DOI: https://doi.org/10.36910/6775-2524-0560-2022-46-10
УДК 004.415.3

Пех Петро Антонович, к.т.н., доцент
https://orcid.org/0000-0002-6327-3319
Христинець Наталія Анатоліївна, к.т.н., ст.викладач
https://orcid.org/0000-0002-4836-7632
Дяченко Роман Олегович, студент
Луцький національний технічний університет

ПРОГРАМНА ОПТИМІЗАЦІЯ ВИБОРУ ТИПУ ФУНКЦІЇ ДЛЯ АПРОКСИМАЦІЇ ЕКСПЕРИМЕНТАЛЬНИХ ДАНИХ

Пех П. А., Христинець Н.А., Дяченко Р. О. Програмна оптимізація вибору типу функції для апроксимаціїї експерментальних даних. В статті запропоновано програма мовою Matlab для оптимального вибору апроксимуючої функції в процесі оброблення експерментальних даних. Програма складається з головного та дев’яти допоміжних модулів, кожен з яких реалізує окрему частину загального алгоритму розв’язку задачі вибору та розрахунку параметрів апроксимуючої функції.

Ключові слова: Метод найменших квадратів, Matlab, апроксимація, підпрограма- функція, модуль

Пех П. А., Христинец Н.А., Дьяченко Р. О. Программная оптимизация выбора типа функции для аппроксимации экспериментальных данных. В статье предложена программа на языке Matlab для оптимального выбора аппроксимирующей функции в процессе обработки экспериментальных данных. Программа состоит из главного и девяти вспомогательных модулей, каждый из которых реализует отдельную часть общего алгоритма решения задачи выбора и расчета параметров аппроксимирующей функции.

Ключевые слова: Метод найменших квадратів, Matlab, апроксимація, підпрограма- функція, модуль

Pekh Petro, Hrystynets Natalia, Diachenko Roman. Software optimization of function type selection for approximation of experimental data. The article proposes a program by means Matlab language for the optimal choice of the approximating function in the process of processing experimental data. The program consists of the main and nine auxiliary modules, each of which implements a separate part of the general algorithm for solving the problem of selection and calculation of the approximating function parameters.

Keywords: least squares method, Matlab, approximation, subroutine-function, module

Постановка задачі. Нехай в результаті проведення *n* дослідів експерименту отримані дані, які чисельно відображують залежність деякої вихідної величини (функції) від іншої вхідної величини (аргумента), значення якої під час проведення експерименту регулюються самим дослідником. Позначимо значення аргументу в i -ому досліді експерименту x_i , а значення функції y_i . Відомо, що експериментальні дані можуть бути апроксимовані різними типами теоретичних функцій: лінійною $y = ax + b,$ параболічною $y = ax^2 + bx + c,$ показниковою $y = ae^{bx},$ степеневою $y = ax^b$ та іншими [4,5]. Задача полягає у розробленні та тестуванні на різних наборах експериментальних даних такої програми, з допомогою якої можна було б оптимізувати процес вибору типу апроксимуючої функції з певної множини функцій та розрахувати її параметри.

Мета дослідження полягає у розробленні, відлагодженні та тестуванні програми для автоматизації процесу вибору типу апроксимуючої функції з певної множини функцій та розрахунку її параметрів.

Новизна дослідження полягає у вирішенні задачі вибору типу апроксимуючої функції програмним шляхом засобами Matlab [2,3].

Основна частина. Розроблена нами засобами Matlab програма складається з головного модуля `opt_model` (рис.1) та допоміжних модулів (рис.2 – рис.7). Призначення головного модуля `opt_model` – формування вихідних наборів експериментальних даних та виклик підлеглих модулів, які крок за кроком реалізують алгоритм розв’язку задачі. Програма розроблена за технологією функціонального програмування [1,2,3], а тому зручна для аналізу, налаштування та удосконалення.

Модуль `[v x y]=choose(x1,y1,x2,y2,x3,y3,x4,y4)` (рис. 2) передає у головний модуль значення певного набору експериментальних даних робочим векторам `x` та `y`, а також номер цього набору `v`.

Модуль `pre_graf(x,y)` (рис.3) наносить точки, координати яких задаються даними, що містяться у векторах x y, з метою їх візуального аналізу та вибору типу апроксимуючої функції.

Модуль `koef_kor(x,y,n)` (рис.4) розраховує коефіцієнт кореляції між даними, що містяться у двох векторах x та y.

Модуль `[ozn]=approximation(x,y,n,ozn)` (рис.5) викликає чотири допоміжні модулі, кожен з яких апроксимує дані, що містяться у двох векторах `x` та `y` за допомогою лінійної, параболічної, показникової та степеневої функцій. У головний модуль передаються середнє значення суми квадратів відхилень експериментальних та теоретичних значень функції.

```
function opt_model
% В результаті проведення експерименту отримано ряд значень функції,
% які відповідають встановленим у кожному досліді значенням аргумента.
% Ці дані можна апроксимувати різними математичними функціями:
% 1 - лінійною; 2 - параболічною; 3 - експоненціальною; 4 - степеневою.
% Необхідно вибрати зі запропонованих варіантів функцію,
% яка найкращим чином описує експериментальні дані,
% та визначити її параметри.

% Формування чотирьох наборів експериментальних даних
% у вигляді значень елементів масивів аргумента x та функції y:
x1=[2.5 2.7 2.9 3.1 3.3 3.5 3.7 3.9 4.1 4.3];
y1=[8.0 8.4 8.8 9.2 9.4 10.1 10.5 10.8 11.2 11.5];
x2=1.2:0.2:4.0;
y2=[9.2 8.5 5.9 5.1 4.4 3.1 2.5 2.3 3.1 3.2 3.0 3.8 5.0 6.4 7.0];
x3=0.1:9.1;
y3=[82.5 66.4 53.5 43.1 34.7 28.0 22. 18.4 14.2 12.4];
x4=5.0:2.0:33;
y4=[26.5 64.4 124.8 211.7 328.7 479.2 666.4 893.2 1162.6 1477.4 1840.2 ...
    2253.8 2720.5 3162.9 3980.5];

% Вибір номера набору експериментальних даних для подальшого аналізу
[v x y]=choose(x1,y1,x2,y2,x3,y3,x4,y4);
n=length(x); % Визначення довжини вектора x
ozn=[];% Формування вектора, що зберігає значення сум квадратів відхилень
% Нанесення експериментальних точок на координатну сітку
% для подальшого аналізу і вибору виду апроксимуючої функції
pre_graf(x,y);
% Розрахунок коефіцієнта кореляції за експериментальними даними
koef_kor(x,y,n);
% Апроксимація експериментальних даних
% різними типами функцій і формування вектора оцінок ozn
[ozn]=approximation(x,y,n,ozn);
% Порівняльний аналіз результатів апроксимації експериментальних даних
% різними типами функцій за допомогою вектра оцінок ozn
est_eff(ozn,v);
end
```

Рисунок 1 – Головний модуль програми

```
function [v x y]=choose(x1,y1,x2,y2,x3,y3,x4,y4)
% Вибір номера набору експериментальних даних
v=input("Введіть номер набору експериментальних даних :\n");
if (v == 1)  x=x1; y=y1; end
if (v == 2)  x=x2; y=y2; end
if (v == 3)  x=x3; y=y3; end
if (v == 4)  x=x4; y=y4; end
end
```

Рисунок 2 – Модуль вибору номера експерименту

```
function pre_graf(x,y)
 % Нанесення експериментальних точок на координатну сітку
 % для аналізу і вибору виду апроксимуючої функції
 figure(1);
 plot(x,y,'ko');
 xlabel('Значення аргумента x');
 ylabel('Значення функції y');
 grid on
end
```

Рисунок 3 – Модуль побудови експериментальних точок на координатній площині

Модуль `[p]=linear(x,y,m,n)` (рис.6) апроксимує експериментальні дані лінійною функцією. Вхідними параметрами цього модуля є вектори аргумента `x` та функції `y`, які зберігають значення конкретного набору експериментальних даних, довжина цих векторів `n` та власне порядковий номер `m` апроксимуючої функції (для лінійної функції він має значення 1). Модуль `[p]=linear(x,y,m,n)` розраховує значення елементів допоміжних векторів: `xx` (містить квадрати елементів вектора `x`), `xy` (містить добутки значень елементів вектора `x` на значення елементів вектора `y`), та середні значення `sx, sy, sxx, sxy` векторів `x, y, xx, xy`. Далі модуль `linear(x,y,m,n)` формує головну матрицю та стовпчик вільних членів системи

```
function koef_kor(x,y,n)
 % Розрахунок коефіцієнта кореляції за експериментальними даними
 xc=sum(x)/n;
 yc=sum(y)/n;
 xic=x-xc;
 yic=y-yc;
 xicyic=abs((x-xc).*(y-yc));
 xicxic=(x-xc).*(x-xc);
 yicyic=(y-yc).*(y-yc);
 sxicyic=sum(xicyic);
 sxicxic=sum(xicxic);
 syicyic=sum(yicyic);
 Rxy=sxicyic/(n-1);
 SigmaX=sqrt(sxicxic/(n-1));
 SigmaY=sqrt(syicyic/(n-1));
 Kxy=Rxy/(SigmaX.*SigmaY);

 fprintf('\n Розрахунок коефіцієнта кореляції за експериментальними ');
 fprintf('даними: \n');
 fprintf('      x(i)       y(i)     v (i)-xc     y(i)-yc   ');
 fprintf('(x(i)-xc)*(y(i)-yc)  (x(i)-xc)^2  (y(i)-yc)^2 \n');
 for i=1:n
  fprintf('\n %10.4f %10.4f %10.4f %10.4f %15.4f %15.4f %10.4f ', ...
        x(i), y(i), xic(i), yic(i), xicyic(i), xicxic(i), yicyic(i) );
 end
 fprintf('\n')
 fprintf('\n %10.4f %10.4f %10.4f %10.4f %15.4f %15.4f %10.4f ', ...
              xc, yc, 0, 0,  sxicyic, sxicxic, syicyic );
 fprintf('\n')
 fprintf('\nДругий змішаний момент Rxy=%10.4f ',Rxy );
 fprintf('\nСереднє квадратичне відхилення аргумента SigmaX=%10.4f ', ...
             SigmaX );
 fprintf('\nСереднє квадратичне відхилення функції SigmaY=%10.4f ', ...
             SigmaY );
 fprintf('\nКоефіцієнт кореляції Kxy=%10.4f ', Kxy );
 fprintf('\n');
 stop=input("Натисніть яку-небудь клавішу, щоб продовжити...");
end
```

Рисунок 4 – Модуль розрахунку коефіцієнта кореляції

```
function [ozn]=approximation(x,y,n,ozn)
 % Апроксимація експериментальних даних
 % різними типами функцій
 for m=1:4
  % Апроксимація експериментальних даних
  % лінійною залежністю y=a0*x+a1
  if (m == 1) ef1=linear(x,y,m,n); ozn=[ozn 1/ef1]; end
  % Апроксимація експериментальних даних
  % параболічною залежністю y=a0*x^2+a1*x+a2
  if (m == 2) ef2=parabolic(x,y,m,n,ozn); ozn=[ozn 1/ef2]; end
  % Апроксимація експериментальних даних
  % експоненціальною залежністю y=a0*exp(a1*x).
  if (m == 3) ef3=exponent(x,y,m,n,ozn); ozn=[ozn 1/ef3]; end
  % Апроксимація експериментальних даних
  % степеневою залежністю y=a0*x^a1.
  if (m ==4 ) ef4=powwer(x,y,m,n,ozn); ozn=[ozn 1/ef4];  end
 end
end
```

Рисунок 5 – Модуль управління процесом апроксимації даних різними функціями

лінійних рівнянь для визначення параметрів лінійної функції і розв'язує цю систему. Це дає змогу розрахувати значення елементів вектора `yt`, тобто теоретичні значення лінійної функції, відхилення `r` теоретичних значень `yt` функції від експериментальних `y` та квадрати цих відхилень `rr`. Результати розрахунків виводяться на екран у зручному для аналізу табличному вигляді. Функція `[p]=linear(x,y,m,n)` повертає одне єдине значення – середню величину квадрата відхилень теоретичних значень `yt` функції від експериментальних `y`, яке стає одним з елементів вектора ознак `[ozn]`. За таким же принципом побудовані ще три модулі для апроксимації експериментальних даних параболічною, показниковою та степеневою функцією. Відрізняються вони лише кількістю допоміжних векторів, тому вони у даній статті не приведені.

```
function [p]=linear(x,y,m,n)
    % Апроксимація експериментальних даних лінійною функцією y=a0*x+a1

    % Формування допоміжних масивів:
    xx=x.*x;
    xy=x.*y;

    % Обчислення допоміжних сум:
    sx=sum(x)/n;
    sy=sum(y)/n;
    sxx=sum(xx)/n;
    sxy=sum(xy)/n;

    % Формування головної матриці СЛАР
    % для визначення коефіцієнтів теоретичної залежності:
    fprintf('\n')
    disp(' Головна матриця СЛАР та стовпчик вільних членів :')
    A=[sxx sx;
    sx 1]
    % Формування вектора-стовпчика вільних членів СЛАР
    b=[sxy;sy]
    % Розв'язування СЛАР за допомогою оберненої матриці
    z=inv(A)*b;

    % Формуємо коефіцієнти СЛАР:
    a0=z(1);
    a1=z(2);

    % Формування вектора теоретичних значень функції:
    fprintf('\n')
    disp(' Коефіцієнти рівняння  y=a0*x+a1*:')
    fprintf('\n a0=%10.4f     a1=%10.4f\n\n', a0, a1)
    yt=a0*x+a1;

    % Будуємо графік теоретичної залежності:
    figure(2);
    plot(x,y,'ko',x,yt,'g-');
    xlabel('Значення аргумента x');
    % Формування вектора квадрата відхилень
    % теоретичних та експериментальних значень функції
    rr=r.*r;
    figure(4);
    plot(x,rr,'g-');
    xlabel('Значення аргумента x');
    ylabel('Квадрат відхилень теоретичних та експериментальних значень rr');
    srr=sum(rr)/n;

    % Обчислюємо суму квадратів відхилень
    % теоретичних та експериментальних значень функції:
    disp('   x(i)        y(i)       x(i)^2    x(i)*y(i)     yt(i)      rr(i)')
    for i=1:n
     fprintf('\n %10.4f %10.4f %10.4f %10.4f %10.4f %10.4f ', ...
                 x(i), y(i), xx(i), xy(i), yt(i), rr(i))
    end
    fprintf('\n')
    fprintf('\n %10.4f %10.4f %10.4f %10.4f %10.4f %10.4f ', ...
                 sx, sy, sxx, sxy, 0, srr)
    fprintf('\n');
    p = srr;
    fprintf('\n ozn(%d)=%10.4f \n',m, p);
    stop=input("Натисніть яку-небудь клавішу, щоб продовжити...");
end
```

Рисунок 6 – Модуль апроксимації даних лінійною функцією

Модуль `est_eff(ozn,v)` (рис.7) здійснює вибір найбільш ефективного варіанту типу теоретичної функції.

Головний результат роботи програми – це тип апроксимуючої функції. Ця функція, як видно з рис. 8, визначається за значенням параметра ефективності. Для даного набору експериментальних даних – це функція №2, тобто параболічна функція.

```
function est_eff(ozn,v)
 % Порівняльний аналіз результатів апроксимації експериментальних даних
 % різними типами функцій
 figure(5);
 m=1:4;
 bar(m,ozn,'r');
 xlabel('Порядковий номер теоретичної залежності m');
 ylabel('Значення параметра ефективності методу ef');
 grid on
 [val_max, pos_max]= max(ozn);
 fprintf('\nЗагальний висновок за результатами досідження:');
 fprintf('\nНайбільш еффективною функцією ');
 fprintf('\nдля апроксимації даних експерименту номер %2d',v);
 fprintf('\nє функція номер %2d.',pos_max);
 fprintf('\nЗначення параметра еффективності для цієї функції ');
 fprintf('дорівнює %6.4f \n',val_max);
 stop=input("Розвязок задачі завершено!.\n");
 fprintf('\n')
end
```

Рисунок 7 – Модуль оптимального вибору типу апроксимуючої функції

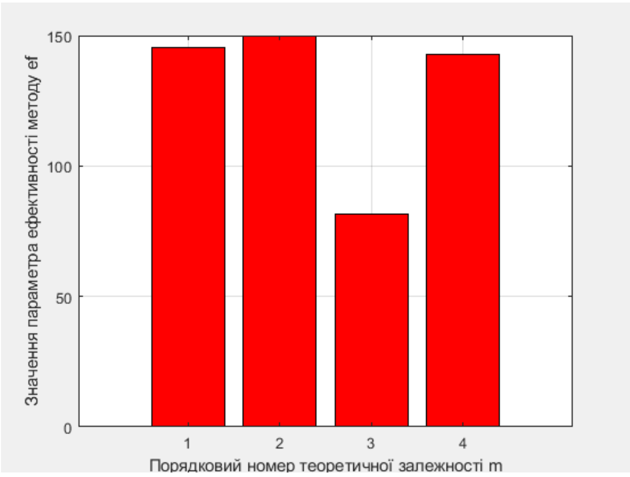

Рисунок 8 – Графічна інтерпретація оптимального вибору типу апроксимуючої функції

Висновки. 1. У даній роботі розроблена програма засобами Matlab для оптимального вибору типу теоретичної функції для апроксимації експериментальних даних зі множини, до якої включено лінійну, параболічну, показникову та степеневу функції, і виконані дослідження за допомогою цієї програми. Програма протестована і показала надійну роботу на чотирьох наборах експериментальних даних, що дає підстави зробити висновок про її придатність для вирішення поставлених у роботі завдань. Запропонована програма дозволяє більш обгрунтовано і точно підібрати тип апроксимуючої функції, ніж це робилося традиційним шляхом, а сам процес вибору вирішити засобами сучасної інформаційної технології.

2. Модульний принцип побудови програми дозволяє у подальшому розширити множину функцій, які використовуються для апроксимації експериментальних даних, шляхом добавляння нових функцій до вже існуючих.

3. Вважаємо, що розроблена програма може бути використана як у навчальному процесі, так і в інженерній практиці під час оброблення даних реальних експериментів.

Список бібліографічного опису.

1. Пех П.А., С.В. Лавренчук, М.В.Делявський, С.В. Гринюк. Лабораторний практикум з програмування мовою С /С++ – Луцьк : Вежа-Друк, 2020. – 228 с.
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.

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

2. 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