Morteza Kazemi | 97243054

Amir Masoud Shaker | 97243081

11/21/2021

Homework #4

# Q1

|  |  |  |
| --- | --- | --- |
| Syntax | Meaning | Description |
| 1. polyval(p,x) 2. [y,delta] = polyval(p,x,S) 3. y = polyval(p,x,[],mu) or [y,delta] = polyval(p,x,S,mu) | Polynomial evaluation | 1. This syntax evaluates the polynomial p at each point in x.   The argument p is a vector of length n+1 and its elements are coefficients of an nth-degree polynomial.   1. This syntax uses the optional output structure S to generate error estimates. delta is an estimate of the standard error in predicting a future observation at x by p(x). 2. This syntax uses the optional output mu to center and scale the data.   mu(1) is mean(x), and mu(2) is std(x). |

Input Arguments:

p — Polynomial coefficients, specified as a vector. For example, the vector [3 0 5] represents the polynomial

Data Types: single | double + Complex Number Support

x — Query points, specified as a vector. polyval evaluates the polynomial p at the points in x and returns the corresponding function values in y.

Data Types: single | double + Complex Number Support

S — Error estimation structure. This structure is an optional output from [p,S] = polyfit(x,y,n) that can be used to obtain error estimates. S contains the following fields:

Fields:

R: Triangular factor from a QR decomposition of the Vandermonde matrix of x

Df: Degrees of freedom

Normr: Norm of the residuals

mu — Centering and scaling values, specified as a two-element vector. This vector is an optional output from [p,S,mu] = polyfit(x,y,n) that is used to improve the numerical properties of fitting and evaluating the polynomial p. The value mu(1) is mean(x), and mu(2) is std(x). These values are used to center the query points in x at zero with unit standard deviation.

Output Arguments:

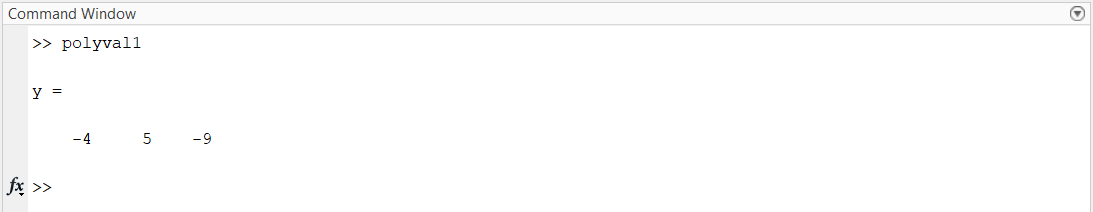
y — Function values, returned as a vector of the same size as the query points x. The vector contains the result of evaluating the polynomial p at each point in x.

delta — Standard error for prediction, returned as a vector of the same size as the query points x. Generally, an interval of y ± Δ corresponds to a roughly 68% prediction interval for future observations of large samples, and y ± 2Δ a roughly 95% prediction interval.

Examples:

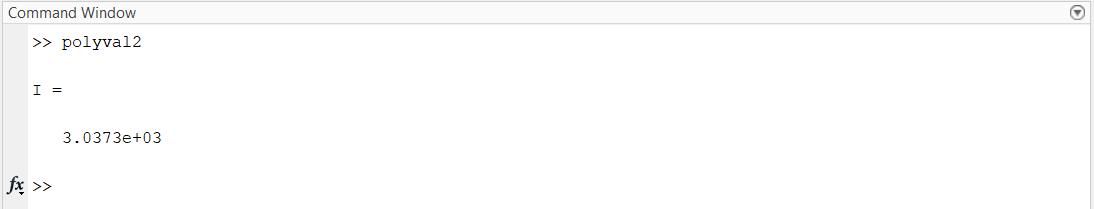
Polyval1:





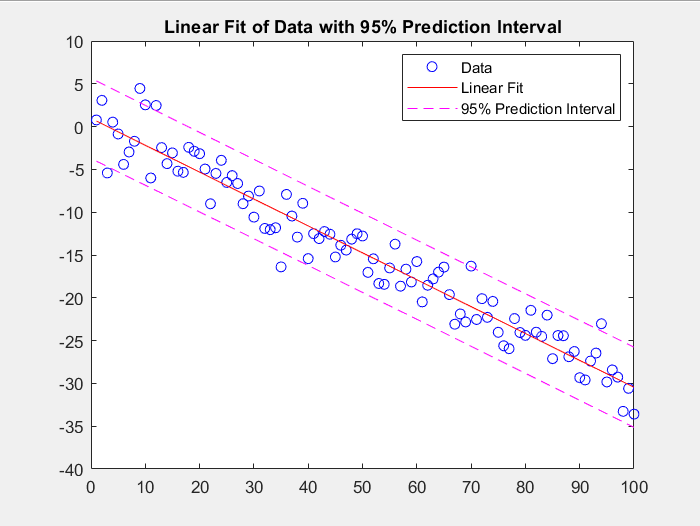
Polyval2:





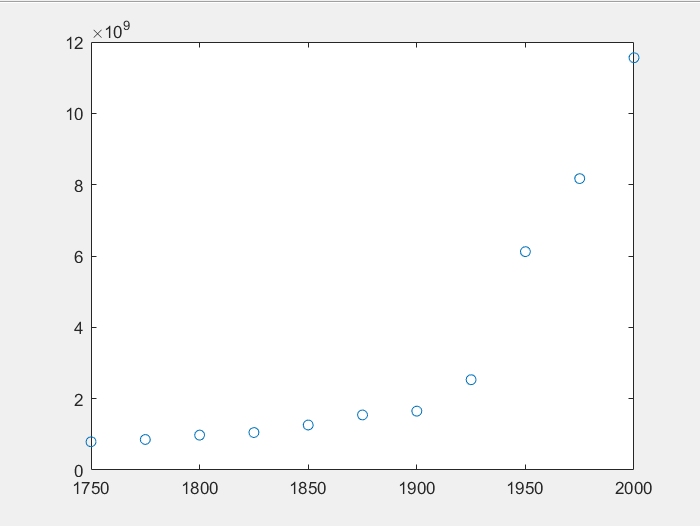
Polyval3:

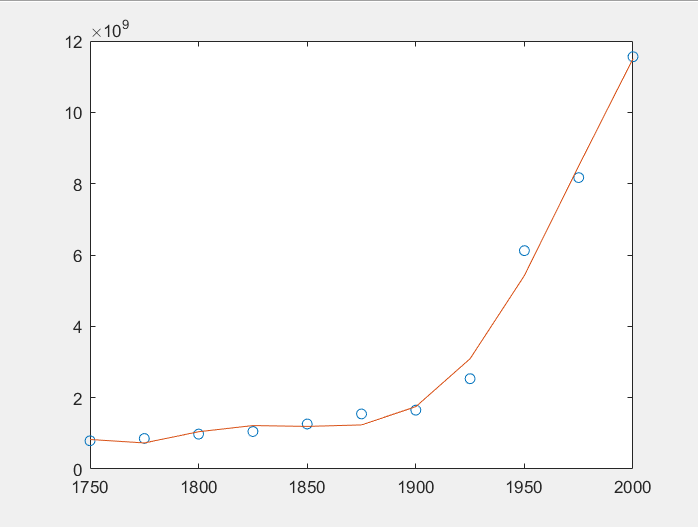




Polyval4:







|  |  |  |
| --- | --- | --- |
| Syntax | Meaning | Description |
| r = roots(p) | Polynomial roots | This syntax returns the roots of the polynomial represented by p as a column vector. |

Input Arguments:

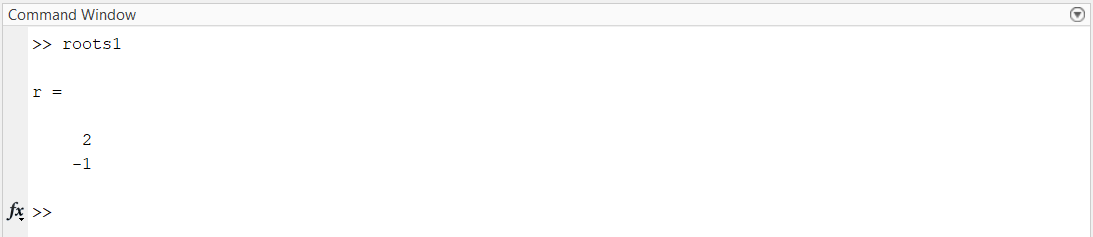
p — Polynomial coefficients, specified as a vector.

Data Types: single | double + Complex Number Support

Examples:

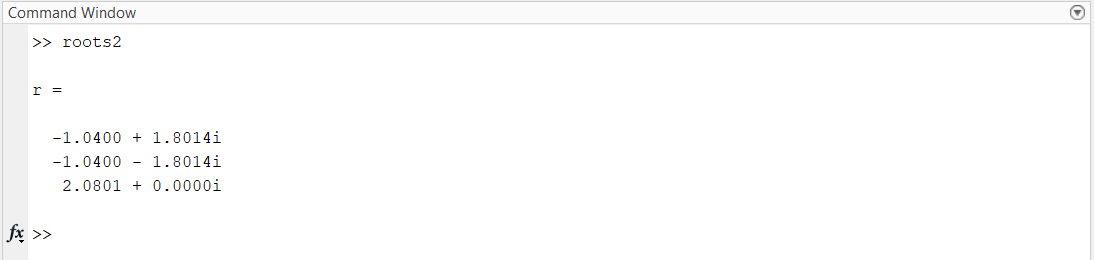
Roots1:





Roots2:





|  |  |  |
| --- | --- | --- |
| Syntax | Meaning | Description |
| 1. p = poly(r) 2. p = poly(A) | Polynomial with specified roots or characteristic polynomial | 1. This syntax returns the coefficients of the polynomial whose roots are the elements of r (r is a vector). 2. This syntax returns the n+1 coefficients of the characteristic polynomial of the matrix, det(λI – A) (A is a matrix). |

Input Arguments:

r — Polynomial roots, specified as a vector.

Data Types: single | double + Complex Number Support

A — Input matrix

Data Types: single | double + Complex Number Support

Output Arguments

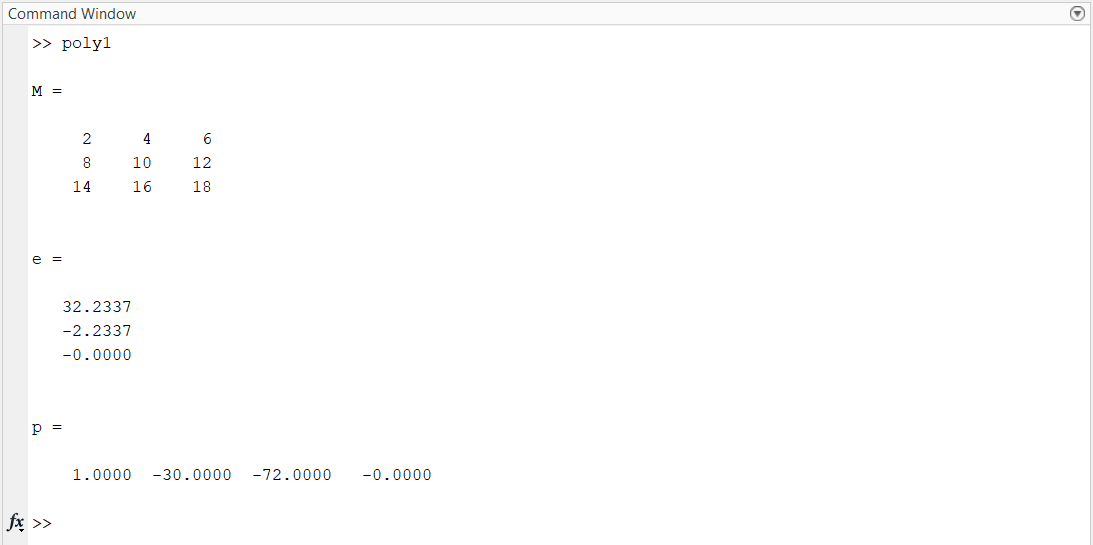
p — Polynomial coefficients, returned as a row vector.

If the input is a square n-by-n matrix, A, then p contains the coefficients for the characteristic polynomial of A. If the input is a vector of roots, r, then p contains the coefficients for the polynomial whose roots are in r.

Examples:

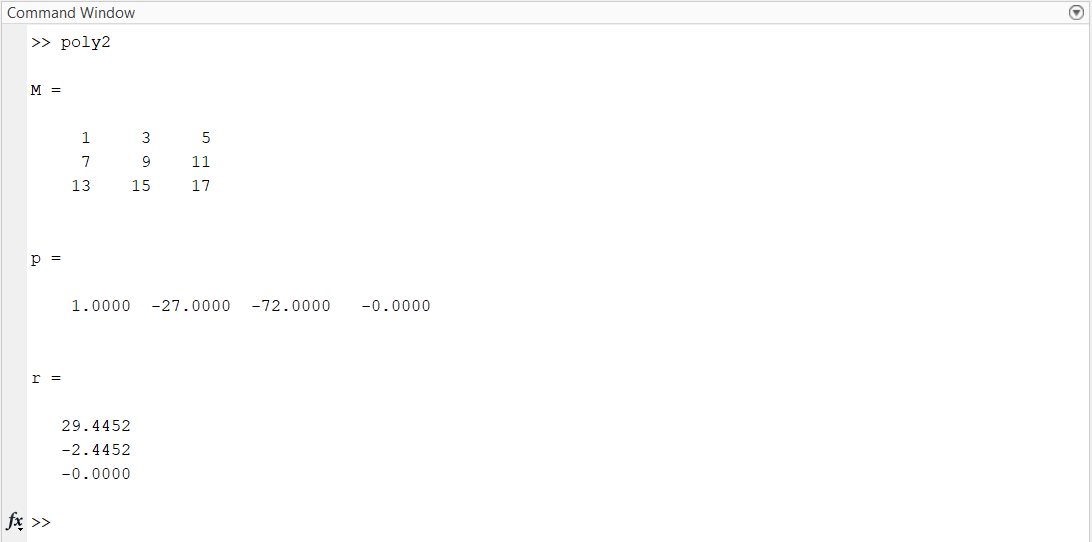
Poly1:





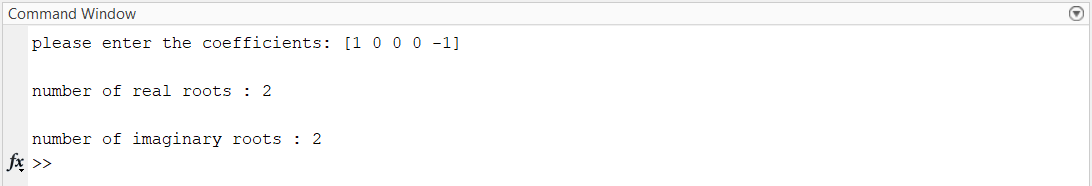
Poly2:





# Q2





# Q3

|  |  |  |
| --- | --- | --- |
| Syntax | Meaning | Description |
| 1. w = conv(u,v) 2. w = conv(u,v,shape) | Convolution and polynomial multiplication | 1. This syntax returns the convolution of vectors u and v.   If u and v are vectors of polynomial coefficients, this function will actually multiply them together.   1. This syntax returns a subsection of the convolution, as specified by shape.   conv(u,v,'same') returns only the central part of the convolution, the same size as u.  conv(u,v,'valid') returns only the part of the convolution computed without the zero-padded edges. |

Input Arguments:

u,v — Input vectors, specified as either row or column vectors. The vectors u and v can be different lengths or data types.

When u or v are of type single, then the output is of type single. Otherwise, conv converts inputs to type double and returns type double.

Data Types: double | single | int8 | int16 | int32 | int64 | uint8 | uint16 | uint32 | uint64 | logical

+ Complex Number Support

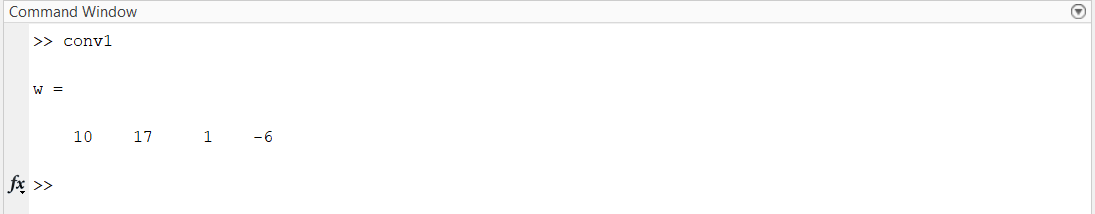
shape — Subsection of the convolution, specified as 'full', 'same', or 'valid'.

* 'full' : Full convolution (default).
* 'same': Central part of the convolution of the same size as u.
* 'valid': Only those parts of the convolution that are computed without the zero-padded edges. Using this option, length(w) is max(length(u)-length(v)+1,0), except when length(v) is zero. If length(v) = 0, then length(w) = length(u).

Examples:

Conv1:





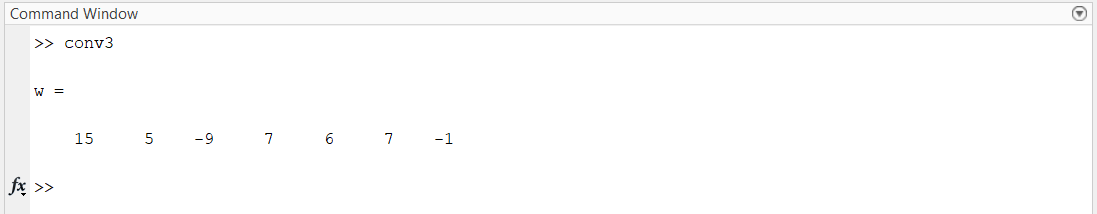
Conv2:





Conv3:





|  |  |  |
| --- | --- | --- |
| Syntax | Meaning | Description |
| [q,r] = deconv(u,v) | Deconvolution and polynomial division | This syntax deconvolves a vector v out of a vector u using long division, and returns the quotient q and remainder r.  If u and v are vectors of polynomial coefficients, this function will actually divide u by v. |

Input Arguments:

u,v — Input vectors, specified as either row or column vectors. u and v can be different lengths or data types.

If one or both of u and v are of type single, then the output is also of type single. Otherwise, deconv returns type double.

The lengths of the inputs should generally satisfy length(v) <= length(u). However, if length(v) > length(u), then deconv returns the outputs as q = 0 and r = u.

Data Types: double | single + Complex Number Support

Output Arguments:

q — Quotient, returned as a row or column vector such that u = conv(v,q)+r.

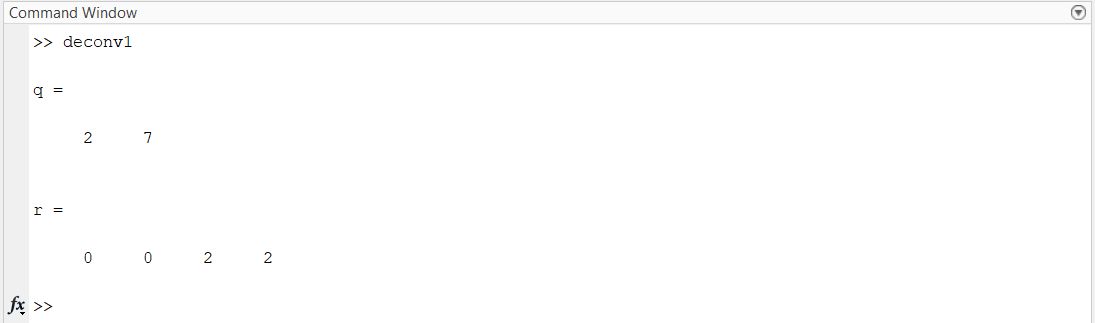
r — Remainder, returned as a row or column vector such that u = conv(v,q)+r.

Data Types: double | single

Examples:

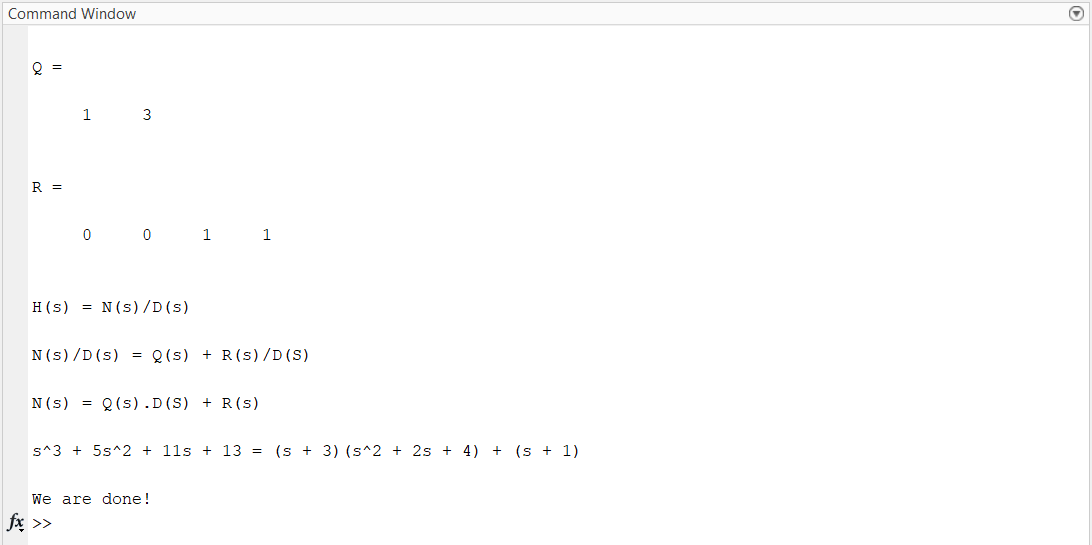
Deconv1:





# Q4





# Q5

|  |  |  |  |
| --- | --- | --- | --- |
| Syntax | Meaning | | Description |
| 1. k = polyder(p) 2. k = polyder(a,b) 3. [q,d] = polyder(a,b) | Polynomial differentiation | 1. returns the derivative of the polynomial represented by the coefficients in p 2. returns the derivative of the product of the polynomials a and b 3. returns the derivative of the quotient of the polynomials a and b | |

Input Arguments:

p — Polynomial coefficients (vector)

For example, the vector [3 0 5] represents the polynomial

a,b — Polynomial coefficients (specified as two separate arguments of row vectors.)

Data Types: single | double + Complex Number Support

Output Arguments:

k — Differentiated polynomial coefficients (row vector)

q — Numerator polynomial (row vector)

d — Denominator polynomial (row vector)

Examples:





|  |  |  |  |
| --- | --- | --- | --- |
| Syntax | Meaning | | Description |
| 1. q = polyint(p,k) 2. q = polyint(p) | Polynomial integration | 1. returns the integral of the polynomial represented by the coefficients in p using a constant of integration k. 2. assumes a constant of integration k = 0. | |

Input Arguments:

p — Polynomial coefficients, specified as a vector.

k — Constant of integration, specified as a numeric scalar.

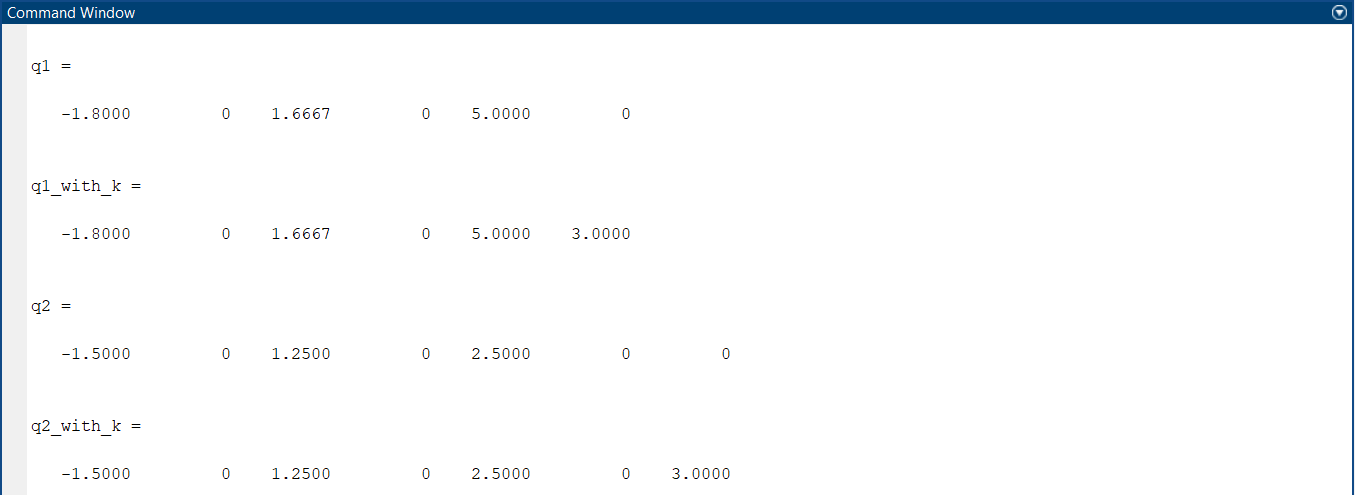
Data Types: single | double + Complex Number Support

Output Arguments:

q — Integrated polynomial coefficients, returned as a row vector.

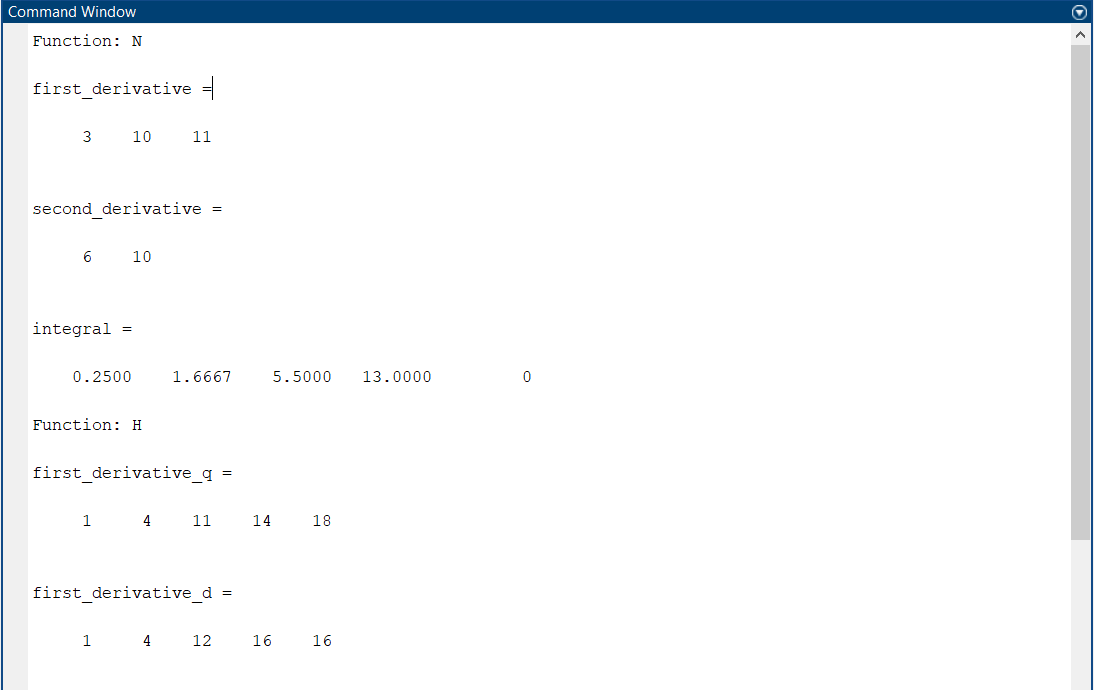
Examples:

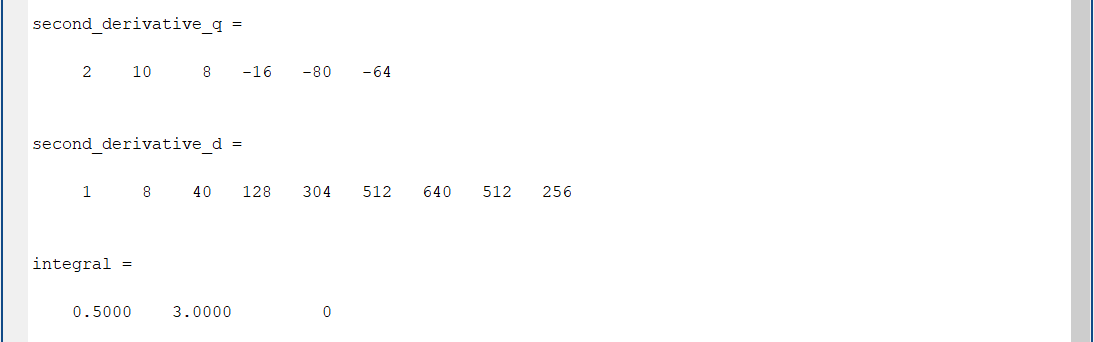




Calculations for N(s) and H(s):





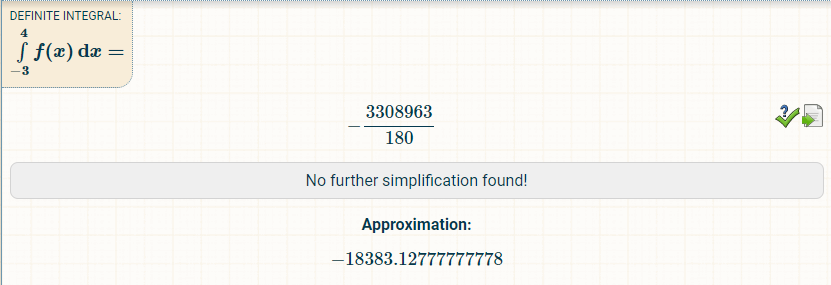
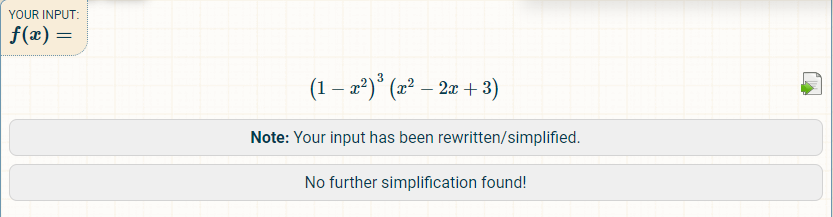


# Q6





We checked the answer here: <https://www.integral-calculator.com/>



# Q7

|  |  |  |  |
| --- | --- | --- | --- |
| Syntax | Meaning | | Description |
| 1. [r,p,k] = residue(b,a) 2. [b,a] = residue(r,p,k) | Partial fraction expansion (decomposition) | 1. finds the residues, poles, and direct term of a Partial Fraction Expansion of the ratio of two polynomials, where the expansion is:   The inputs to residue are vectors of coefficients of the polynomials b = [bm ... b1 b0] and a = [an ... a1 a0]. The outputs are the residues r = [rn ... r2 r1], the poles p = [pn ... p2 p1], and the polynomial k. For most textbook problems, k is 0 or a constant.   1. converts the partial fraction expansion back to the ratio of two polynomials and returns the coefficients in b and a. | |

Notes:

If the degree of the numerator is equal to the degree of the denominator, the output k can be nonzero and if the degree of the numerator is greater than the degree of the denominator, the output k is a vector that represents the coefficients of a polynomial in s.

Input Arguments:

b/a — Coefficients of numerator/denominator polynomial (vector of numbers)

Specified as a vector of numbers representing the coefficients of the polynomial in descending powers of s.

Data Types: single | double + Complex Number Support

Output Arguments:

r/p — Residues/Poles/ Direct term of partial fraction expansion (r and p: column vector of numbers that are specified in the table above, k: row vector of numbers that specify the coefficients of the polynomial in descending powers of s.)

Examples:



|  |  |  |
| --- | --- | --- |
|  |  |  |

Expanding Y(s):



