Лабораторна робота № 3

**Тема роботи:** Оператори керування обчислювальним процесом MatLab.

**Мета роботи**: Вивчення операцій оператори керування обчислювальним процесом.

Постановка задачі : Обчислити значення функції для будь-якого множини значень аргументу, матричні обчислення.

і. **Оператори керування обчислювальним процесом**

Взагалі оператори керування необхідні, головним чином, для організації обчислювального процесу, що записується у вигляді деякого тексту програми мовою програмування високого рівня. При цьому до операторів керування обчислювальним процесом звичайно відносять оператори безумовного переходу, умовних переходів (розгалуження обчислювального процесу) та оператори організації циклічних процесів.

Однак система MatLAB побудована таким чином, що ці оператори можуть бути використані при роботі MatLAB у режимі калькулятора.

У мoвi MatLAB відсутній оператор безумовного переходу і у зв'язку з цим, немає поняття мітки. Це е недоліком цієї мови i робить утрудненим повернення обчислювального/процесу до будь якого попереднього чи наступного місця програми.

Уci оператори умовного переходу i оператори циклу побудовані у MatLAB у вигляді складеного оператора, який починається зі службового слова **if**, **while** або for i закінчується службовим словом **end**.

Уci оператори всередині між цими словами сприймаються системою як частини одного складного оператора. Тому натискання на клавішу <**Enter**> для переходу до іншого рядка не приводить у цьому випадку до виконання цих операторів. Виконання операторів починається лише тоді, коли є введеним слово **end** i потім натиснути клавішу < **Enter** >.

Якщо кілька складених onepaторів такого типу вкладені один в другий, обчислення починаються лише тоді, коли записаний кінець **end** найбільш охоплюючого (зовнішнього) складеного оператора. 3 цього випливає можливість здійснення у режимі калькулятора досить складних i об'ємних (що складаються з багатьох рядків i onepaтopiв) обчислень, якщо вони охоплені складеним оператором.

1.10.1. Оператори умовного переходу

Конструкція оператора переходу за умовою у загальному випадку е такою:

if <умова>

<оператори 1>

**else**

<оператори 2>

**end**

Виконується він таким чином. Спочатку перевіряється, чи виконується задана умова. Якщо її виконано, програма виконує сукупність операторів, яку записано у розділі <оператори 1>. Якщо умову не виконано, виконується послідовність операторів розділу <оператори 2>.

Скорочена форма умовного оператора має вигляд:

**if** <умова>

<оператори>

**end**

Дія оператора у цьому випадку е аналогічною, окрім того, що при невиконанні заданої умови виконується наступний оператор (після оператора **end**).

Легко помітити недоліки цього оператора, які випливають з відсутності оператора безумовного переходу: уся частина програми, що виконується залежно від умови, повинна міститися усередині операторних дужок **if** та **end**.

Для опису умови використовується вираз типу:

<ім'я змінної 1> <операція порівнювання> <ім'я змінної 2>.

Операції порівнювання у мові Mat LAB можуть бути такими:

< - менше;

> - більше;

<= - менше або дорівнює;

>= - більше або дорівнює;

== - дорівнює;

~= - не дорівнює.

Умова може бути складеною, тобто складатися з кількох простих умов, які об'єднуються знаками логічних операцій. Знаками логічних операцій у мові MatLAB є такі:

& - логічна операція " ТА" ("AND");

| - логічна операція "АБО" ("OR");

~ - логічна операція "HI" ("NOT").

Логічна операція "Виключне АБО" може бути реалізованою за допомогою функції ***хоr***{АВ), де A i В - деякі умови.

**Оператори циклу**

У мовi MatLAB існують два різновиди операторів циклу – умовний та арифметичний.

Оператор циклу з передумовою має вигляд:

**while** <умова>

<оператори>

**end**

Оператори усередині циклу виконуються лише тоді, коли виконано умову, яку записано після слова **while**. При цьому серед операторів усередині циклу обов'язково повинен бути такий, що змінює значення одну з змінних, що вказана в умові циклу.

Наведемо приклад обчислення значення синуса при 3-х значеннях аргументу від 0.2 до 4 з кроком у 0.2:

» i = 1

» **while** i <= 3

х = i/5;

si = sin(х);

disp([x,si])

i = i+l;

**end**

0.2000 0.1987

0.4000 0.3894

0.6000 0.5646

*Примітка.* Зверніть увагу на те, якими засобами у вказаному прикладі забезпечено виведення на екран кількох змінних у один рядок. Для цього використовується застосовуваний раніше оператор disp. Але згідно з правилами застосування цього оператора в ньому повинен бути лише один аргумент (текст, змінна або матриця). Щоб обійти цю перепону, потрібно кілька числових змінних об’єднати у єдиний об’єкт - вектор-рядок, а останнє легко виконується за допомогою звичайної операції формування вектора-рядка з окремих елементів [xl,x2, ...,xN]. Таким чином, за допомогою оператора вигляду:

disp ([xl,x2,...,xN])

можна забезпечити виведення результату обчислень у вигляді таблиці даних.

Арифметичний оператор циклу має вигляд:

**for** <1м'я> = <ПЗ> : <К> : <КЗ>

<оператори>

**end**,

де <ім'я> - ім'я керуючої змінної циклу - "лічильника" циклу; <ПЗ> - задане початкове значення uiєї змінної; <K> - значення кроку, за яким вона має змінюватися; <КЗ> - кінцеве значення змінної циклу. У цьому випадку <оператори> усередині циклу виконуються декілька разів (кожен раз за інших значень керуючої змінної) доти, поки значення керуючої змінної не вийде за межі інтервалу м1ж <ПЗ> i <K3>. Якщо параметр <К> не вказано, за замовчуванням його значення вважається рівним одиниці.

Щоб достроково вийти з циклу (наприклад, при виконанні деякої умови) застосовують оператор **break**. Якщо програма стикається з цим оператором, виконання циклу достроково припиняється i починає виконуватися наступний після слова **end** оператор.

Для прикладу використаємо попереднє завдання:

» а = [‘ i ‘,’ х ‘,’ sin(x) ‘];

» for i = 1 : 3

x = i/5;

si = sin(x);

if i==1

disp(a)

end

disp([i,x,si])

end

i x sin(x)

і.0000 0.2000 0.і987

2.0000 0.4000 0.3894

3.0000 0.6000 0.5646

**Завдання**

Завдання 1.1. Обчислити суму нескінченного ряду з заданою точністю ε. Визначити кількість членів, необхідних для досягнення заданої точності. Створити два типи програм за допомогою різних операторів циклу мови MatLAB. Виведіть до командного вікна таблицю результатів обчислень. Варіанти вибрати з таблиці.

|  | | | |
| --- | --- | --- | --- |
| №  вар. | Загальний член ряду | Х | ε |
| 1 |  | 0,12 | 2\*10-5 |
| 2 |  | 0,81 | 4\*10-5 |
| 3 |  | 0,54 | 10-5 |
| 4 |  | 0,21 | 10-4 |
| 5 |  | 0,27 | 10-5 |
| 6 |  | 0,85 | 10-6 |
| 7 |  | 0,29 | 2\*10-5 |
| 8 |  | 0,72 | 10-5 |
| 9 |  | 0,19 | 10-5 |
| 10 |  | 0,64 | 5\*10-6 |
| 11 |  | 0,56 | 2\*10-5 |
| 12 |  | 0,63 | 3\*10-5 |
| 13 |  | 0,4 | 2\*10-5 |
| 14 |  | 1,2 | 5\*10-5 |
| 15 |  | 0,8 | 10-5 |

**Запитання**

1. Які засоби керування перебігом обчислювального процесу icнyють у мовi MatLAB?

2. Як можна організувати обчислення за циклом мовою MatLAB?

3. Як організувати виведення таблиці результатів обчислення до командного вікна MatLAB?

4. Як здійснити складні (багато операторні) обчислення у режимі калькулятора?

Література

1. Ю.Ф. Лазарев. Початки програмування у середовищі *MatLab* /Навчальний посібник. –К.: Корнійчук, 1999. – 160 с.