

Computació Numèrica

Laboratori 2. Algorismes

M. Àngela Grau Gotés

Departament de Matemàtica Aplicada II
Universitat Politècnica de Catalunya · BarcelonaTech.

28 de febrer de 2018

drets d'autor

“Donat el caràcter i la finalitat exclusivament docent i eminentment il·lustrativa de les explicacions a classe d'aquesta presentació, l'autor s'acull a l'article 32 de la Llei de propietat intel·lectual vigent respecte de l'ús parcial d'obres alienes com ara imatges, gràfics o altre material contingudes en les diferents diapositives”

Índex

1 Sessió 2.

- Bucles FOR
- Bucles WHILE
- Sentències BREAK - RETURN - ERROR
- Sentència IF
- Operadors
- Per practicar

2 Exercicis

El manual de referència és

<http://www.mathworks.es/es/help/matlab/>

Bucles FOR

Permeten de repetir una sentència, o un grup de sentències un nombre fix de vegades. La seva expressió general és:

```
for i=n1:n2:n3
    instruccions;
    ...
end
```

on n_1 , n_2 , n_3 són el valor inicial, l'increment i el valor final de l'índex del bucle. Si les instruccions de l'interior del bucle s'acaben amb ";" els passos intermitjos no es veuen en pantalla.

Bucles WHILE

Permeten de repetir una sentència fins que es compleix una condició lògica. La seva expressió general és:

```
while condició  
    instruccions;  
end
```

Sentències BREAK - RETURN - ERROR

La sentència `break` permet de sortir fora d'un bucle `for` o d'una sentència `if`.

Sentències BREAK - RETURN - ERROR

La sentència `break` permet de sortir fora d'un bucle `for` o d'una sentència `if`.

La sentència `return` obliga a Matlab a sortir d'una funció abans de la fi d'aquesta.

Sentències BREAK - RETURN - ERROR

La sentència `break` permet de sortir fora d'un bucle `for` o d'una sentència `if`.

La sentència `return` obliga a Matlab a sortir d'una funció abans de la fi d'aquesta.

La sentència `error('msg')` obliga a Matlab a mostrar un missatge d'error.

Sentència IF

Permet bifurcar el flux del programa.

```
if condició
    instruccions si es
    verifica la condició
else
    altrament
end
```

Language Fundamentals

Syntax, operators, data types, array indexing and manipulation

MATLAB is an abbreviation for "matrix laboratory." While other programming languages usually work with numbers one at a time, *MATLAB*® operates on whole matrices and arrays. Language fundamentals include basic operations, such as creating variables, array indexing, arithmetic, and data types.

Entering Commands

Build and run *MATLAB* statements

Matrices and Arrays

Array indexing, concatenation, sorting, and reshaping

Operators and Elementary Operations

Arithmetic, relational, logical, set, and bit-wise operations

Special Characters

Symbols

Data Types

Numeric arrays, character arrays, tables, structures, and cell arrays; data type conversion

Operators and Elementary Operations

Arithmetic, relational, logical, set, and bit-wise operations

Arithmetic

Addition, subtraction, multiplication, division, power, rounding

Relational Operations

Value comparisons

Logical Operations

True or false (Boolean) conditions

Set Operations

Unions, intersection, set membership

Bit-Wise Operations

Set, shift, or compare specific bit fields

Relational Operators

Relational operators compare operands quantitatively, using operators like "less than" and "not equal to." The following table provides a summary. For more information, see the [relational operators](#) reference page.

Operator	Description
<	Less than
<=	Less than or equal to
>	Greater than
>=	Greater than or equal to
==	Equal to
~=	Not equal to

Operator	Operation
&	Logical AND
&&	Logical AND with shortcut evaluation
	Logical OR
	Logical OR with shortcut evaluation
xor	Logical exclusive OR
~	Logical NOT

Truth Table for Logical Operations

The following reference table shows the results of applying the binary logical operators to a series of logical 1 (true) and logical 0 (false) scalar pairs. To calculate NAND, NOR or XNOR logical operations, simply apply the logical NOT operator to the result of a logical AND, OR, or XOR operation, respectively.

Inputs A and B		and A & B	or A B	xor xor(A,B)	not ~A
0	0	0	0	0	1
0	1	0	1	1	1
1	0	0	1	1	0
1	1	1	1	0	0

Logical Operators: Short-Circuit && ||

Logical operations with short-circuiting

[collapse all in page](#)

Syntax

```
expr1 && expr2
```

[example](#)

```
expr1 || expr2
```

[example](#)

Description

`expr1 && expr2` represents a logical AND operation that employs [short-circuiting behavior](#). That is, `expr2` is not evaluated if `expr1` is logical 0 (false). Each expression must evaluate to a scalar logical result.

[example](#)

`expr1 || expr2` represents a logical OR operation that employs [short-circuiting behavior](#). That is, `expr2` is not evaluated if `expr1` is logical 1 (true). Each expression must evaluate to a scalar logical result.

[example](#)

Iteracions

Calcular el valor x_{10} del mètode iteratiu següent:

$$x_k = \frac{1}{2} \left(x_{k-1} + \frac{2}{x_{k-1}} \right) \quad k \geq 1 \text{ i } x_0 = 2.$$

Iteracions

Calcular el valor x_{10} del mètode iteratiu següent:

$$x_k = \frac{1}{2} \left(x_{k-1} + \frac{2}{x_{k-1}} \right) \quad k \geq 1 \text{ i } x_0 = 2.$$

Fent ús de les instruccions

Bucle for

Bucle while

En tots els casos comparar el resultat obtingut amb el valor $\sqrt{2}$

Funcions

Avalueu les funcions

$$f(x) = \sqrt{x^2 + 1} - 1, \quad g(x) = x^2 / \sqrt{x^2 + 1} + 1$$

per a la successió de valors de $x_n = 8^{-n}$, $n \geq 1$.

Funcions

Avalueu les funcions

$$f(x) = \sqrt{x^2 + 1} - 1, \quad g(x) = x^2 / \sqrt{x^2 + 1} + 1$$

per a la successió de valors de $x_n = 8^{-n}$, $n \geq 1$.

Encara que $f(x) = g(x)$, l'ordinador dona resultats diferents. Quins resultats són de fiar i quins no? Per què?

Autoavaluació

Exercici 1 Escriviu un script que:

Calculi $1 + 2 + \dots + n$ per a diferents valors de n .

Calculi $1^p + 2^p + \dots + n^p$ per a diferents valors de n i p .

Autoavaluació

Exercici 1 Escriviu un script que:

Calculi $1 + 2 + \cdots + n$ per a diferents valors de n .

Calculi $1^p + 2^p + \cdots + n^p$ per a diferents valors de n i p .

Exercici 2 Definim el nombre e com $e = \sum_{k=0}^{\infty} \frac{1}{k!}$. Per calcular-ne una aproximació considerem el mètode iteratiu definit per

$$x_k = x_{k-1} + \frac{1}{k!}, \quad k \geq 1, \quad x_0 = 1$$

Escriviu un script que calculi els 20 primers termes de la recurrència, compareu els vostres resultats amb el valor $\exp(1)$ retornat per Matlab.

Autoavaluació

Exercici 3 Sigui $p(x) = (x - 1)(x - 2)(x - 3)\dots(x - 10)$, el polinomi amb arrels els deu primers nombres naturals, definim el polinomi

$q(x) = p(x) + \frac{1}{2^{13}} x^9$, modificant lleugerament el coeficient de x^9

respecte de $p(x)$. Com haurien de ser les arrels del polinomi $q(x)$?
Calculeu-les. Com són en realitat?

Autoavaluació

Exercici 3 Sigui $p(x) = (x - 1)(x - 2)(x - 3)\dots(x - 10)$, el polinomi amb arrels els deu primers nombres naturals, definim el polinomi

$q(x) = p(x) + \frac{1}{2^{13}} x^9$, modificant lleugerament el coeficient de x^9

respecte de $p(x)$. Com haurien de ser les arrels del polinomi $q(x)$? Calculeu-les. Com són en realitat?

Exercici 4 Resolució de sistemes lineals $Ax = b$

Resoleu els sistemes d'equacions lineals,

$$a) \begin{cases} x + 2y = 3 \\ 0.499x + 1.001y = 1.5 \end{cases} \quad b) \begin{cases} x + 2y = 3 \\ 0.5x + 1.001y = 1.5 \end{cases}$$

per qualsevol mètode que conegueu. Com són les dues solucions?

Autoavaluació

Exercici 5 Càlcul d'integrals per recurrència

Per calcular les integrals $I_n = \int_0^1 x^n e^{x-1} dx$, $n \geq 1$, disposem de dos mètodes iteratius diferents:

$$\text{a) } I_{n-1} = \frac{1 - I_n}{n}, \quad n \geq 2 \quad \text{on } I_0 = 1,$$

$$\text{b) } I_n = 1 - n I_{n-1}, \quad n \geq 1 \quad \text{on } I_0 = 1/e.$$

Calculeu I_{30} pels dos mètodes. Obteniu els mateixos resultats ? Sabrieu donar una explicació?

Autoavaluació

Exercici 6 Escriviu un script per a resoldre les equacions de segon grau $ax^2 + bx + c = 0$, on a , b , c són nombres reals. Cal distingir els casos trivials i els casos $a = 0$, $b^2 - 4ac < 0$ i $b^2 - 4ac > 0$. Feu un joc de proves. Especialment ompliu la taula següent:

a	1	0	0	1	1	1	1	1	1	10^{-30}	10^{-25}
b	4	4	0	2	2	1	0	0	4	10^{30}	10^{32}
c	2	2.3	2.3	2.3	1	0	-1	1	3.999999999	10^{30}	10^{30}
x_1											
x_2											