DOI: https://doi.org/10.36910/6775-2524-0560-2022-46-11
УДК 004.415.3
Пех Петро Антонович, к.т.н., доцент
https://orcid.org/0000-0002-6327-3319
Хільчишин Андрій Віталійович, студент
Луцький національний технічний університет

ПОРІВНЯЛЬНИЙ АНАЛІЗ РІЗНИХ КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ ЧИСЕЛЬНОГО ІНТЕГРУВАННЯ ФУНКЦІЙ

Пех П. А., Хільчишин А. В. Порівняльний аналіз різних комп'ютерних технологій чисельного інтегрування функцій. В статті порівнюються різні технології чисельного інтегрування функцій: на базі розробленої авторами програми мовою Matlab, за допомогою електронних таблиць Excel та за стандарною функцією `quad()`.
Ключові слова: формула Сімпсона, підпрограми, Matlab, модуль, чисельне інтегрування

Пех П. А., Хильчишин А. В. Сравнительный анализ различных компьютерных технологий численного интегрирования функций. В статье сравниваются различные технологии численного интегрирования функций: на базе разработанной авторами программы на языке Matlab; с помощью электронных таблиц Excel и с помощью стандарной функции `quad()`
Ключевые слова: **ф**ормула Симпсона, подпрограммма, Matlab, модуль, численное интегрирование

Petro Pekh, Andriy Khilchyshyn. Comparative analysis of different computer technologies of numerical integration functions. The article compares different technologies of functions numerical integration : on the basis of the program developed by the authors in Matlab, using Excel spreadsheets and using the standard function quad ().
Keywords: Simpson's formula, subroutines, Matlab, module, numerical integration

Постановка задачі. Під час чисельного інтегрування функції за формулою Сімпсона проміжок інтегрування розбивається на парну кількість інтервалів n, і через кожні три сусідні точки, що ділять проміжок інтегрування, проводяться ланки параболи. Тоді величина означеного визначається за формулою [4]: $I = h\left(y_0 + y_n + 4\sum_{i=1}^{n} y_{2i} + 2\sum_{i=1}^{n} y_{2i-1}\right);\ n - \text{парне число}$. Існують багато різних інструментальних засобів, які автоматизують процес інтегрування за формулою Сімпсона[5]. У даній роботі акцент робиться на використанні мови Matlab [1, 2, 3] та електоронних таблиць Excel.

Метою нашого дослідження було розроблення програми засобами мови Matlab та електоронних таблиць Excel для чисельного інтегрування функцій на базі формули Сімпсона і порівняти результати їх роботи.

Новизна полягає у вирішенні задачі обчислення означених інтегралів різними комп'ютерними технологіями.

Основна частина. Насамперед розглянемо структуру розробленої нами мовою Matlab програми для чисельного визначення величини означеного інтеграла (одного із трьох зазначених у «шапці» програми) з наперед заданою точністю та призначення її окремих частин. Ця програма складається з головної функції та п'ятнадцяти підпрограм різного рівня.

Головна функція `Simpson_Formula_Function` (рис. 1) послідовно викликає до роботи п'ять підпрограм першого рівня, кожна з яких реалізує окрему частину основної задачі, а саме:
- введення вхідних даних;
- ітераційне обчислення послідовних наближень означеного інтеграла;
- ініціалізацію початкових значень робочих змінних;
- виведення на друк всіх послідовних наближень означе ного інтеграла;
- виведення на друк підсумкових результатів обчислення інтеграла різними методами.

Після запуску програми на виконання, вона викликає підпрограму першого рівня `inp_data()` (рис. 2), яка забезпечує ввід вхідних даних: прядкового номера v інтеграла з трьох зазначених у «шапці» програми; нижню xmin та верхню xmax межі інтегрування; кількість інтервалів розбиття n проміжку інтегрування:

```
Введіть номер інтеграла: 1
Введіть нижню межу інтегрування: 0
Введіть верхню межу інтегрування: 1
Введіть кількість інтервалів розбиття проміжку [xmin; xmax]: 10
```

Введені підпрограмою `inp_data()` дані повертаються у головну функцію програми за допомогою параметрів `[v,xmin,xmax,n,eps]`.

Далі з головної функції викликається підпрограма першого рівня `init_data()` (рис. 3), яка встановлює початкові значення робочих змінних програми і повертає їх у головну функцію через відповідні параметри. Серед цих змінних – попереднє Ip та наступне Ik значення інтеграла, номер ітерації k. На кожній k-ій ітерації поточні результати розрахунків інтеграла будуть накопичуватися у відповідних векторах: ks – змінна для зберігання сумарної кількості ітерацій; kk – вектор для зберігання значень номерів ітерацій; nk – вектор для зберігання значень кількості інтервалів розбиття; Ikk – вектор для зберігання значень інтеграла на k-ій ітерації.

```
function Simpson_Formula_Function
 % Обчислення значень трьох означених інтегралів від таких функцій:
 % f1(x)= 1./(x.^2+4).^(1/2); межі інтегрування a=0; b=1; n=10
 % f2(x)= (x.^3+1)./(x.^4-3*x.^3+3*x.^2-x); a=2.1; b=4.1; n=10
 % f3(x)= x./(1-x.^3).^(1/2); a=0; b=0.9; n=10
 % за формулою Сімпсона з точністю eps=0.001

 % Введення вхідних даних
 [v,xmin,xmax,n,eps]=inp_data();

 % Ініціалізація початкових значень робочих змінних
 [Ip,Ik,k,ks,kk,nk,Ikk]=init_data();

 % Ітераційне обчислення послідовних наближень означеного інтеграла
 [Ik,k,ks,kk,nk,Ikk]=iterat_integr_calc(v,xmin,xmax, ...
                               n,eps,Ip,Ik,k,ks,kk,nk,Ikk)
 % Виведення на друк всіх послідовних наближень означе ного інтеграла:
 print_all_iteration(Ip,Ik,k,ks,kk,nk,Ikk,n);

 % Підсумкові результати обчислення інтеграла різними методами:
 print_res_total(v,Ik,xmin,xmax);
 stop=input('\nРозвязок задачі завершено...\n');
end
```

Рис. 1 – Головна функція програми Simpson_Formula_Function

```
 % Введення вхідних даних
function [v,xmin,xmax,n,eps]=inp_data()
 v=input('Введіть номер інтеграла: ');
 xmin=input('Введіть нижню межу інтегрування: ');
 xmax=input('Введіть верхню межу інтегрування: ');
 n=input('Введіть кількість інтервалів розбиття проміжку [xmin; xmax]: ');
 eps=0.001; % Точність обчислення означеного інтеграла
end
```

Рис. 2 – Підпрограма введення вхідних даних

```
 % Ініціалізація початкових значень робочих змінних
function [Ip,Ik,k,ks,kk,nk,Ikk]=init_data()
   Ip=0; % Попереднє наближення значення інтеграла
   Ik=1; % Наступне наближення значення інтеграла
   k=0;  % Номер поточної ітерації
   ks=0; % Змінна для зберігання сумарної кількості ітерацій
   kk=[];  % Вектор для зберігання значень номерів ітерацій
   nk=[];  % Вектор для зберігання значень кількості інтервалів розбиття
   Ikk=[]; % Вектор для зберігання значень інтеграла на певній ітерації
end
```

Рис. 3 – Підпрограма ініціалізації початковими значеннями робочих змінних

Третьою за порядком з головної функції викликається підпрограма першого рівня `iterat_integr_calc(v,xmin,xmax,n,eps,Ip,Ik,k,ks,kk,nk,Ikk)` (рис. 4), яка за допомогою оператора `while abs(Ik-Ip)>eps` реалізує ітераційний процес обчислення означеного інтеграла шляхом багаторазового виклику підпрограми другого рівня `integr_k(v,xmin,xmax,n,eps,Ip,Ik,k,ks,kk,nk,Ikk)`. Виконання оператора `while` припиняється, як тільки буде досягнута задана точність обчислень. Крім того, після завершення

```
% Ітераційне обчислення послідовних наближень означеного інтеграла
function  [Ik,k,ks,kk,nk,Ikk]=iterat_integr_calc(v,xmin,xmax, ...
              n,eps,Ip,Ik,k,ks,kk,nk,Ikk)
 while abs(Ik-Ip)>eps
   % Обчислення k-го наближення означеного інтеграла
   [Ip,Ik,k,n,x,y,ks,kk,nk,Ikk]=integr_k(v,xmin,xmax,n,eps, ...
                                       Ip,Ik,k,ks,kk,nk,Ikk);
   if k==1 % Побудова гістограми підінтегральної функції
     plot_figure(v,x,y);
   end
 end
end
```

Рис. 4 – Підпрограма ітераційного обчислення послідовних наближень означеного інтеграла

першої ітерації за допомогою підпрограми другого рівня `plot_figure(v,x,y)` (рис. 11) на екран виводиться гістограма підінтегральної функції. Результати побудов для всіх трьох розглядуваних інтегралів наведені на рис. 15.

Підпрограма `integr_k(v,xmin,xmax,n,eps,Ip,Ik,k,ks,kk,nk,Ikk)` (рис.5) під час виконання ітераційного процесу у свою чергу послідовно викликає п’ять підпрограм третього рівня. Вони реалізують такі частини основного алгоритму.

- `cycle_preparation(k,xmin,xmax,n,Ik)` (рис. 6) – піготовка ітераційного циклу;
- `tab_func(v,n,xmin,dx,s1,s2)` (рис. 7) – табулювання функції та обчислення допоміжних сум;
- `integr_calc(s1,s2,dx,y,n,k)` (рис. 8) – обчислення інтеграла за формулою Сімпсона та виведення результату;
- `accumulation(ks,kk,nk,Ikk,k,n,Ik)` (рис. 9) – накопичення даних для їх подальшого аналізу;
- `correction_data(n)` (рис. 10) – коррекція значення кількості інтервалів n.

Зокрема, результати табулювання підінтегральної функції та обчислення значення означеного інтеграла на першій ітерації виводяться у такому вигляді:

```
Результати табулювання підінтегральної функції:
x= 0.00000 y=  0.50000
x= 0.10000 y=  0.49938
x= 0.20000 y=  0.49752
x= 0.30000 y=  0.49447
x= 0.40000 y=  0.49029
x= 0.50000 y=  0.48507
x= 0.60000 y=  0.47891
x= 0.70000 y=  0.47193
x= 0.80000 y=  0.46424
x= 0.90000 y=  0.45596
x= 1.00000 y=  0.44721
s=  0.94721 s1=  1.93096 s2=  2.40681
Наближене значення інтеграла на 1-ій ітерації:
k=    1 n=       10   Ik=  0.44949
```

Як тільки точність обчислень буде досягнута, головна функція викличе підпрограму другого рівня `print_all_iteration(Ip,Ik,k,ks,kk,nk,Ikk,n)` (рис. 12), яка виведе на друк всі послідовні наближення означеного інтеграла у такому вигляді:

```
Послідовні наближення значення інтеграла:
k=    1 n=       10   Ik=  0.44949
```

```
k=    2 n=      100   Ik=   0.47805
k=    3 n=     1000   Ik=   0.48090
k=    4 n=    10000   Ik=   0.48118
```

```
% Обчислення k-го наближення означеного інтеграла
function    [Ip,Ik,k,n,x,y,ks,kk,nk,Ikk]=integr_k(v,xmin,xmax,n,eps, ...
                                                  Ip,Ik,k,ks,kk,nk,Ikk);
 % Пітотовка ітераційного циклу
 [k,dx,s1,s2,x,y,Ip]=cycle_preparation(k,xmin,xmax,n,Ik);

 % Табулювання функції та обчислення допоміжних сум
 [x,y,s1,s2]=tab_func(v,n,xmin,dx,s1,s2);

 % Обчислення інтеграла за формулою Сімпсона та виведення результату
 [s1,s,Ik]=integr_calc(s1,s2,dx,y,n,k);

 % Накопичення даних для подальшого аналізу
 [ks,kk,nk,Ikk,k,n,Ik]=accumulation(ks,kk,nk,Ikk,k,n,Ik);

 % Коррекція значення кількості інтервалів n
 [n]=correction_data(n);
end
```

Рис. 5 – Підпрограма обчислення k-го наближення означеного інтеграла

```
% Пітотовка ітераційного циклу
function [k,dx,s1,s2,x,y,Ip]=cycle_preparation(k,xmin,xmax,n,Ik)
 k=k+1; % Поточне значення номера ітерації
 if k==1
  Ip=0; % Попереднє наближення інтеграла
 else
  Ip=Ik; % Запамятовуємо наступне наближення інтеграла як попереднє
 end
 dx=(xmax-xmin)/n; % Крок інтегрування
 s1=0; % Обнулення допоміжної суми парних елементів
 s2=0; % Обнулення допоміжної суми непарних елементів
 fprintf('\nРезультати табулювання підінтегральної функції:\n');
 x=[]; % Очищення вектора абсцис функції x
 y=[]; % Очищення вектора ординат функції y
end
```

Рис. 6 – Підпрограма підготовки ітераційного циклу

```
% Табулювання функції та обчислення допоміжних сум
function [x,y,s1,s2]=tab_func(v,n,xmin,dx,s1,s2)
 for i=1:n+1; % Формування значень елементів векторів x та y
  x(i)=xmin+dx*(i-1); % Формування вектора абсцис точок розбиття
  if v==1
    y(i)=f1(x(i)); % Формування вектора ординат в точках розбиття
  end
  if v==2
   y(i)=f2(x(i)); % Формування вектора ординат в точках розбиття
  end
  if v==3
   y(i)=f3(x(i)); % Формування вектора ординат в точках розбиття
  end
  if rem(i,2)>0
   s1=s1+y(i);
  end
```

Рис. 7 – Підпрограма табулювання підінтегральної функції та обчислення допоміжних сум (початок)

```
    if  rem(i,2)==0
     s2=s2+y(i);
    end
    fprintf('\nx=%8.5f y= %8.5f',x(i), y(i)); % Друк абсцис та ординат
  end
end
```

Рис. 7 – Підпрограма табулювання підінтегральної функції та обчислення допоміжних сум (кінець)

```
% Обчислення інтеграла за формулою Сімпсона та виведення результату
function [s1,s,Ik ]=integr_calc(s1,s2,dx,y,n,k)
 s1=s1-y(1)-y(n+1);
 s=y(1)+y(n+1)+4*s1+2*s2;
 Ik=s*dx./3; % Інтеграл за формулою Сімпсона
 fprintf('\ns= %8.5f s1= %8.5f s2= %8.5f',y(1)+y(n+1), s1, s2)
 fprintf('\n\nНаближене значення інтеграла на %2d-ій ітерації:', k);
 fprintf('\nk=%4d n=%8d  Ik= %8.5f',k, n, Ik);
end
```

Рис. 8 – Підпрограма обчислення інтеграла за формулою Сімсона та виведення результату

```
% Накопичення даних для подальшого аналізу
function [ks,kk,nk,Ikk,k,n,Ik]=accumulation(ks,kk,nk,Ikk,k,n,Ik)
 ks=ks+1; % Змінна для зберігання сумарної кількості ітерацій
 kk=[kk k]; % Вектор для зберігання значень номерів ітерацій
 nk=[nk n]; % Вектор для зберігання значень кількості інтервалів розбиття
 Ikk=[Ikk Ik];% Вектор для зберігання значень інтеграла на певній ітерації
end
```

Рис. 9 – Підпрограма накопичення даних для їх подальшого аналізу

```
% Коррекція значення кількості інтервалів n
function [n]=correction_data(n)
 n=10*n; % Збільшення у 10 разів кількості інтервалів
 stop=input('\nНатисніть яку-небудь кнопку, щоб продовжити...');
end
```

Рис. 10 – Підпрограма корекції значення кількості інтервалів розбиття проміжку інтегрування

```
% Побудова гістограми підінтегральної функції
function plot_figure(v,x,y)
 % Вибір графічного вікна для побудови графіка функції
 figure(v);
 area(x,y), grid on; % Побудова графіка підінтегральної функції
end
```

Рис. 11 – Підпрограма побудови гістограми підінтегральної функції

```
% Виведення на друк всіх послідовних наближень означеного інтеграла:
function  print_all_iteration(Ip,Ik,k,ks,kk,nk,Ikk,n)
 fprintf('\n\nПослідовні наближення значення інтеграла:');
 for i=1:ks
  fprintf('\nk=%4d n=%8d  Ik= %8.5f',kk(i), nk(i), Ikk(i));
 end
 stop=input('\n\nНатисніть яку-небудь кнопку, щоб продовжити...');
end
```

Рис. 12 – Підпрограма виведення на друк всіх послідовних значень означеного інтеграла

Останньою головна функція викличе підпрограму другого рівня `print_res_total(v,Ik,xmin,xmax)` (рис. 12), яка виведе на друк підсумкові результати обчислення інтеграла різними методами:

```
Підсумкові результати обчислення  1-го інтеграла:
Значення інтеграла за формулою Сімпсона Ik=  0.48118
Значення інтеграла, обчислене за первісною It=  0.48121
Значення інтеграла за стандартною функцією Is=  0.48121
```

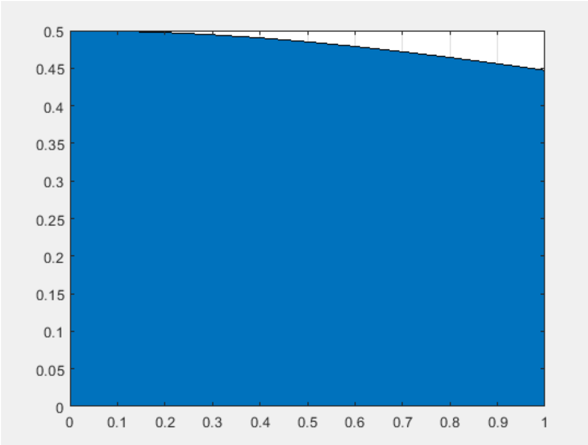
```
% Підсумкові результати обчислення інтеграла  різними методами::
function print_res_total(v,Ik,xmin,xmax)
 fprintf('\n\nПідсумкові результати обчислення %2d-го інтеграла:',v);
 if v==1
   fp=inline('log(x+(x.^2+4).^(1/2))'); % Первісна функція від  f1(x)
   It=fp(xmax)-fp(xmin); %Значення інтеграла, обчислене за первісною
   Is=quad('1./(x.^2+4).^(1/2)',xmin, xmax); % Використано функцію Matlab
 end
 if v==2
  fp=inline('-x./(x-1).^2+log(((x-1).^2)./abs(x))'); % Первісна від  f2(x)
  It=fp(xmax)-fp(xmin); %Значення інтеграла, обчислене за первісною
  Is=quad('(x.^3+1)./(x.^4-3*x.^3+3*x.^2-x)', xmin, xmax); % функція Matlab
 end
 if v==3
   Is=quad('x./(1-x.^3).^(1/2)',0,0.9); % Використано функцію Matlab
 end
 fprintf('\n\nЗначення інтеграла за формулою Сімпсона Ik= %8.5f', Ik);
 if (v==3)
  fprintf('\nЗначення інтеграла, обчислене за первісною не існує');
 else
  fprintf('\nЗначення інтеграла, обчислене за первісною It= %8.5f', It);
 end
 fprintf('\nЗначення інтеграла за стандартною функцією Is= %8.5f', Is);
 fprintf('\n');
 stop=input('\n\nНатисніть яку-небудь кнопку, щоб продовжити...');
end
```

Рис. 13 – Підпрограма виведення на друк результатів обчислення інтеграла різними методами

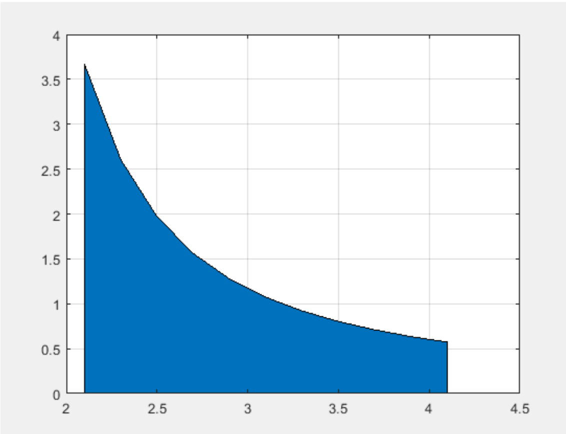
Зазначимо, що кожного разу, коли є потреба обчислити значення підінтегральної функції, викликаються підпрограми першого рівня f1(x), f2(x) чи f3(x) (рис. 14).

```
% Оголошення підпрограми-функції f1
function y1=f1(x);
 y1=1./(x.^2+4).^(1/2);
end

% Оголошення підпрограми-функції f2
function y2=f2(x);
 y2= (x.^3+1)./(x.^4-3*x.^3+3*x.^2-x);
end

% Оголошення підпрограми-функції f3
function y3=f3(x);
 y3= x./(1-x.^3).^(1/2);
end
```

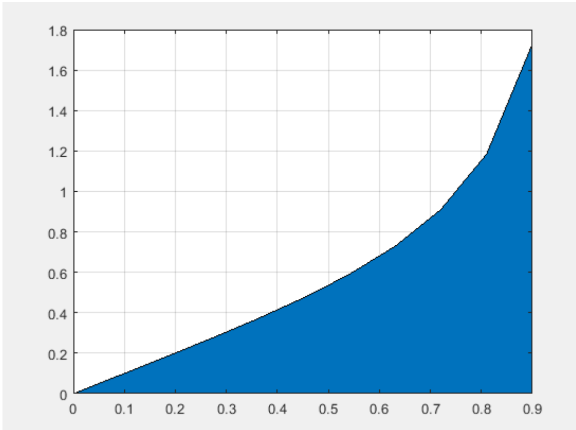
Рис. 14 – Підпрограми оголошень трьох підінтегральних функцій



a



b



c

Рис. 15 – Гістограми підінтегральних функцій:
a – $f_1(x) = \frac{1}{\sqrt{x^2+4}}$; b – $f_2(x) = \frac{x^3+1}{x^4-3x^3+3x^2-x}$; c –$f_2(x) = \frac{x}{\sqrt{1-x^3}}$.

Розглянемо тепер фрагмент електронної таблиці (рис. 16), розроблену нами для обчислення інтеграла $I_1 = \int_0^1 \frac{dx}{\sqrt{x^2+4}}$ за формулою Сімпсона. Формула Сімпсона відрізняється, по-перше, тим, що відрізок інтегрування ділиться на парне число частин, по-друге, у ній використовуються три допоміжні суми. Перша допоміжна сума (клітина C15) – це сума значень функції на початку (клітина C4) та кінці (клітина C14) проміжку інтегрування. Друга допоміжна сума (клітина D15) – це сума значень функції з непарними індексами (клітини D5 – D13); далі ця сума множиться на 4. Третя допоміжна сума (клітина E15) – це сума значень функції з парними індексами (клітини E5 – E13); далі ця сума множиться на 2. Отримані у таких спосіб величини додаються і множаться на крок інтегрування 0,1 та діляться на 3. Це й буде значення означеного інтеграла (клітина C16). Далі ця електронна таблиця конструюється аналогічно з кроками 0,01; 0,001; 0,0001 і т.д., поки два послідовних значення означеного інтеграла не співпадуть у межах заданої точності.

	A	B	C	D	E
1					
2	Формула Сімпсона				
3	h=0,1	x	x^2+4)^(1/2)	Непарні	Парні
4	1	0	0,50000		
5	2	0,1	0,49938	0,00000	0,49938
6	3	0,2	0,49752	0,49752	0,00000
7	4	0,3	0,49447	0,00000	0,49447
8	5	0,4	0,49029	0,49029	0,00000
9	6	0,5	0,48507	0,00000	0,48507
10	7	0,6	0,47891	0,47891	0,00000
11	8	0,7	0,47193	0,00000	0,47193
12	9	0,8	0,46424	0,46424	0,00000
13	10	0,9	0,45596	0,00000	0,45596
14	11	1	0,44721		
15			0,94721	1,93096	2,40681
16			0,44949		

Рис. 16 – Електронна таблиця для обчислення інтеграла $I_1 = \int_0^1 \frac{dx}{\sqrt{x^2+4}}$ за формулою Сімпсона

У табл. 1 наводяться підсумкові результати обчислення трьох означених інтегралів різними методами. Порівнюючи ці результати, можна стверджувати, що всі розглядувані методи обчислення означених інтегралів забезпечують необхідну точність.

Табл. 1 – Підсумкові результати обчислення означених інтегралів різними методами

Метод обчислення інтеграла	$\int_0^1 \frac{dx}{\sqrt{x^2+4}}$	$\int_{2.1}^{4.1} \frac{x^3+1}{x^4-3x^3+3x^2-x}dx$	$\int_0^{0.9} \frac{xdx}{\sqrt{1-x^3}}$
За розробленою програмою	0.48118	2.71200	0.50348
За первісною функцією	0.48121	2.71203	не існує
За стандартною функцією Matlab	0.48121	2.71203	0.50353

Висновки. 1. У даній статті розроблені програма засобами Matlab та електронні таблиці засобами Excel для розрахунку трьох означених інтегралів з наперед заданою точністю. Крім того, величини інтегралів визначалися також за допомогою стандарної функції quad(). Розрахунки показали, що всі три методи забезпечують задану точність обчислень.

2. Модульний принцип побудови програми дозволяє у подальшому розширити кількість підінтегральних функцій, що є важливим як з теоретичної, так і практичної точки зору.

3. Вважаємо, що розроблена програма може бути використана як у навчальному процесі, так і в інженерній практиці під час обчислення означених інтегралів.

References.

1. P.Pekh, O.Kuzmych, N.Zdolbitska, N.Bahniuk, I.Pasternak. Generators of Some Kinds Random Erlang Numbers and Estimation of Their Complexity // IEEExplore Digital Library (Scopus), Published in: 2020 10th International Conference on Advanced Computer Information Technologies (ACIT). DOI: 10.1109/ACIT49673.2020.9208831, ISBN: 978-1-7281-6760-2. https://ieeexplore.ieee.org/document/9208831

Список бібліографічного опису.

1. Пех П.А. Методи обчислень та моделювання: Конспект лекцій для здобувачів першого (бакалаврського) рівня освітньо-професійної програми «Комп'ютерна інженерія» галузі знань 12 «Інформаційні технології» спеціальності 123 «Комп'ютерна інженерія» денної та заочної форм навчання – Луцьк : Луцький НТУ, 2020. – 162 с. Формат А4
2. Дьяконов В.П. Matlab і Simulink для радіоінженерів. – М.: «ДМК-Пресс», 2011. -976 с.
3. Бахвалов, М.С. Чисельні методи: Підручник / М.С. Бахвалов, М.П. Жидков, Г.М. Кобельков. - М.: Біном. ЛЗ, 2011. - 636 с.
4. Björck, Åke . Numerical methods for least squares problems. Philadelphia: SIAM, 1996. ISBN 0-89871-360-9.Greene, William H. Econometric analysis (5th ed.). New Jersey: Prentice Hall, 2002