MATLAB: Scripts and Functions

1. Introduction

2. In-class exercise

MATLAB: Scripts and Functions

Introduction

Scripts

- Scripts are the simplest kind of MATLAB program file because they have no input or output arguments

- Scripts are useful for automating series of MATLAB commands, such as computations that you have to perform repeatedly from the command line

- Scripts are created from the menu by choosing File → New → Script

- Commands can be entered sequentially and then saved as filename.m

- The script is executed by typing filename at the command line

Script Example

□ Create a script to evaluate the solution $x(t)$ of the logistic equation for any value x_0 and r

$$x(t) = \frac{1}{1 + \left(\dfrac{1}{x_0} - 1\right)e^{-rt}}$$

□ logistic_script.m

```
x0 = 0.1;
r = 0.5;
t = [0:0.1:10];
x = 1./(1+(1/x0-1)*exp(-r*t));
plot(t,x)
xlabel('t')
ylabel('x')
legend(['r=',num2str(r)])
```
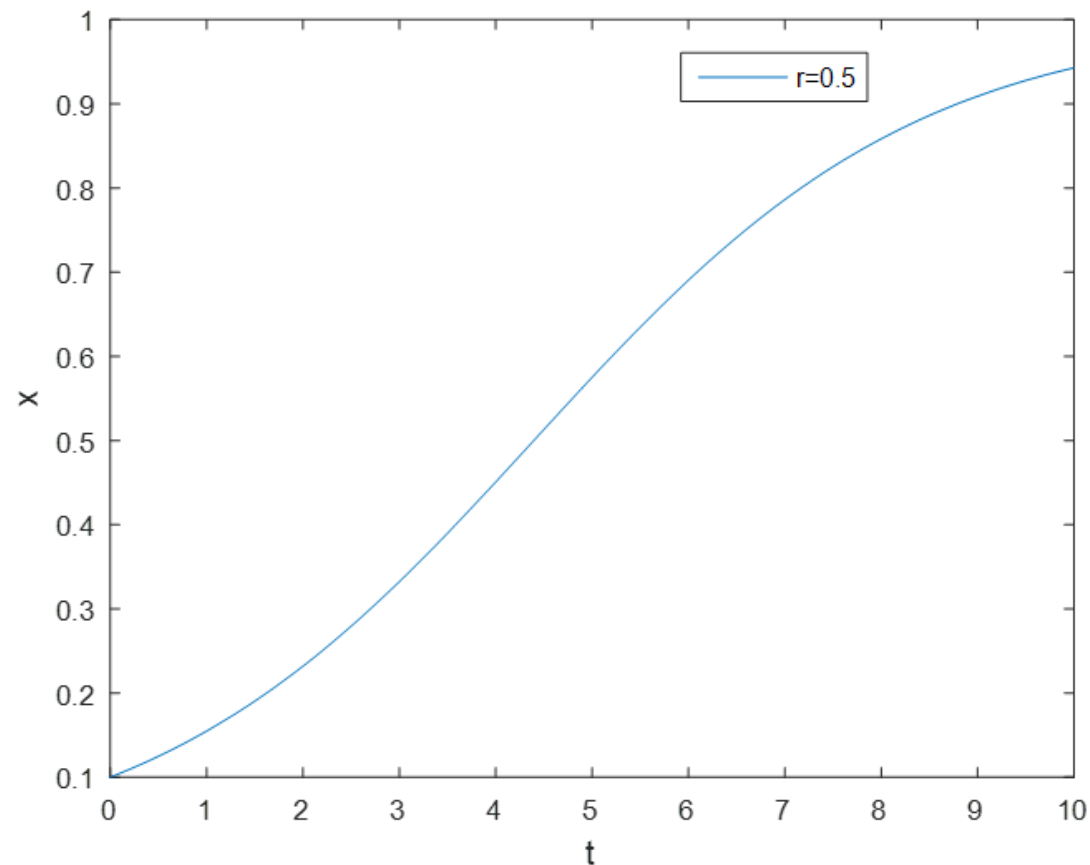
Script Example

>> logistic_script

- Can also specify inputs at command line

- Alternative logistic_script.m

x = 1./(1+(1/x0-1)*exp(-r*t));

plot(t,x)

>> x0 = 0.2;

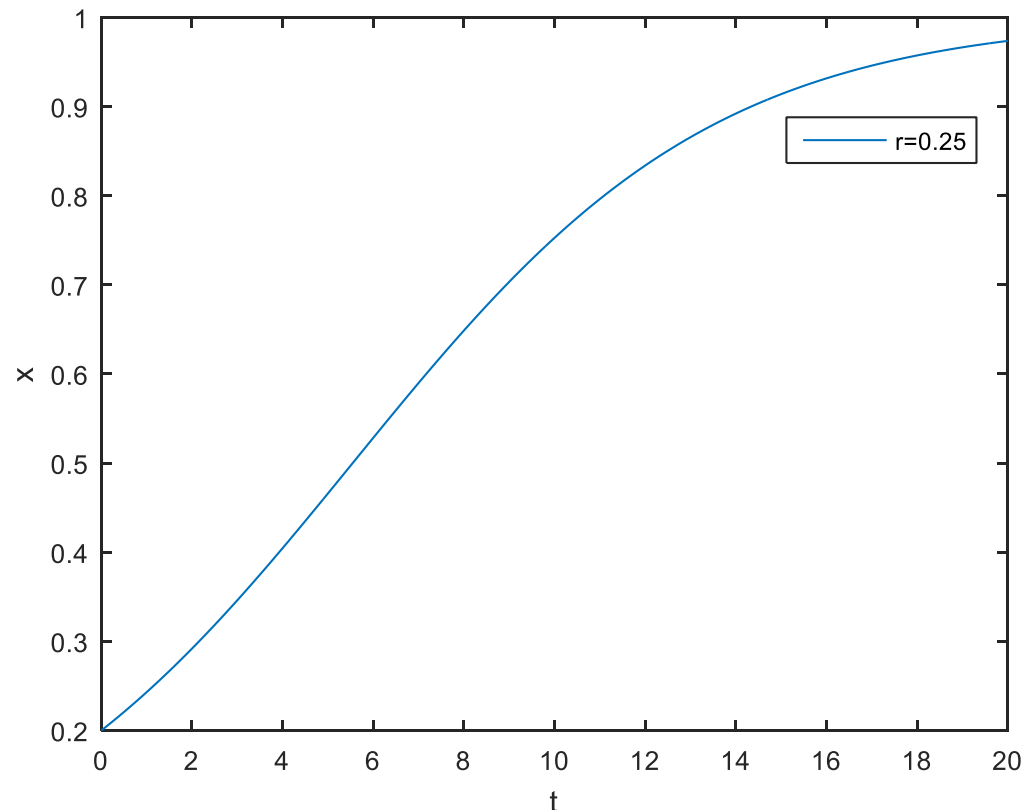>> r = 0.25;

>> t = [0:0.1:20];

>> logistic_script

- Need to add labeling commands

Functions

- Program files can be functions that accept input arguments and produce outputs

- Both scripts and functions contain MATLAB code, and both are stored in text files with a .m extension

- Functions are more flexible and more easily extensible than scripts

- Functions are created from the menu by choosing File → New → Function

- The function is written and then saved as filename.m

- The function is executed at the command line by providing the inputs are arguments

UMASS

Functions

function [y1,...,yN] = myfun(x1,...,xM)

- Declares a function named myfun that accepts inputs x1,...,xM and returns outputs y1,...,yN. This declaration statement must be the first executable line of the function

- Save the function code in a text file with a .m extension. The name of the file should match the name of the first function in the file

- Valid function names begin with an alphabetic character, and can contain letters, numbers, or underscores

- Files can include multiple local functions or nested functions

UMASS

Function Example

- Create a function with x_0, r and t as inputs and x as the output

$$x(t) = \frac{1}{1+\left(\dfrac{1}{x_0}-1\right)e^{-rt}}$$

- logistic.m

function [x] = logistic(t,r,x0)
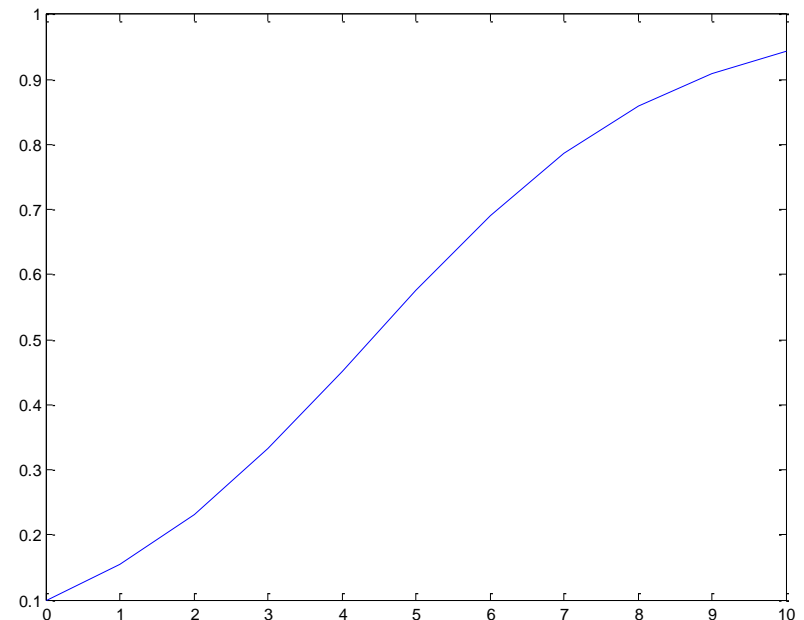
x = 1./(1+(1/x0-1).*exp(-r*t));

- Using the function

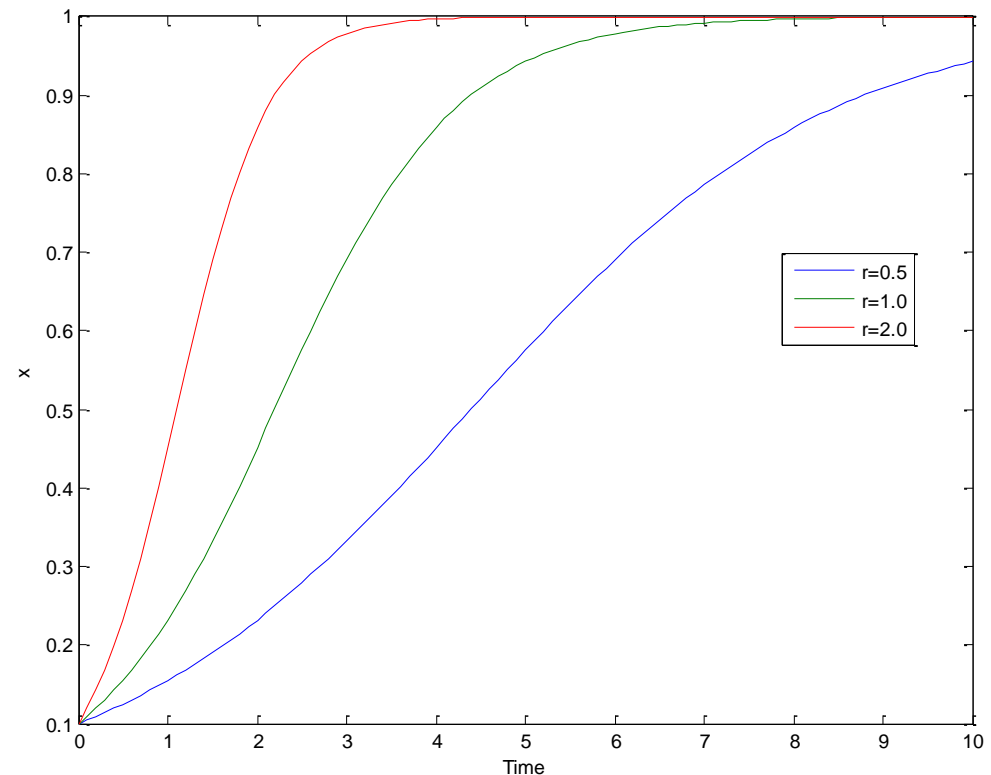>> t = [0:1:10];

>> r = 0.5;

>> x0 = 0.1;

>> x = logistic(t,r,x0);

>> plot(t,x)

Function Example

```
>> t = [0:0.1:10];
>> x0 = 0.1;
>> r1 = 0.5;
>> r2 = 1;
>> r3 = 2;
>> x1 = logistic(t,r1,x0);
>> x2 = logistic(t,r2,x0);
>> x3 = logistic(t,r3,x0);
>> plot(t,x1,t,x2,t,x3)
>> xlabel('Time')
>> ylabel('x')
>> legend('r=0.5','r=1.0','r=2.0')
```

MATLAB: Scripts and Functions

In-class Exercise

Solve Quadratic Equation

- Write a function quadratic to find the roots of any second-order polynomial

$$ax^2 + bx + c = 0 \implies x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

- Find the roots of the following polynomials:

$$x^2 + 3x + 2 = 0$$

$$4x^2 + 4x + 1 = 0$$

$$x^2 + x + 1 = 0$$