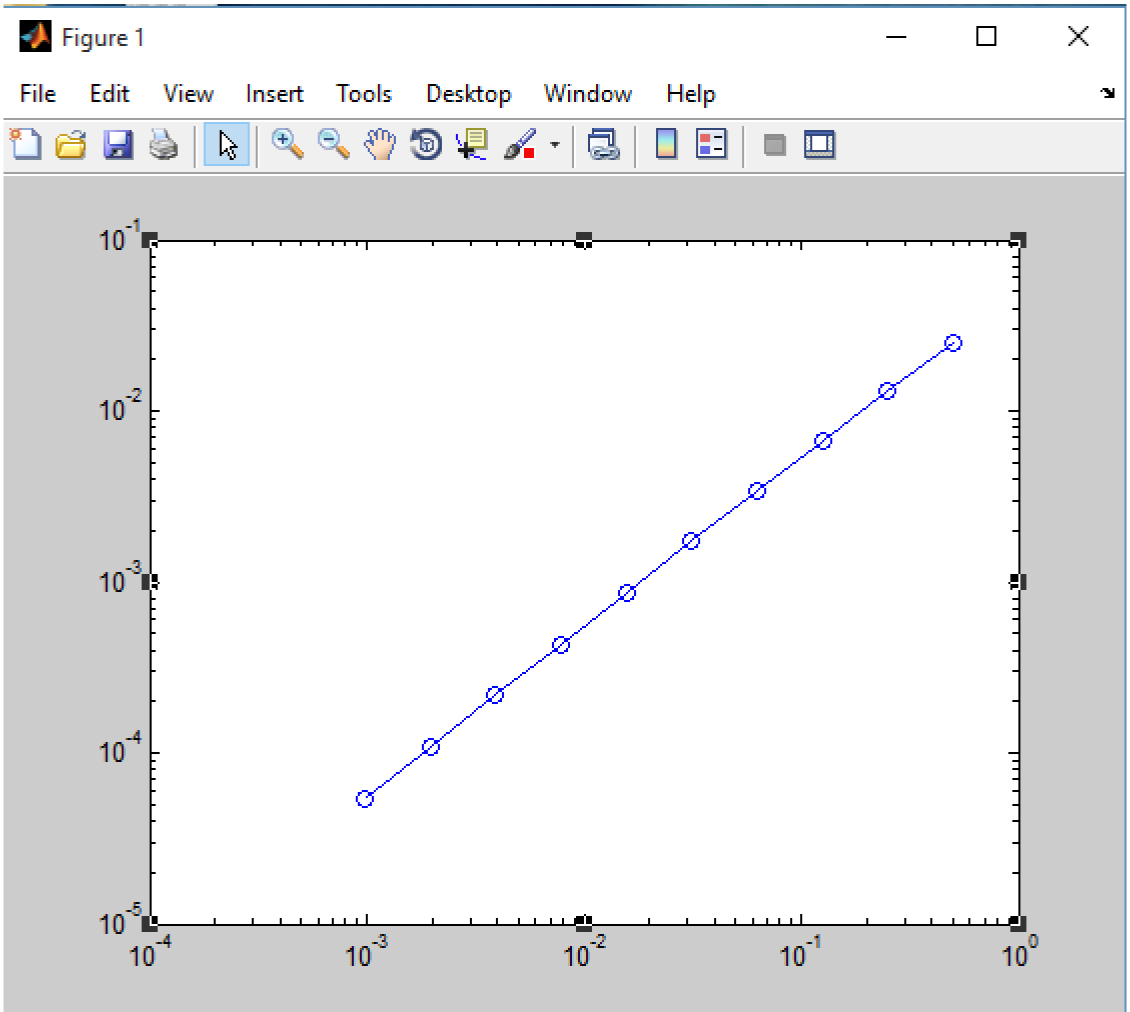


Figure 6: The variation of $\log(E_h)$ against $\log(h)$ (N = 10)

Let's assume that ,

$$E_h \propto h^{\gamma}$$

$$E_h = C\, h^{\gamma}$$

$$\log(E_h) = k + \log(h^{\gamma})$$

$$\log(E_h) = k + \gamma\log(h)$$

- γ can be estimate by 'polyfit' function
- The value of gamma according to the polyfit 1.013012
- If $E_h = O(h)$ value of gamma is 1
- These values above are so close. Therefore, $E_h = O(h)$ is a valid expression

(c)

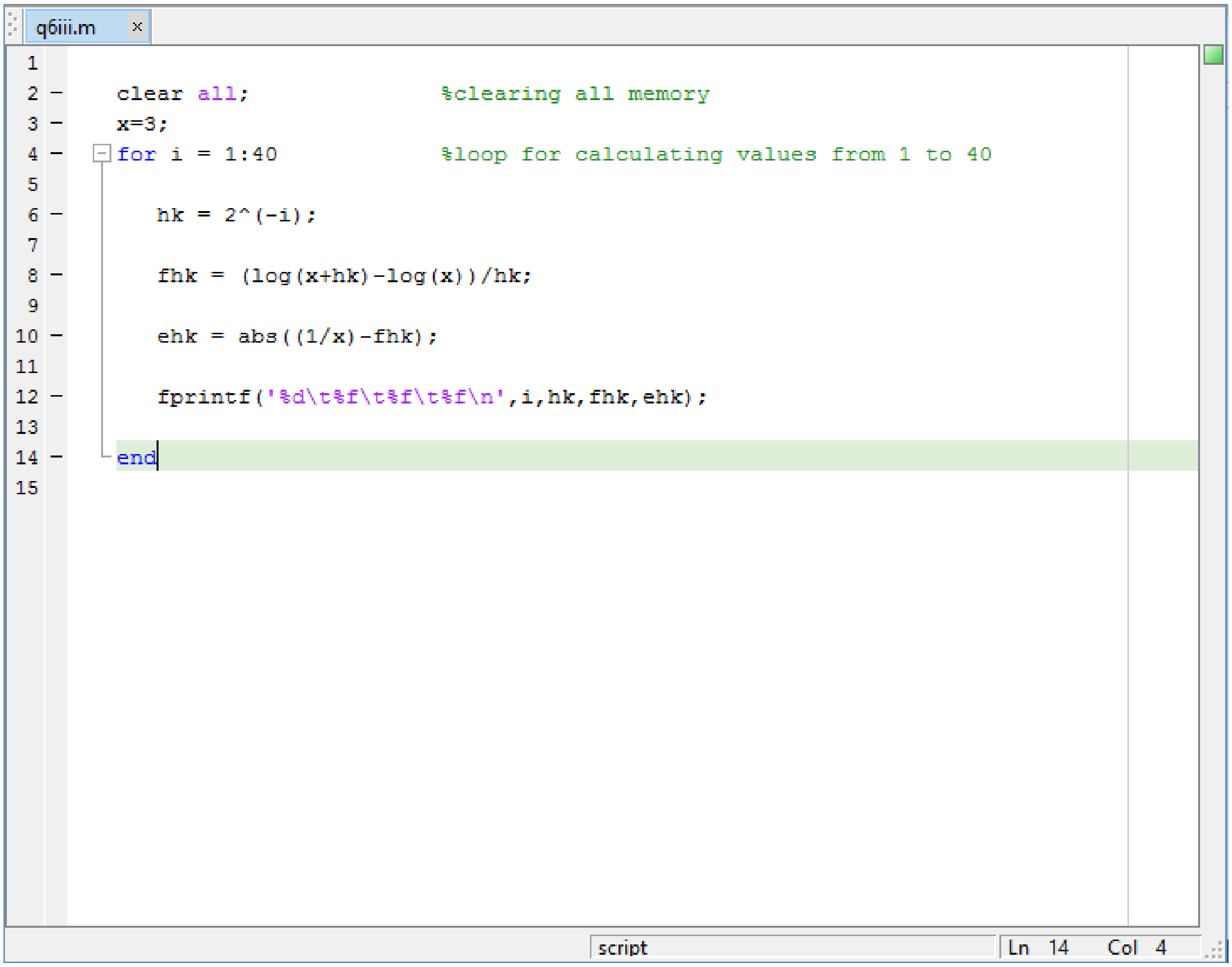


Figure 5: Script screenshot for q6iii

Code for question (6) (c)

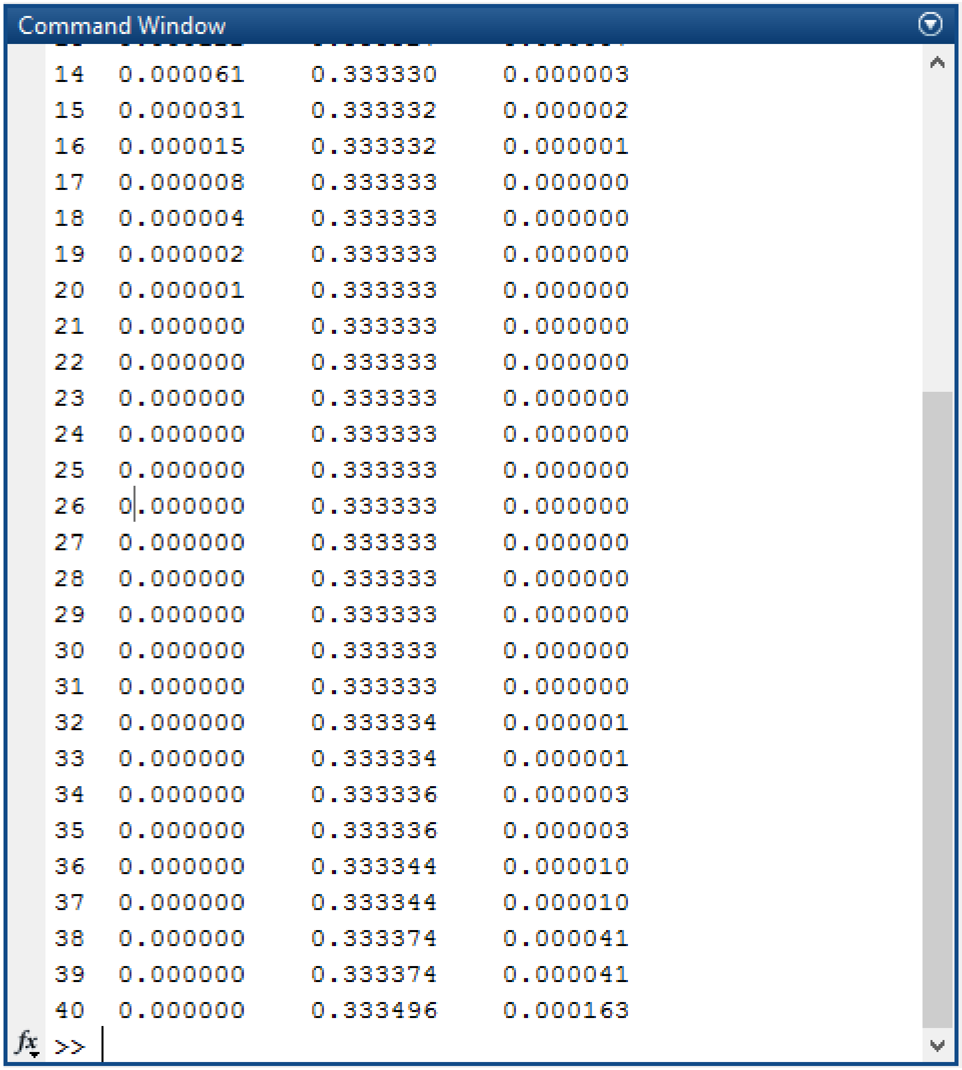
```
clear all;                  %clearing all memory
x=3;
for i = 1:40                %loop for calculating values from 1 to 40

   hk = 2^(-i);
   fhk = (log(x+hk)-log(x))/hk;
   ehk = abs((1/x)-fhk);
   fprintf('%d\t%f\t%f\t%f\n',i,hk,fhk,ehk);

end
```

```
Command Window
>> q6iii
1    0.500000    0.308301    0.025032
2    0.250000    0.320171    0.013163
3    0.125000    0.326576    0.006757
4    0.062500    0.329909    0.003425
5    0.031250    0.331609    0.001724
6    0.015625    0.332468    0.000865
7    0.007813    0.332900    0.000433
8    0.003906    0.333117    0.000217
9    0.001953    0.333225    0.000108
10   0.000977    0.333279    0.000054
11   0.000488    0.333306    0.000027
12   0.000244    0.333320    0.000014
13   0.000122    0.333327    0.000007
14   0.000061    0.333330    0.000003
15   0.000031    0.333332    0.000002
16   0.000015    0.333332    0.000001
17   0.000008    0.333333    0.000000
18   0.000004    0.333333    0.000000
19   0.000002    0.333333    0.000000
20   0.000001    0.333333    0.000000
21   0.000000    0.333333    0.000000
22   0.000000    0.333333    0.000000
23   0.000000    0.333333    0.000000
24   0.000000    0.333333    0.000000
25   0.000000    0.333333    0.000000
26   0.000000    0.333333    0.000000
fx 27   0.000000    0.333333    0.000000
```

Figure 6.1: The resulting output screenshot for q6iii



14	0.000061	0.333330	0.000003
15	0.000031	0.333332	0.000002
16	0.000015	0.333332	0.000001
17	0.000008	0.333333	0.000000
18	0.000004	0.333333	0.000000
19	0.000002	0.333333	0.000000
20	0.000001	0.333333	0.000000
21	0.000000	0.333333	0.000000
22	0.000000	0.333333	0.000000
23	0.000000	0.333333	0.000000
24	0.000000	0.333333	0.000000
25	0.000000	0.333333	0.000000
26	0.000000	0.333333	0.000000
27	0.000000	0.333333	0.000000
28	0.000000	0.333333	0.000000
29	0.000000	0.333333	0.000000
30	0.000000	0.333333	0.000000
31	0.000000	0.333333	0.000000
32	0.000000	0.333334	0.000001
33	0.000000	0.333334	0.000001
34	0.000000	0.333336	0.000003
35	0.000000	0.333336	0.000003
36	0.000000	0.333344	0.000010
37	0.000000	0.333344	0.000010
38	0.000000	0.333374	0.000041
39	0.000000	0.333374	0.000041
40	0.000000	0.333496	0.000163

>>

Figure 6.2: The resulting output screenshot for q6iii (continued)

k	h_k	$f'_{hk}(x)$	E_{hk}
1	0.500000	0.308301	0.025032
2	0.250000	0.320171	0.013163
3	0.125000	0.326576	0.006757
4	0.062500	0.329909	0.003425
5	0.031250	0.331609	0.001724
6	0.015625	0.332468	0.000865
7	0.007813	0.332900	0.000433
8	0.003906	0.333117	0.000217
9	0.001953	0.333225	0.000108
10	0.000977	0.333279	0.000054
11	0.000488	0.333306	0.000027
12	0.000244	0.333320	0.000014
13	0.000122	0.333327	0.000007
14	0.000061	0.333330	0.000003
15	0.000031	0.333332	0.000002
16	0.000015	0.333332	0.000001
17	0.000008	0.333333	0.000000
18	0.000004	0.333333	0.000000
19	0.000002	0.333333	0.000000
20	0.000001	0.333333	0.000000
21	0.000000	0.333333	0.000000
22	0.000000	0.333333	0.000000
23	0.000000	0.333333	0.000000
24	0.000000	0.333333	0.000000
25	0.000000	0.333333	0.000000
26	0.000000	0.333333	0.000000
27	0.000000	0.333333	0.000000
28	0.000000	0.333333	0.000000
29	0.000000	0.333333	0.000000
30	0.000000	0.333333	0.000000
31	0.000000	0.333333	0.000000
32	0.000000	0.333334	0.000001
33	0.000000	0.333334	0.000001
34	0.000000	0.333336	0.000003
35	0.000000	0.333336	0.000003
36	0.000000	0.333344	0.000010
37	0.000000	0.333344	0.000010
38	0.000000	0.333374	0.000041
39	0.000000	0.333374	0.000041
40	0.000000	0.333496	0.000163

Table 2: The resulting output for q6iii

(d) According to the values in the table 2, E_{hk} decreases at the beginning. But after the 32^{nd} iteration, again E_{hk} starts rising.

E_{hk} is the absolute difference of f'(x) and $f'_{hk}(x)$. As it appears up to 32^{nd} iteration E_{hk} reaches to zero and conceptually it should be happened afterwards too. For MATLAB the machine epsilon (the lowest number that can be represented) is about 10^{-16}. As E_{hk} value decreases below this eps value, particular value can't be represent properly and therefore, E_{hk} value varies in an unusual manner.

(e)

q6ii.m

```matlab

clear all;                %clearing all memory
x=3;
for i = 1:40              %loop for calculating values from 1 to 40

   hk(i) = 2.^(-i);
   fhk(i) = (log(x+hk(i))-log(x))/hk(i);
   ehk(i) = abs((1/x)-fhk(i));

   %hk, fhk, ehk has to taken as arrays in order to graph
end

 loglog(hk,ehk)
 P = polyfit(log(ehk),log(hk),1);
 fprintf('%f\n',P(1))
```

Figure 7: Script screenshot for q6iv

Code for question (6) (c)

```
clear all;                  %clearing all memory
x=3;
for i = 1:40                %loop for calculating values from 1 to 40

   hk(i) = 2.^(-i);
   fhk(i) = (log(x+hk(i))-log(x))/hk(i);
   ehk(i) = abs((1/x)-fhk(i));

   %hk, fhk, ehk has to taken as arrays in order to graph
end

 loglog(hk,ehk)
 P = polyfit(log(ehk),log(hk),1);
 fprintf('%f\n',P(1))
```

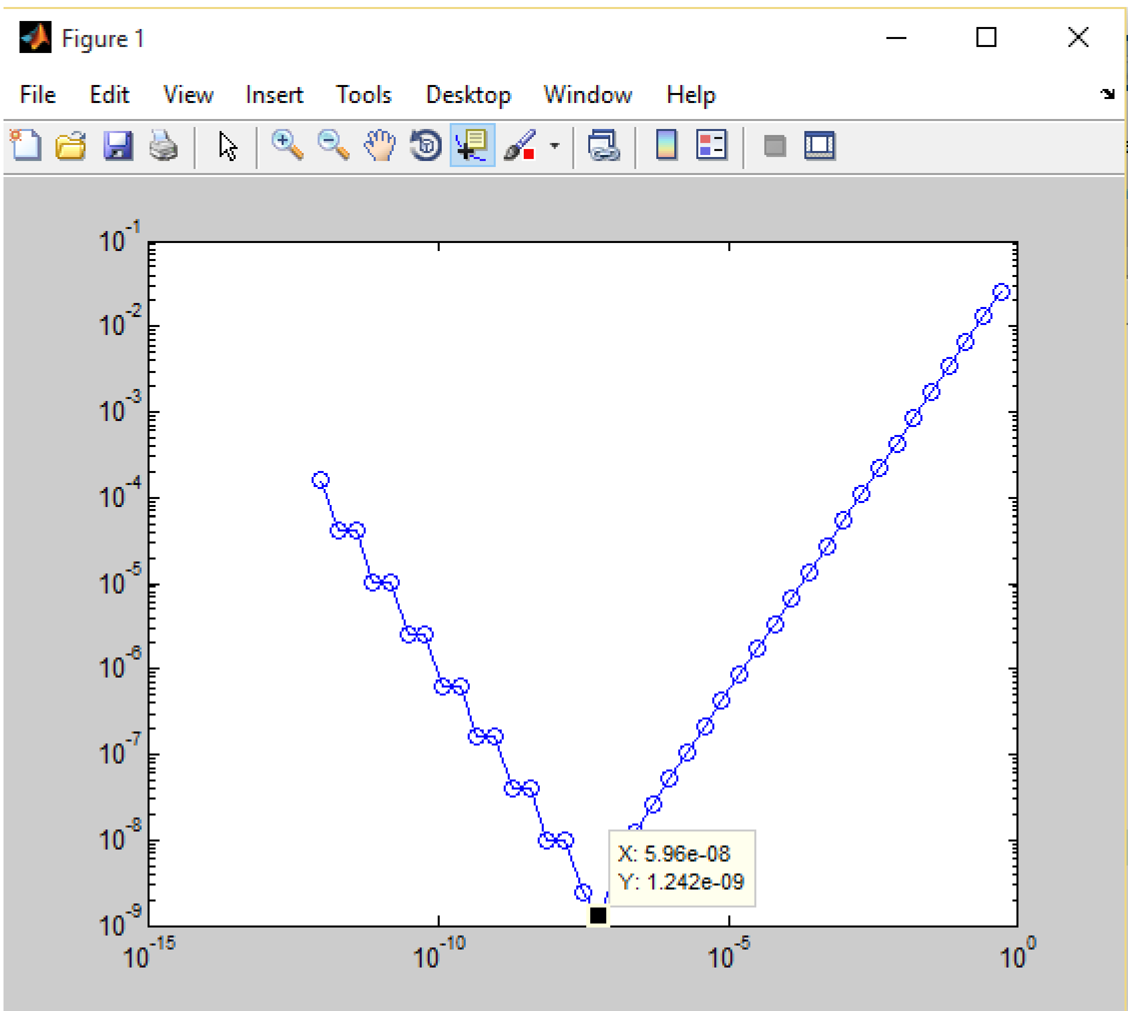


Figure 8: The variation of log(E_h) against log(h) (N = 40)

log(h_{min}) = 5.96 x 10^{-8}

h_{min} = 1.0000