|  |  |
| --- | --- |
| **Name** |  |
| **Reg. #** |  |
| **Marks** |  |

# Experiment #2 Polynomials in MATLAB

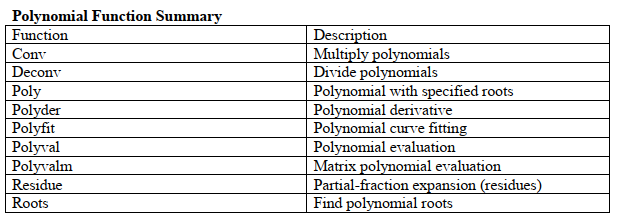
:

**Introduction:**

Objective: The objective of this session is to learn how to represent polynomials in  
MATLAB, find roots of polynomials, create polynomials when roots are known and obtain  
partial fractions.

**Polynomial Overview:**

MATLAB provides functions for standard polynomial operations, such as polynomial roots, evaluation, and differentiation. In addition, there are functions for more advanced applications, such as curve fitting and partial fraction expansion.



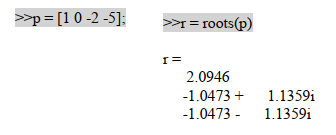
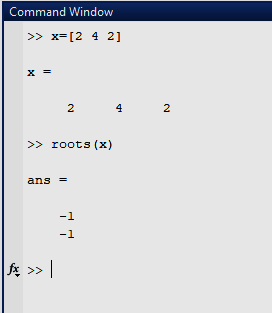
MATLAB represents polynomials as row vectors containing coefficients ordered by descending powers.  
For example: p(x)= ax2 + bx + c is represented by p=[a b c] in MATLAB.

**Objective:**

Using MATLAB to solve polynomials and partial Fraction.

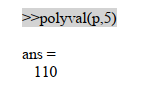
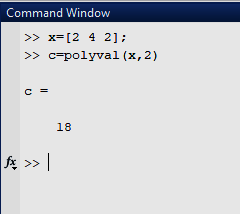
**Polynomial ROOTS:**

Roots are calculated by roots(g) command, where g is a polynomial equation



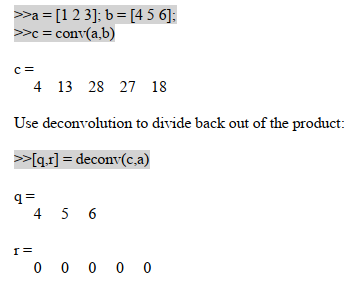
* **Polynomial Evaluation:**

It is used to find the value of a polynomial value at a specific. Point by command polyval(s,p), where p is a number and s polynomial as shown below.



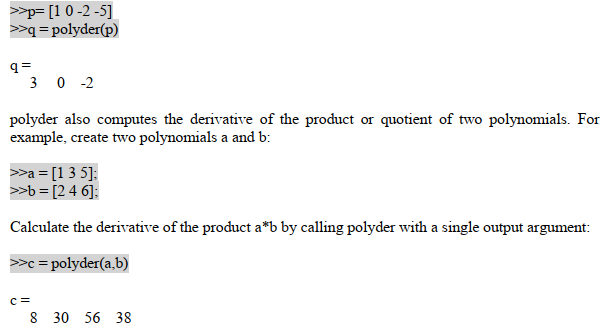
**Convolution and Deconvolution:**

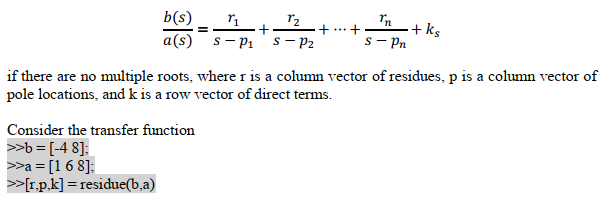
Polynomial multiplication and division correspond to the operations convolution and deconvolution. The functions conv and deconv implement these operations. Consider the polynomials a()= and b(s)= . To compute their product,

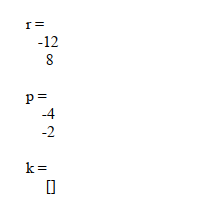


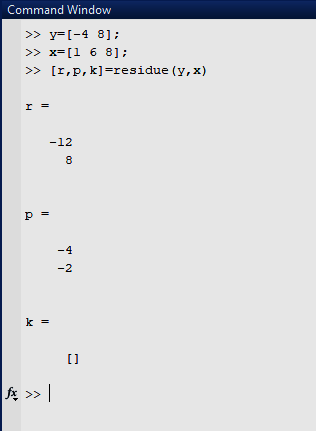
**Polynomial Derivatives:**

The polyder function computes the derivative of any polynomial. To obtain the derivative of  
the polynomial

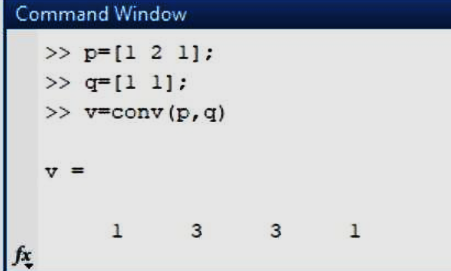
**  
Partial Fraction Expansion**‘residue’ finds the partial fraction expansion of the ratio of two polynomials. This is particularly useful for applications that represent systems in transfer function form. For polynomials b and a,



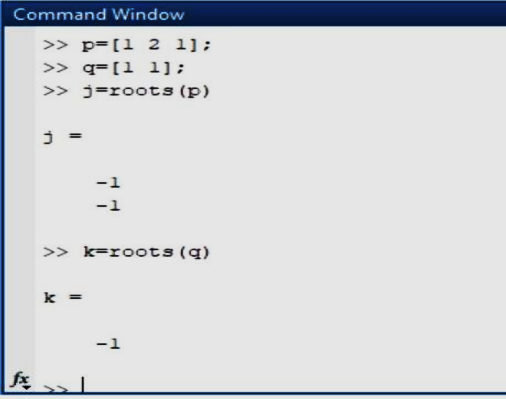


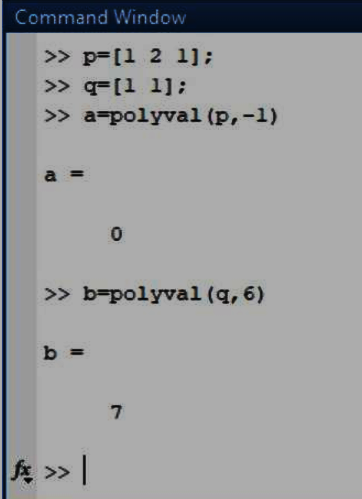
**Residue:**Residue is used for solving partial fraction in MATLAB. We use residue as [r,p,k]=residue(b,a).

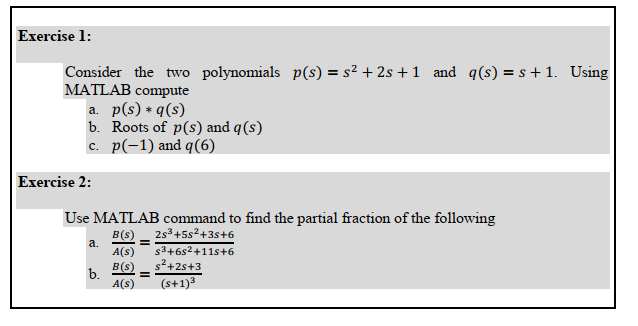
***Exercise 1:*** Consider polynomials p(s)= s2 +2s+1 & q(s)=s+1, Now:

***a) P(s)\*q(s):***  


***b) Roots of p(s) and q(s):***



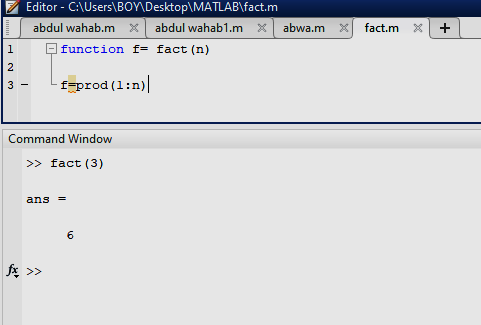




* ***Flow Control:***

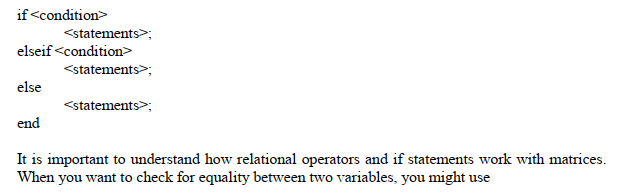
Flow control is the part of MATLAB which controls the if-else statements, for loop and switch statements.

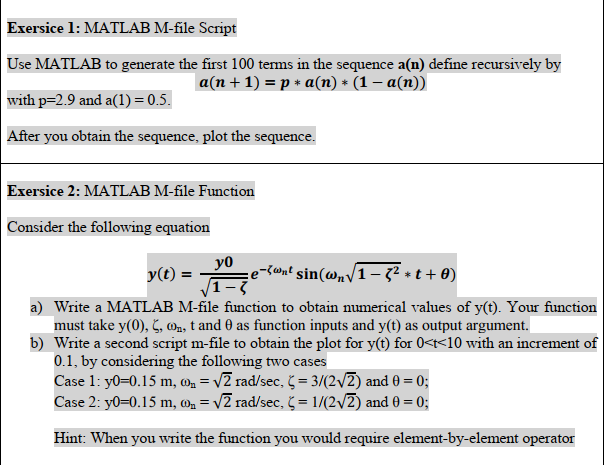
* ***Functions (m-Files):***

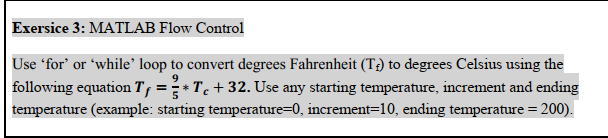
M.file is just like a command window in which we can write the input argument, the name of the M-file and the function should be the same. 

Functions are M-files that can accept input arguments and return output arguments. The names of the M-file and of the function should be the same. Functions operate on variables within their own workspace, separate from the workspace you access at the MATLAB command prompt. An example is provided below:

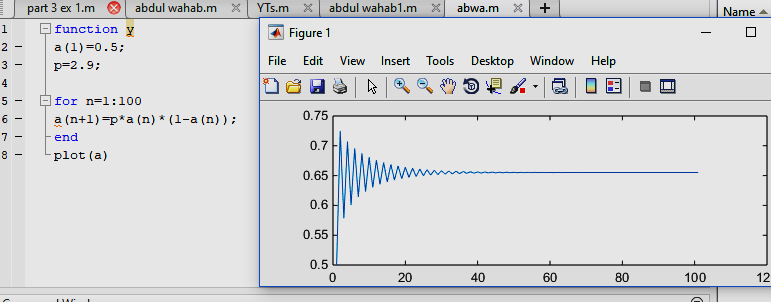
**Flow Control:  
Conditional Control – if, else, switch:**This section covers those MATLAB functions that provide conditional program control. if, else, and elseif. The if statement evaluates a logical expression and executes a group of statements when the expression is true. The optional elseif and else keywords provide for the execution of alternate groups of statements. An end keyword, which matches the if, terminates the last group of statements. The groups of statements are delineated by the four keywords—no braces or brackets are involved as given below.







***Exercise 1:*** M-file scripts, Use MATLAB for 100 terms of a(n+1)=p\*a(n)\*(a-a(n)) With p=2.9 & a=0.5 draw plot too.



**Fig : Program to generate 100 terms of a(n)**