Math 406 Homework 2

Xander Naumenko

10/10/23

Question 1. See tables 1, 2, 3, 4 and 5 for the required tables. For the log-log plots see figures 1, 2, 3, 4 and 5.

The code for this question can be seen here:

```
disp('midpoint')
numerical_integration(-1, 1, @(x) 1/(1+x^2)^0.5, 32, 1);
disp('trap')
numerical_integration(-1, 1, @(x) 1/(1+x^2)^0.5, 32, 2);
disp('simpson')
numerical_integration(-1, 1, @(x) 1/(1+x^2)^0.5, 32, 3);
disp('gauss_legendre')
numerical_integration(-1, 1, @(x) 1/(1+x^2)^0.5, 32, 4);
disp('real')
% -2*log(2^0.5-1)

functions = {@(x) 1/(1+x^2)^0.5, @(x) sin(2*x)^2, @(x) x^(4.0/3), @(x) x^(1.0/3), @
bounds = [-1,1; 0,pi; 0,1; 0,2; 0,1];
real_ans = [-2*log(2^0.5-1), pi/2, 3.0/7, 3/(2^(2.0/3)), pi^0.5/2];
Ns = [2,4,16,32];
tables = zeros(5,4,5);
for i = 1:length(functions)
    f = functions{i};
    for j = 1:length(Ns)
        N = Ns(j);
        for choice = [1,2,3,4]
            if i ~= 5 || (choice ~= 2 && choice ~= 3)
                tables(i,j,choice) = numerical_integration(bounds(i,1),bounds(i,2),
            end
        end
        tables(i,j,5) = real_ans(i);
    end
end

for i = 1:5
    disp(array2table(squeeze(tables(i,:,:))))
end
```

```
% Calculate the errors
errors = abs(tables - real_ans');

methods = {'Midpoint', 'Trapezium', 'Simpson', 'Gauss-Legendre'};
colors = {'r', 'g', 'b', 'k'};

for func_idx = 1:5
    figure('Name', ['Function ' num2str(func_idx)]); % Creates a new figure for eac
    hold on;
    for choice = 1:4
        loglog(Ns, squeeze(errors(func_idx, :, choice)), '-o', 'Color', colors{cho
    end
    xlabel('N');
    ylabel('Error');
    title(['Log-Log plot of N vs Error for Function ', num2str(func_idx)]);

    % Ensuring that the axes are in log-log scale
    set(gca, 'XScale', 'log', 'YScale', 'log');

    legend('Location', 'southwest');
    grid on;
    hold off;
end

% 1. Midpoint rule with N cells
% 2. Trapezium rule with N cells
% 3. Simpson''s rule with 2N cells
% 4. Three-point Gauss-Legendre quadrature with N cells
function result = numerical_integration(a, b, f, N, choice)

    switch choice
        case 1
            result = midpoint_rule(f, a, b, N);
        case 2
            result = trapezium_rule(f, a, b, N);
        case 3
            result = simpsons_rule(f, a, b, N);
        case 4
            result = gauss_legendre(f, a, b, N);
        otherwise
            return;
    end

    % fprintf('The result of the integration is: %.5f\n', result);
end

function result = midpoint_rule(f, a, b, N)
    result = 0;
```

```
    for i = 0:(N-1)
        point = a+(i+0.5)*(b-a)/N;
        result = result + f(point);
    end

    result = result * (b-a)/N;
end


function result = trapezium_rule(f, a, b, N)
    result = 0;
    for i = 1:(N-1)
        point = a+i*(b-a)/N;
        result = result + 2*f(point);
    end
    result = result + f(a) + f(b);

    result = result * (b-a)/N/2;
end

function result = simpsons_rule(f, a, b, N)
    result = 0;
    for i = 1:N
        left = a+(i-1)*(b-a)/N;
        right = a+i*(b-a)/N;
        result = result + 1/3*(b-a)/N/2*(f(left)+4*f((left+right)/2)+f(right));
    end
end

function result = gauss_legendre(f, a, b, N)
    x = [-sqrt(3/5), 0, sqrt(3/5)];
    w = [5/9, 8/9, 5/9];

    result = 0;
    for i = 1:N
        left = a + (i-1)*(b-a) / N;
        right = a + i*(b-a)/N;
        for j = 1:3
            xi = 0.5*(left+right + (right-left) * x(j));
            result = result+w(j)*f(xi);
        end
    end
    result = 0.5 * (b-a)/N*result;
end
```

N	Midpoint	Trapezium	Simpson	Gauss-Legendre
2	1.78885438199983	1.70710678118655	1.76160518172874	1.76266240387583
4	1.77014250014533	1.74798058159319	1.76275519396128	1.76274697462844
16	1.76320768598367	1.7618262836833	1.76274721855022	1.76274717401768
32	1.76286227284615	1.76251698483349	1.76274717684193	1.76274717403875

Table 1: Question 1a

N	Midpoint	Trapezium	Simpson	Gauss-Legendre
2	3.14159265358979	7.06745147303987e-32	2.0943951023932	1.60606730241802
4	1.5707963267949	1.5707963267949	1.5707963267949	1.5707963267949
16	1.5707963267949	1.5707963267949	1.5707963267949	1.5707963267949
32	1.5707963267949	1.5707963267949	1.5707963267949	1.5707963267949

Table 2: Question 1b

N	Midpoint	Trapezium	Simpson	Gauss-Legendre
2	0.419455176774456	0.448425131496025	0.429111828348312	0.428525592513674
4	0.426049167558007	0.43394015413524	0.428679496417085	0.428562331914328
16	0.428391866554757	0.428943359698886	0.4285756976028	0.428571070406974
32	0.428524607238247	0.428667613126822	0.428572275867772	0.428571357502592

Table 3: Question 1c

N	Midpoint	Trapezium	Simpson	Gauss-Legendre
2	1.93841476853743	1.62996052494744	1.83559668734077	1.89373800719319
4	1.91040464923353	1.78418764674243	1.86833231506983	1.89141202322811
16	1.89332086475362	1.8728210349583	1.88648758815518	1.89012260552418
32	1.89126652809474	1.88307094985596	1.88853466868182	1.8899772279319

Table 4: Question 1d

N	Midpoint	Trapezium	Simpson	Gauss-Legendre
2	0.856885021909063	0	0	0.881394330687781
4	0.870845677383917	0	0	0.883847456341008
16	0.882289474604227	0	0	0.885658742625317
32	0.884274758802113	0	0	0.88595040876536

Table 5: Question 1e

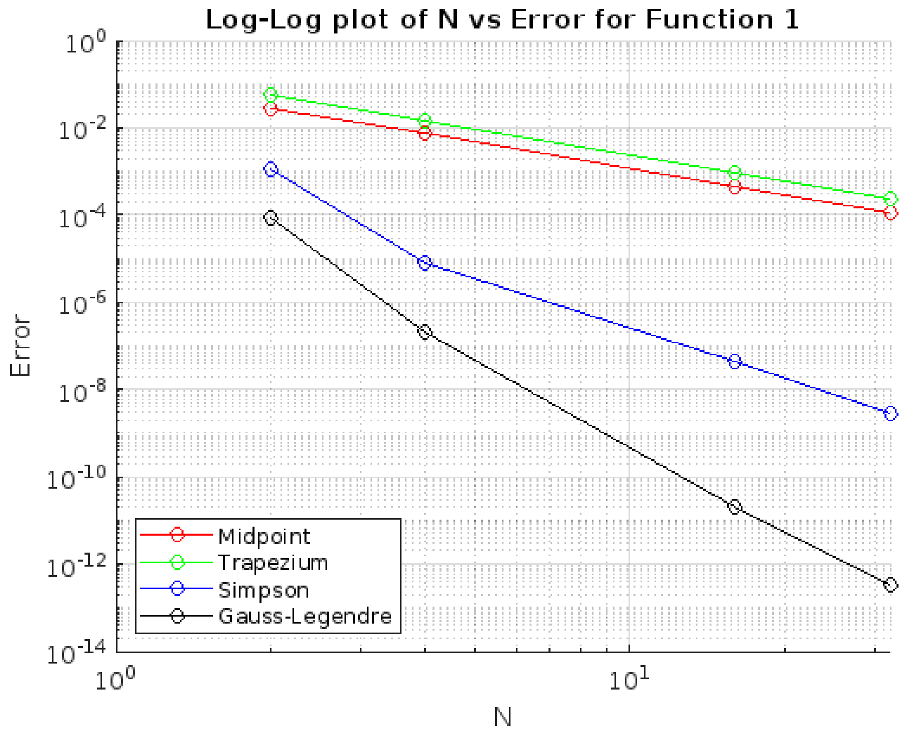

Figure 1: Error of Question 1a.

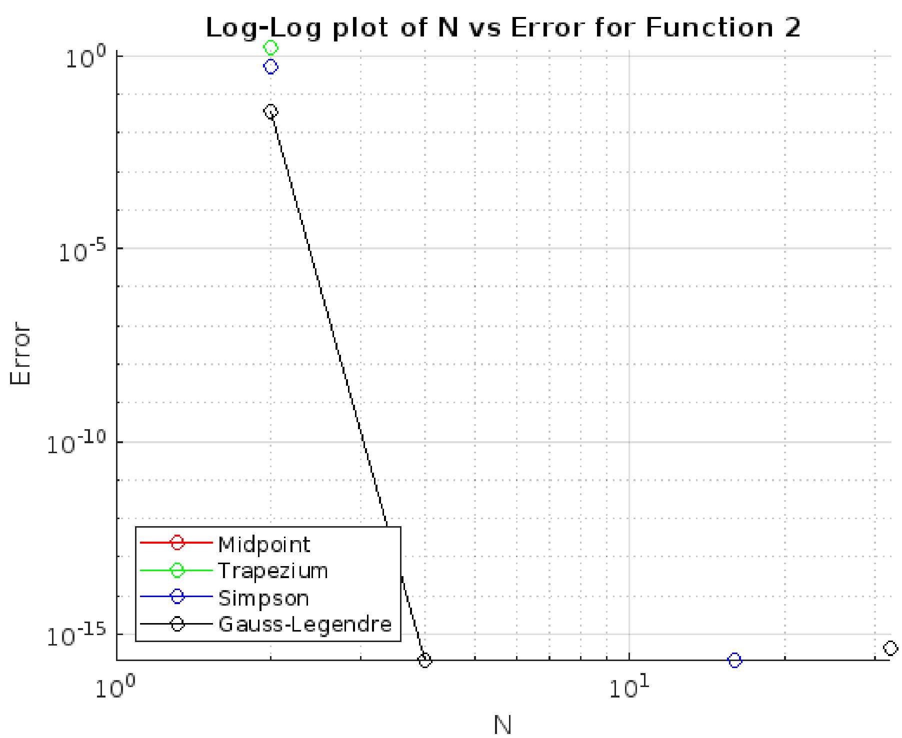

Figure 2: Error of Question 1b. Note that due to the fact that the error goes to zero very quickly for the different methods by chance means this plot doesn't look like the others.

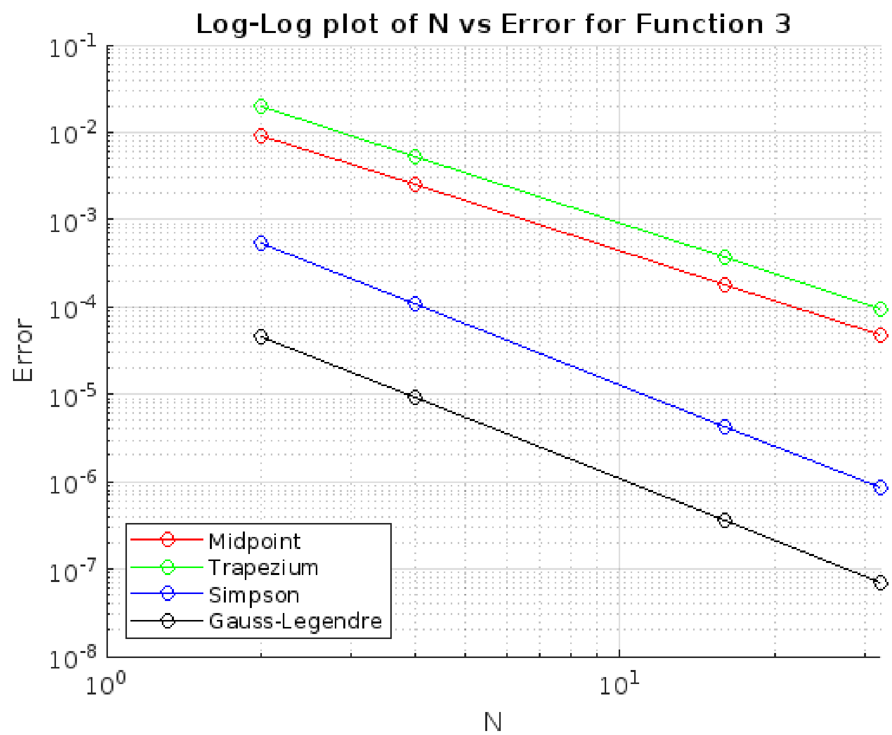

Figure 3: Error of Question 1c.

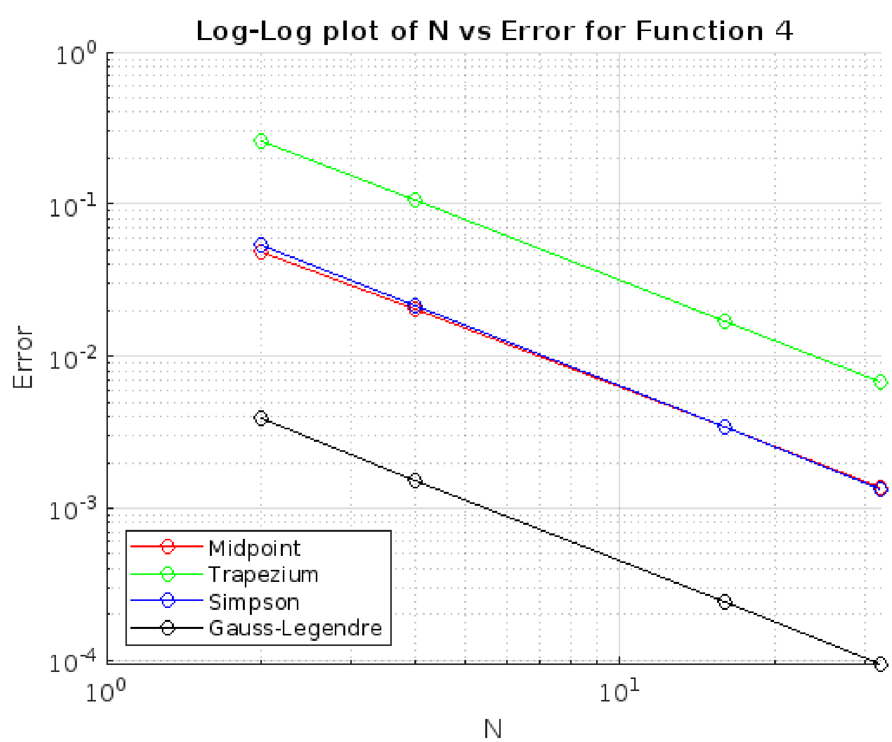

Figure 4: Error of Question 1d.

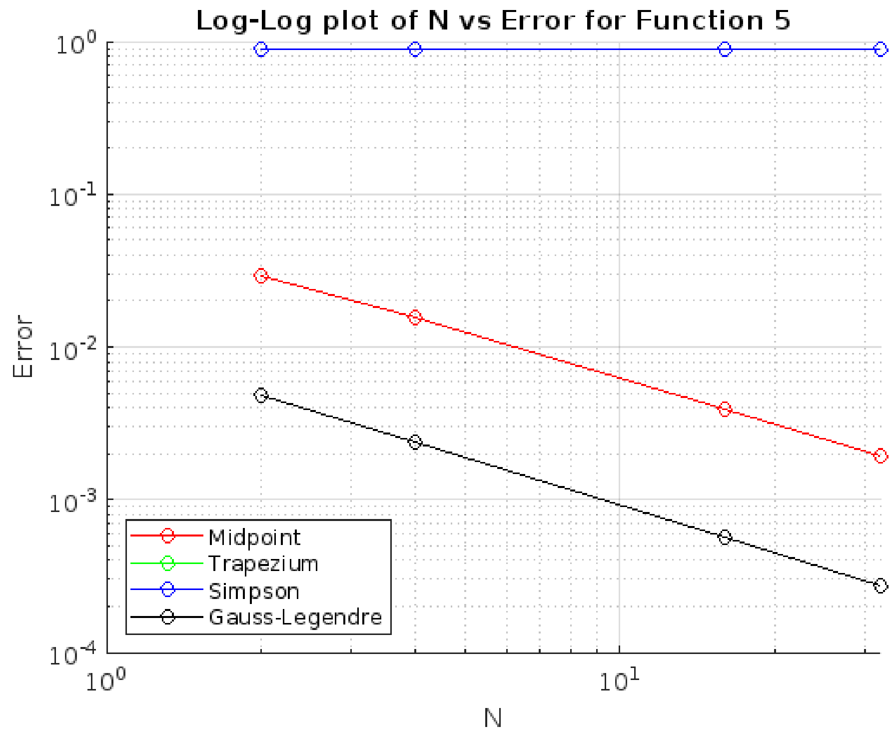

Figure 5: Error of Question 1e. Ignore the error for trapezium/Simpson, they were set to 0 in the code so had a constant error.

Question 2. Using the given asymptotic expansion for the Trapezium rule error, we get

$$I(0) - I(h_{s+1}) = I(0) - I(\frac{1}{2}h_s) = \sum_{i=1}^{\infty} \frac{1}{2^{2i}} c_i h_s^{2i}$$

$$\implies 4I(\frac{1}{2}h_s) = 4I(0) - \sum_{i=1}^{\infty} \frac{1}{2^{2i}} c_i h_s^{2i}$$

$$\implies I(0) = \frac{4I(\frac{h_2}{2}) - I(h_2)}{3} + \sum_{i=2}^{\infty} \frac{2^{2(i-1)} - 1}{3 \cdot 2^{2(i-1)}} c_i h_s^{2i}.$$

Define $a_s^{(1)} = I(h_s)$ and $c_i^{(2)}$ as in the question. Then simply rearranging the above formula algebraically, we get

$$I(0) - \left(I(\frac{h_2}{2}) + \frac{I(\frac{h_2}{2}) - I(h_s)}{3} \right) = I(0) - a_{s+1}^{(1)} - \frac{a_{s+1}^{(1)} - a_s^{(1)}}{3} = I(0) - a_s^2 = \sum_{i=2}^{\infty} c_i^{(2)} h_s^{2i}$$

as required. To eliminate the $O(h^4)$ term we can again rearrange, this time leaving off some algebra and leaving $c_i^{(3)}$ to be defined in the next part:

$$I(0) - a_{s+1}^2 + \frac{a_{s+1}^2 - a_s^{m-1}}{15} = \sum_{i=3}^{\infty} c_i^{(3)} h_s^{2i}.$$

To find the general recursion formula, we follow a similar process to what we just did. We've already shown the result $m = 2$, using recursion we just need to prove that the result holds for m given that it holds for $m-1$. Assume that for some m,

$$I(0) - a_s^{(m-1)} = \sum_{i=m-1}^{\infty} c_i^{(m-1)} h_s^{2i}$$

and

$$I(0) - a_{s+1}^{(m-1)} = \sum_{i=m-1}^{\infty} \frac{1}{2^{2i}} c_i^{(m-1)} h_s^{2i}.$$

Subtracting these, we get

$$I(0) - \frac{(4^{m-1})a_{s+1}^{(m-1)} - a_s^{(m-1)}}{4^{m-1}-1} = \sum_{i=m}^{\infty} \left((\frac{1}{4^{i-1}} - 1)\frac{1}{4^{m-1}-1} c_i^{(m-1)} \right) h_s^{2i}.$$

Define $c_i^{(m)} = \left((\frac{1}{4^{i-1}} - 1)\frac{1}{4^{m-1}-1} c_i^{(m-1)} \right)$. Then the previous equation is equivalent to

$$I(0) - a_{s+1}^{(m-1)} - \frac{a_{s+1}^{(m-1)} - a_s^{(m-1)}}{4^{m-1}-1} = \sum_{i=m}^{\infty} c_i^m h_s^{2i}.$$

Thus clearly we have that $a_s^{(m)} = a_{s+1}^{(m-1)} + \frac{a_{s+1}^{(m-1)} - a_s^{(m-1)}}{4^{m-1}-1}$ as required.

Question 3. Subtracting the first term of the Taylor series is already done for us in the question, so we just need to subtract the three term. $\cos x = 1 - \frac{x^2}{2} + \ldots$, so we want to evaluate

$$\begin{aligned} I &= \int_0^{\pi/2} x^{-\frac{1}{2}} dx - \int_0^{\pi/2} \frac{1}{2} x^{\frac{3}{2}} dx + \int_0^{\pi/2} x^{-\frac{1}{2}} \left(\cos x - 1 + \frac{x^2}{2} \right) dx \\ &= (2\pi)^{\frac{1}{2}} - \frac{\pi^{5/2}}{20\sqrt{2}} + \int_0^{\pi/2} x^{-\frac{1}{2}} \left(\cos x - 1 + \frac{x^2}{2} \right) dx. \end{aligned}$$

Plugging in each of the methods previously coded for question 1, we get table 6. This is the code used to generate the table:

```
f1 = @(x)x^(-0.5)*cos(x);
f2 = @(x)x^(-0.5)*(cos(x)-1);
f3 = @(x)x^(-0.5)*(cos(x)-1+x^2/2);

c1 = (2*pi)^0.5
c2 = (2*pi)^0.5-pi^(5/2)/20/2^0.5

N1 = 2^(4);
N2 = 2^(6);
a = eps^0.5;
b = pi/2;

T = zeros(8,2);
```

```
T(1,1) = midpoint_rule(f1,a,b,N1)
T(1,2) = midpoint_rule(f1,a,b,N2)

T(2,1) = gauss_legendre(f1,a,b,N1)
T(2,2) = gauss_legendre(f1,a,b,N2)

T(3,1) = midpoint_rule(f2,a,b,N1)+c1
T(3,2) = midpoint_rule(f2,a,b,N2)+c1

T(4,1) = gauss_legendre(f2,a,b,N1)+c1
T(4,2) = gauss_legendre(f2,a,b,N2)+c1

T(5,1) = trapezium_rule(f2,a,b,N1)+c1
T(5,2) = trapezium_rule(f2,a,b,N2)+c1

T(6,1) = midpoint_rule(f3,a,b,N1)+c2
T(6,2) = midpoint_rule(f3,a,b,N2)+c2

T(7,1) = gauss_legendre(f3,a,b,N1)+c2
T(7,2) = gauss_legendre(f3,a,b,N2)+c2

T(8,1) = trapezium_rule(f3,a,b,N1)+c2
T(8,2) = trapezium_rule(f3,a,b,N2)+c2


function result = midpoint_rule(f, a, b, N)
    result = 0;
    for i = 0:(N-1)
        point = a+(i+0.5)*(b-a)/N;
        result = result + f(point);
    end

    result = result * (b-a)/N;
end


function result = trapezium_rule(f, a, b, N)
    result = 0;
    for i = 1:(N-1)
        point = a+i*(b-a)/N;
        result = result + 2*f(point);
    end
    result = result + f(a) + f(b);

    result = result * (b-a)/N/2;
end

function result = simpsons_rule(f, a, b, N)
```

Integration Rule	$h = \left(\frac{1}{2}\right)^2$	$h = \left(\frac{1}{2}\right)^6$
Direct Midpoint	1.765666283681793	1.860155868481409
Direct 3 pt Gauss	1.876838968734573	1.915870385659986
Subtract 1 term Midpoint	1.955096450266878	1.954915723357975
Subtract 1 term 3 pt Gauss	1.954903132540786	1.954902857457019
Subtract 1 term Trapezium	1.954504413847847	1.954876746976996
Subtract 3 terms Midpoint	1.954743822379750	1.954892907353329
Subtract 3 terms 3 pt Gauss	1.954902848557130	1.954902848582609
Subtract 3 terms Trapezium	1.955220931501738	1.954922731169139

Table 6: Question 3.

```
    result = 0;
    for i = 1:N
        left = a+(i-1)*(b-a)/N;
        right = a+i*(b-a)/N;
        result = result + 1/3*(b-a)/N/2*(f(left)+4*f((left+right)/2)+f(right));
    end
end

function result = gauss_legendre(f, a, b, N)
    x = [-sqrt(3/5), 0, sqrt(3/5)];
    w = [5/9, 8/9, 5/9];

    result = 0;
    for i = 1:N
        left = a + (i-1)*(b-a) / N;
        right = a + i*(b-a)/N;
        for j = 1:3
            xi = 0.5*(left+right + (right-left) * x(j));
            result = result+w(j)*f(xi);
        end
    end
    result = 0.5 * (b-a)/N*result;
end
```

Question 4.