



# Anonymous functions

```
>> v
v =
    -4
    -7

>> |
```

Command Window

```
1 A = [1 2 3; 4 5 6];
2 z = [2; -3; 0];
3
4 v = fonctionExample(A,z);
5
6
7 f = @(A,x) A*x;
```

GUI

MATLAB and Octave for beginners

User-defined functions are very useful. They go into an M file and then you can reuse them or correct them or complete at a later time and no problem. You can have a library with several functions. OK. You can then use these functions when you want. There are also anonymous functions. OK. They are much faster to use. They are easier to define. But then you put them into trash and you cannot use them again. So How do you do with these anonymous functions. You arrived here. We had A and Z and then we would like to build a new function that does exactly the same thing as our function example so A and Z or A times X. But we don't want to store an M file. We just want to define a function f. OK. What is f? It's a function that takes A and X as input and here is the syntax. So ampersand A comma X. And then here is the algorithm is just A times X. And that's it. And the result is this A times X. And now we have f which is the function that takes these two arguments and returns A and X and then I can call A comma Z and I can see if I get the same results as V before and V is exactly the same. Yeah so these two functions do exactly the same but they are not done in the same.

Notes

Summary



# Anonymous functions

```
-12
-21

>> f
f =

@(A, x) a * A * x

>> |
```

Command Window

```
1 A = [1 2 3; 4 5 6];
2 z = [2; -3; 0];
3
4 v = fonctionExample(A,z);
5
6 % f calcule Ax
7 a = 3;
8 f = @(A,x) a*A*x;
9 clear a;
10
11
```

GUI

MATLAB and Octave for beginners

You cannot add commands unless you just do some commands before the definitions. And you cannot build a very complex algorithm with these anonymous functions. So boys we have a parameter to be inserted, for example a parameter which is a 3. OK. And then we want to use it there. We can use it by defining A equal to 3 before and then in the anonymous function make use of this A. So it's a parameter. It is not an input just something that we may want to change from time to time. Then I call f of A comma Z. So it is 3 times the vector that I had before so everything is correct. If I change my variable A, what do you expect that the function gives you a different result. But indeed not. It uses the value of a little A that was available at the definition of f. OK. I can even clear the variable A and the function will still work and now the initial value of A. So it is also useful to use these anonymous functions to integrate parameters directly and after just forget about those parameters. Here I can tell myself that we clean the small 'a' and because there is no need for that. So now I would like to generate another kind of functions, the inline functions. So how they are defined.

Notes

Summary



1m 54s

# Anonymous functions

```
-12
-21

>> f
f =

@(A, x) a * A * x

>> callExample
>>
```

Command Window

```
1 A = [1 2 3; 4 5 6];
2 z = [2; -3; 0];
3
4 v = fonctionExample(A,z);
5
6 % f calcule Ax
7 a = 3;
8 f = @(A,x) a*A*x;
9 clear a;
10
11 g = inline('A*x', 'A', 'x');
12 w = g(A,z);
```

GUI

MATLAB and Octave for beginners

So G is equal to inline and then I have to give a string which will be executed so I have the string A times X. And now I have to tell what are the input parameters. So my input are A comma X. Also given as string so here you see the syntax is quite different. And now I can evaluate G given A and X. So W is equal to G of A X. And I can run my script. So look at what W is? I see it is exactly the same as V.

Notes

Summary



3m 54s

# 4.1 Functions



- Build in functions
- User defined functions
- Anonymous functions
- Inline functions

There are functions that are predefined. They are built-in functions. These are accessible at any time. Then there are functions defined by the user. We saw a small example on how to implement it. We will see more later in the course. After there are anonymous and inline functions that we will use rather for simple definitions.

Notes

Summary



4m 46s

# 4.1 Functions



Use functions is the right way. Define new ones. Modify old ones. It is essential to have these functions and write down the algorithms and then your program inside files that you can then keep for ever. Organizing them into files has isolated some features and also avoid possible bugs. Unfortunately it is not possible to avoid all the bugs. There is no simple recipe for that. But still organize your work and then it will be easier. Often it is the experience that you will help you of identifying and solve your bugs.

Notes

Summary



5m 11s