Documentation MATLAB MATLAB Functions

MATLAB Function Reference poly

Polynomial with specified roots

Syntax

p = poly(A)p = poly(r)

Description



p = poly(A) where A is an n-by-n matrix returns an n+1 element row vector whose elements are the coefficients of the characteristic polynomial, det(sl - A). The coefficients are ordered in descending powers: if a vector c has n+1 components, the polynomial it represents is $c_1s^n + \ldots + c_ns + c_{n+1}$

p = poly(r) where r is a vector returns a row vector whose elements are the coefficients of the polynomial whose roots are the elements of r.

Remarks

Note the relationship of this command to

r = roots(p)

which returns a column vector whose elements are the roots of the polynomial specified by the coefficients row vector p. For vectors, roots and poly are inverse functions of each other, up to ordering, scaling, and roundoff error.

Examples

MATLAB displays polynomials as row vectors containing the coefficients ordered by descending powers. The characteristic equation of the matrix

A = 1 2 3 4 5 б 7 8 Ω

is returned in a row vector by poly:

p = poly(A)p = 1 -6 -72 -27

The roots of this polynomial (eigenvalues of matrix A) are returned in a column vector by roots:

r = roots(p)r = 12.1229 -5.7345 -0.3884

Algorithm

The algorithms employed for poly and roots illustrate an interesting aspect of the modern approach to eigenvalue

computation. poly(A) generates the characteristic polynomial of A, and roots(poly(A)) finds the roots of that polynomial, which are the eigenvalues of A. But both poly and roots use eig, which is based on similarity transformations. The classical approach, which characterizes eigenvalues as roots of the characteristic polynomial, is actually reversed.

If A is an n-by-n matrix, poly(A) produces the coefficients c(1) through c(n+1), with c(1) = 1, in

$$det(\lambda I - A) = c_1 \lambda^n + \dots + c_n \lambda + c_{n+1}$$

The algorithm is

This recursion is easily derived by expanding the product.

$$(\lambda - \lambda_1)(\lambda - \lambda_2)...(\lambda - \lambda_n)$$

It is possible to prove that poly(A) produces the coefficients in the characteristic polynomial of a matrix within roundoff error of A. This is true even if the eigenvalues of A are badly conditioned. The traditional algorithms for obtaining the characteristic polynomial, which do not use the eigenvalues, do not have such satisfactory numerical properties.

See Also

conv, polyval, residue, roots

🔶 polar

polyarea 🕨

© 1994-2005 The MathWorks, Inc. - Trademarks - Privacy Policy