Практична робота №2
Оператори керування обчислювальним процесом в середовищі обробки біомедичної інформації MATLAB

Мета роботи: вивчити процедуру обчислювального процесу в процесі обробки біосигналів в середовищі MATLAB із застуванням операторів безумовного переходу, умовних переходів і операторів організації циклічних процесів.

ТЕОРЕТИЧНІ ВІДОМОСТІ

2.1 Загальні відомості

Загалом кажучи, оператори керування необхідні, головним чином, для організації обчислювального процесу в процесі обробки біосигналів, який записується у виді деякого тексту програми на мові програмування високого рівня. При цьому до операторів керування обчислювальним процесом звичайно відносять оператори безумовного переходу, умовних переходів (розгалуження обчислювального процесу) і оператори організації циклічних процесів. Проте система MatLAB побудована таким чином, що ці оператори можуть бути використані і при роботі MatLAB у режимі калькулятора.

Усі оператори циклу й умовного переходу побудовані в MatLAB у виді складного оператора, що починається з одного зі службових слів **if, while, switch** або **for** і закінчується службовим словом **end.** Оператори всередині між цими словами сприймаються системою як частини одного складного оператора. Тому натискання клавіші <Enter> для переходу до наступного рядка не призводить у цьому випадку до виконання цих операторів. Виконання операторів починається лише тоді, коли введена "завершувальна дужка" складного оператора у виді мітки **end** , а потім натиснуто клавішу <Enter>. Якщо декілька складних операторів такого типу вкладені один в інший, обчислення починаються лише тоді, коли записаний кінець **end** найбільш охоплюючого (зовнішнього) складного оператора. З цього випливає можливість здійснення навіть у режимі калькулятора досить складних і об'ємних (що складаються з багатьох рядків і операторів) обчислень, якщо вони охоплені складним оператором.

2.2 Оператор умовного переходу

Конструкція оператора переходу за умовою в загальному виді є такою:

```
if <умова>
        <оператори1>
        else
        <оператори2>
end
```

Працює оператор у такий спосіб. Спочатку перевіряється, чи виконується зазначена умова. Якщо її виконано, програма виконує сукупність операторів, що записана в поділі <оператори1>. Якщо умову не виконано, виконується послідовність операторів поділу <оператори2>.

Скорочена форма умовного оператора має вид:

if <умова>
<оператори>
end

Дія оператора в цьому випадку аналогічно, за винятком того, що при невиконанні заданої умови виконується оператор, наступний за оператором **end.**

Легко помітити хиби цього оператора, що випливають із відсутності оператора безумовного переходу: усі частини програми, що виконуються в залежності від умови, повинні розміщатися усередині операторних дужок **if** і **end.**

Як умова використовується вирази типу:
<ім'я змінної 1> <операція порівнювання> <ім'я змінної2>
Операції порівнювання в мові MatLAB можуть бути такими:

< - менше;
> - більше;
<= - менше або дорівнює;
>= - більше або дорівнює;
= = - дорівнює;
~ = - не дорівнює.

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

& - логічна операція "І" ("AND");
| - логічна операція "АБО" ("OR");
~ - логічна операція "НІ" ("NOT").

Логічна операція "Виняткове АБО" може бути реалізована за допомогою функції **хог**(А,У), де А і В - деякі умови.

Є припустимою ще одна конструкція оператора умовного переходу:

if <умова1>
<оператори1>
elseif
<умова2>
<оператори2>
elseif

```
   <умова3>
   <оператори3>
else
   <оператори>
end
```

Оператор **elseif** виконується тоді, коли <умова1> не виконана. При цьому спочатку перевіряється <умова2>. Якщо її виконано, виконуються <оператори2>, якщо ж ні, <оператори2> ігноруються, і відбувається перехід до наступного оператора **elseif,** тобто до перевірки виконання <умови3>. Аналогічно, при виконанні її виконуються <операторьі3>, у противному випадку відбувається перехід до наступного оператора **elseif.** Якщо жодну з умов в операторах **elseif** не виконано, виконуються <оператори>, що містяться за оператором **else.** У такий спосіб може бути забезпечене розгалуження програми по кількох напрямках.

2.3 Оператор переключення

Оператор переключення має таку структуру:

```
switch <вираз, скаляр або рядок символів>
   case <значення1>
      <оператори1>
   case <значення2>
      <оператори2>
   otherwise
      <оператори>
end
```

Він здійснює розгалужування обчислень у залежності від значень деякої змінної або виразу, порівнюючи значення, отримане в результаті обчислення виразу в рядку **switch,** із значеннями, зазначеними в рядках із словом **case.** Відповідна група операторів **case** виконується, якщо значення виразу збігається зі значенням, зазначеним у відповідному рядку **case.** Якщо значення виразу не збігається з жодним із значень у групах **case,** виконуються оператори, що наслідують **otherwise.**

2.4 Оператори циклу

У мові MatLAB є два різновиди операторів циклу - умовний і арифметичний.

Оператор циклу з передумовою має вид:

```
while <умова> <оператори>
end
```

Оператори усередині циклу виконуються лише в тому випадку, якщо є виконаною умова, записана після слова **while.** При цьому серед операторів усередині циклу обов'язково повинні бути такі, що змінюють значення однієї зі змінних, зазначених в умові циклу.

Наведемо приклад обчислення значення синуса при 3 значеннях аргументу від 5 до 15 із кроком 5:

```
i = 1;
while i <= 3
   x = i*5;
   si =sin(x);
   disp([x,si])
   i=i+1;
end
```

Результат:

```
5.0000   -0.9589
10.0000   -0.5440
15.0000    0.6503
```

Примітка. Зверніть увагу на те, якими засобами в зазначеному прикладі забезпечено виведення на екран значень кількох змінних одним рядком.

Для цього використовується оператор **disp,** який раніше застосовувався. Але, відповідно до правил застосування цього оператора, у ньому повинний бути тільки один аргумент (текст, змінна або матриця). Щоб обминути цю перешкоду, потрібно декілька числових змінних об'єднати в єдиний об'єкт - вектор-рядок, а останнє легко виконується за допомогою звичайної операції формування вектора-рядка з окремих елементів:

[xl, x2, ... , x].

Таким чином, за допомогою оператора виду:

disp([x1, x2, ... , x])

можна забезпечити виведення результатів обчислень у виді таблиці даних.

Арифметичний оператор циклу має вид:

```
for <ім'я> = <ПЗ> : <К> : <КЗ>
        <оператори>
end,
```

де <ім'я> - ім'я керуючої змінної циклу - "лічильника" циклу; <ПЗ> - задане початкове значення цієї змінної; <К> - значення кроку, із яким вона повинна змінюватися; <КЗ> - кінцеве значення змінної циклу. У цьому випадку

<оператори> усередині циклу виконуються кілька разів (кожного разу при новому значенні керуючої змінної) доти, поки значення керуючої змінної не вийде за межі інтервалу між <ПЗ> і <КЗ>. Якщо параметр <К> не зазначений, за замовчуванням його значення приймається 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, і починає виконуватися оператор, наступний за словом **end** циклу.

Для приклада використаємо попереднє завдання: »

```
a = ['     i','      x','      sin(x)'];
for i = 1:5
   x = i/5;
   si = sin(x);
   if i==1 disp(a)
   end
   disp([i,x,si])
end;
```

Результат:

i	x	sin(x)
1.0000	0.2000	0.1987
2.0000	0.4000	0.3894
3.0000	0.6000	0.5646
4.0000	0.8000	0.7174
5.0000	1.0000	0.84151

У такий спосіб можна забезпечити виведення інформації у вигляді таблиць.

ЕКСПЕРИМЕНТАЛЬНА ЧАСТИНА

2.5 Завдання для виконання

Відповідно до таблиці 1.1 обчислити точні (по стандартних функціях MatLAB) значення відповідної функції у діапазоні змінювання аргументу від x_1 до x_2 в m рівновіддалених точках цього діапазону, включаючи його межі.

Таблиця 1.1 – Варіанти завдань

Варіант	x1	x2	m	Функція	Критерій прийняття рішення
1	0.2	5	20	$Y = \sum (-1)^2 \frac{x^{2k-1}}{2k-1}$	$Результат = \begin{cases} Норма, & якщо \quad Y < 5 \\ Патологія, & якщо \quad Y > 5 \end{cases}$
2	1	10	30	$Y = 1 - \sum (-1)^k \frac{x^{2k}}{2k}$	$Результат = \begin{cases} Норма, & якщо \quad Y \le 3 \\ Патологія, & якщо \quad Y \ge 3 \end{cases}$

3	0.3	3	40	$Y = 1 + \sum (-1)^k \frac{x^{2k}}{2k}$	$Результат = \begin{cases} Норма, & якщо \quad 1 < Y \le 3 \\ Патологія, & якщо \quad Y > 3 \end{cases}$
4	0.4	4	50	$Y = \sum (-1)^2 \frac{x^{2k+1}}{2k}$	$Результат = \begin{cases} Норма, & якщо \quad 3 > Y > 5 \\ Патологія, & якщо \quad 4 > Y > 3 \end{cases}$
5	0.5	5	30	$Y = \sum (-1)^3 \frac{x^{2k-2}}{2k-2}$	$Результат = \begin{cases} Норма, & якщо \quad 0 < Y < 1 \\ Патологія, & якщо \quad Y < 2 \end{cases}$

За обчисленими значення функції Y прийняти рішення (результат), а саме норма чи патологія в залежності від умови таблиці 1.1

2.6 Зміст звіту

2.6.1 Тема та мета роботи.
2.6.2 Завдання до роботи.
2.6.3 Відобразити отримані результати (програму та результати її роботи)
2.6.4 Висновки за результатами виконаної роботи.

КОНТРОЛЬНІ ЗАПИТАННЯ

1. Які засоби керування перебігом обчислювального процесу передбачені в мові MatLAB?
2. Як можна організувати обчислення за циклом мовою MatLAB?
3. Як організувати виведення таблиці результатів обчислень у командне вікно MatLAB?
4. Як здійснити складні (багатооператорні) обчислення в режимі калькулятора?