

```

Q = [];
f = @(x) (x.^(-1)).*sin((x.^(-1)).*log(x));
g = @(x) (x.^(-1)).*cos((x.^(-2)).*log(x));
for n=100:100:1000
    Q(n/100) = 0;
    a(1) = 1;
    for i=2:n
        b=fzero(@(x) x*exp(x)-i*pi,0);
        a(i)=exp(-b);
        %Compute integral with numerical quadrature
        I = integral(f, a(i-1), a(i));
        %Compute Q(n) sum
        Q(n/100) = Q(n/100) + I;
    end
end

%How many digits of I can you accurately compute?
disp("Last 5 original Q for original function:");
for i=length(Q)-5:length(Q)
    format longg
    disp(Q(i));
end

q=[];
N=7000;
for n=100:100:N
    Q(n/100) = 0;
    a(1) = 1;
    for i=2:n
        b=fzero(@(x) x*exp(x)-(i-1)*pi,0);
        a(i)=exp(-b);
        %Compute integral with numerical quadrature
        I = integral(f, a(i-1), a(i));
        %Compute Q(n) sum
        Q(n/100) = Q(n/100) + I;
    end
    if (n/100)>3
        %Compute Aitken's triangle square method
        q((n/100)-3) = Q((n/100)-3) - (( Q((n/100)-2) - Q((n/100)-1) ).^2 / ( Q(n/100) -
2.*Q((n/100)-1) + Q((n/100)-2) ));
    end
end

disp("Last 5 Q hat for original function:")
for i=length(q)-5:length(q)
    format longg
    disp(q(i));
end

%Should be 0.279506096632
q=[];

```

```
N=20000;
for n=100:100:N
    Q(n/100) = 0;
    a(1) = 1;
    for i=2:n
        b=fzero(@(x) (x/2)*exp(x)-((i-1)*pi),0);
        a(i)=exp(-b/2);
        %Compute integral with numerical quadrature
        I = integral(g, a(i-1), a(i));
        %Compute Q(n) sum
        Q(n/100) = Q(n/100) + I;
    end
    if (n/100)>3
        %Compute Aitken's triangle square method
        q((n/100)-3) = Q((n/100)-3) - (( Q((n/100)-2) - Q((n/100)-1) ).^2 / ( Q(n/100) - 2.*Q((n/100)-1) + Q((n/100)-2) ));
    end
end

q=-q;

disp("Last 5 Q hat for secondary function:")
for i=length(q)-5:length(q)
    format longg
    disp(q(i));
end
```