Report of Computing Assignment 7 of MACM 316

During this assignment, I use numerical quadrature to finally compute the integration of the unpleasant function: $I = \int_0^1 \frac{\sin(x^{-1}\ln(x))}{x} dx$. As suggested in the instruction, I write the codes, and when n = 50, 100, …, 500, the results of Q_n are shown below:

n	50	100	150	200
Q_n	-0.454338693601235	-0.456779250065744	-0.457616179248386	-0.458040888016786
n	250	300	350	400
Q_n	-0.458298222249268	-0.458471019505472	-0.458595144421121	-0.458688667274635
n	450	500		
Q_n	-0.458761688610921	-0.458820299431395		

Given Q_n will finally converge, I compute the last three values of Q_n(i.e. n = 498:500) to find the highest precision that *I* can have. Since the result is Q (498) = -0.458818177795791, Q (499) = -0.459902584904609, Q (500) = -0.458820299431395, it is clear that the first two digits of Q_n are the same. And therefore, when n = 500, the highest precision *I* can have is 2 digits.

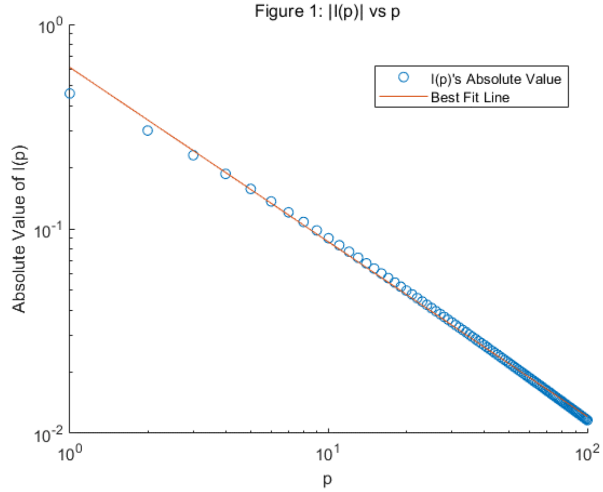
Then I use the Aitken's Δ^2Method to compute the integration. Since at this time, to compute Q_n_hat, I need to compute Q_{n+1} & Q_{n+2} first, and therefore I set the range of n as 50:50:550. I will not use higher upper bound than 550 for reducing the computation time. As suggested in the instruction, I write the codes, and when n = 50, 100, …, 500, the results of Q_n _Hat are shown below:

n	50	100	150	200
$\hat{Q_n}$	-0.459360467172999	-0.459360853057809	-0.459360894574992	-0.459360904981172
n	250	300	350	400
$\hat{Q_n}$	-0.459360908752602	-0.459360910440039	-0.459360911306135	-0.459360911795865
n	450	500		
$\hat{Q_n}$	-0.459360912093478	-0.459360912284717		

Given $\hat{Q_n}$ will finally converge, I compute the last three values of $\hat{Q_n}$(i.e. n = 498:500) to find the highest precision that *I* can have. Since the result is Q_Hat (498) = -0.459360912278480, Q_Hat (499) = -0.459360913331947, Q_Hat (500) = -0.459360912284717, so the first eight digits of $\hat{Q_n}$ are the same. And therefore, when n = 500, the highest precision *I* can have is 8 digits.

Last is to compute the integration that $I(p) = \int_0^1 \frac{\sin(x^{-p}\ln(x))}{x} dx$. The value of I(p) is negative and I(p)→0 as p→ +∞. I use the first method, and set the iteration i to 2000 for higher accuracy, and if i is over 10000, the computation time would be so long. I use p=1:100 for plotting, since in this case, the behaviour is easy to find, and the running time is reduced. I choose the absolute value of I(p) to avoid doing a logarithm of a negative number. The loglog scale is chosen, shown in Figure 1, and the function of |I(p)| seems much like linear, but actually NOT. |I(p)| decreases(converges) to about 10^{-2} as p increases in Figure 1. I also plot the best fit line, according to the output of the *polyfit ()* function, and the rate is -0.8540. (i.e. ln(|I(p)| = -0.8540 ×ln(p)-0.4843) If other axes are chosen, it looked like a logarithmic function's graph, which is not as simple as the loglog one. But as p=1:100, whatever axes are chosen, the value of |I(p)| will converge to about 10^{-2}.

Figure 1: |I(p)| vs p



```
%Codes for Qn and Qn_Hat.
clear all;
close all;
format long;

%Codes for Computing Qn

f = @(x) sin(log(x)./x)./x;
n_array = 50:50:550;
for n = n_array
    a = [];
    I = 0;
    Q = [];
    for i = 1:n
        b = fzero( @(x) x*exp(x)-i*pi , 0);
        a = [a exp(-b)];
        if i==1
          I = I + integral(f,a(i),1);
          Q_0 = I;
        else
          I = I + integral(f, a(i),a(i-1));
          Q = [Q I];
        end
    end
end

%Codes for Computing Qn_Hat via Aitken's Method

for i = 1:500
    Q_Hat(i) = Q(i) - ((Q(i+1)-Q(i))^2)/(Q(i+2)-2*Q(i+1)+Q(i));
end
Q_Hat_0 = Q_0 - (Q(1) - Q_0)^2/(Q(2) - 2*Q(1) + Q_0);
Q = [Q_0 Q];
Q_Hat = [Q_Hat_0 Q_Hat];



%Codes for I(p)
clear all;
close all;

p_value = 1:1:100;
I_p = [];
for poly=p_value
   f = @(x) sin(log(x)./(x.^(poly)))./x;
   I = 0;
   a = [];
   Q = [];
   for i = 1:2000
       b = fzero( @(x) x*exp(poly*x)-i*pi , 0);
       a = [a exp(-b)];
```

```
        if i==1
            I = I + integral(f,a(i),1);
            Q_0 = I;
        else
            I = I + integral(f, a(i),a(i-1));
            Q = [Q I];
        end
    end
    I_p = [I_p I];
end

% Generating Best Fit Line
poly = polyfit(log(p_value), log(abs(I_p)), 1);
x_value = p_value;
y_value = polyval(poly, log(x_value));

scatter(p_value,abs(I_p));
hold on;
plot(x_value, exp(y_value));%y is always negative, and we draw the absolute value of y.
set(gca,'xscale','log');
set(gca,'yscale','log');
title('Figure 1: |I(p)| vs p');
xlabel('p');
ylabel('Absolute Value of I(p)');
legend('I(p)’s Absolute Value','Best Fit Line');
```