# Vincent Purcell - MATH323 - Honors Option - Fadeev Laverrier Algorithm

## Contents

## Example 1 of Fadeev-Laverrier Function

Test fadeev-laverrier algorithm of 2x2 matrix

```
test1 = [6 -1;2 3];
[coeff, inv] = fadeevLaverrier(test1);

% output results of Fadeev Laverrier algorithm
outputResults(test1, inv, 1, coeff);
```

## Example 2 of Fadeev-Laverrier Function

Test fadeev-laverrier algorithm of 3x3 matrix

```
test2 = [  3   -5    5;
           2  -10    7;
          -1   20   11];

[coeff2, inv2] = fadeevLaverrier(test2);

% output results of Fadeev Laverrier algorithm
outputResults(test2, inv2, 2, coeff2);
```

## Example 3 of Fadeev-Laverrier Function

Test fadeev-laverrier algorithm of 4x4 matrix

```
test3 = [ 2  5  6   7;
          6  7 -10  6;
          2 -4  2  -1;
         -2 -2  20  5 ];

[coeff3, inv3] = fadeevLaverrier(test3);

% output results of Fadeev Laverrier algorithm
outputResults(test3, inv3, 3, coeff3);
```

## Example 4 of Fadeev-Laverrier Function

Test fadeev-laverrier algorithm of 7x7 Magic Matrix

```
test4 = magic(7);
[coeff4, inv4] = fadeevLaverrier(test4);

% output results of Fadeev Laverrier algorithm
outputResults(test4, inv4, 4, coeff4);
```

## Fadeev-Laverrier Function

This function takes an input matrix A and outputs the coefficients of the characteristic polynomial. Also with the final increment of the Eigenvalue diagonal, matrix B in the below function, you can calculate the inverse of A without any extra computational power.

```
function [coeff,inv] = fadeevLaverrier(A)
    %
    % fadeevLaverrier
    % Function to generate characteristic polynomial of a given MATRIX A
    % as well as the inverse of A without extra computational power.
    %
    [n,~]=size(A);
    coeff = ones(1,n+1);

    LF_mat = A;
    for i = 2:n
        % copy of matrix saved to calculate inverse after coefficient
        % polynomial is found
        inv_mat = LF_mat;
        % Take negative sum of diagonal of LF_mat divided by n-1
        % to get coeff of increment i
        coeff(i) = -trace(LF_mat)/(i-1);
        % Calculate new LF_mat by adding coefficient to diagonal of LF_mat
        % and then multiplying it the original matrix A
        LF_mat = A*(LF_mat+coeff(i)*eye(n));
    end
    % Take negative sum of diagonal of LF_mat divided by n to get final
    % coefficient
    coeff(n+1) = -trace(LF_mat)/n;
    % Get Inverse of original function at no extra cost to computation time
    inv=-(inv_mat+coeff(n)*eye(n))/coeff(n+1);
end
```

## Function to Display Polynomial from Coefficients

Takes a input coefficient vector and returns a string that can display the function for the Matlab Publisher

```
function [poly_string] = dispPolynomial(vec)
    lambda_num = size(vec,2)-1;
    poly_string = "\x03bb^" + num2str(lambda_num);
    for i = 2:size(vec,2)
       lambda_num = size(vec,2)-i;
```

```
        poly_string = poly_string + " + " + num2str(vec(i)) + "\x03bb^" + num2str(lambda_num);
    end
end
```

```
*** Example1 ***

Test Matrix 1:
     6    -1
     2     3

Coefficient Vector:
     1    -9    20

Polynomial of Matrix:
λ^2 + -9λ^1 + 20λ^0

Inverse of Matrix:
    0.1500    0.0500
   -0.1000    0.3000




*** Example2 ***

Test Matrix 2:
     3    -5     5
     2   -10     7
    -1    20    11

Coefficient Vector:
     1    -4  -232   455

Polynomial of Matrix:
λ^3 + -4λ^2 + -232λ^1 + 455λ^0

Inverse of Matrix:
    0.5495   -0.3407   -0.0330
    0.0637   -0.0835    0.0242
   -0.0659    0.1209    0.0440




*** Example3 ***

Test Matrix 3:
     2     5     6     7
     6     7   -10     6
     2    -4     2    -1
    -2    -2    20     5

Coefficient Vector:
     1   -16    51   688  -604

Polynomial of Matrix:
λ^4 + -16λ^3 + 51λ^2 + 688λ^1 + -604λ^0

Inverse of Matrix:
    0.7715   -0.4139    0.4834   -0.4868
    0.8675   -0.5298    0.1788   -0.5430
    0.4305   -0.2781    0.1689   -0.2351
   -1.0662    0.7351   -0.4106    0.7285




*** Example4 ***

Test Matrix 4:
    30    39    48     1    10    19    28
    38    47     7     9    18    27    29
    46     6     8    17    26    35    37
     5    14    16    25    34    36    45
    13    15    24    33    42    44     4
    21    23    32    41    43     3    12
    22    31    40    49     2    11    20

Coefficient Vector:
   1.0e+11 *

  Columns 1 through 7

    0.0000   -0.0000   -0.0000    0.0000    0.0001   -0.0101   -0.0199

  Column 8

    3.4805

Polynomial of Matrix:
λ^7 + -175λ^6 + -4802λ^5 + 840350λ^4 + 5764801λ^3 + -1008840175λ^2 + -1988873152λ^1 + 3480528
01600λ^0

Inverse of Matrix:
    0.0008    0.0008    0.0212   -0.0195   -0.0021    0.0041    0.0004
   -0.0021    0.0241   -0.0195    0.0012    0.0004    0.0008    0.0008
    0.0212   -0.0191    0.0004   -0.0021    0.0037    0.0008    0.0008
   -0.0170    0.0008    0.0008    0.0008    0.0008    0.0008    0.0187
    0.0008    0.0008   -0.0021    0.0037    0.0012    0.0207   -0.0195
    0.0008    0.0008    0.0012    0.0004    0.0212   -0.0224    0.0037
    0.0012   -0.0025    0.0037    0.0212   -0.0195    0.0008    0.0008
```

## Function to Output Results

```
function outputResults(mat,inv,experiment_num,coeff)
    poly_string = dispPolynomial(coeff);
    fprintf("\n\n*** Example" + num2str(experiment_num) + " ***\n\n");
    fprintf("Test Matrix " + num2str(experiment_num) + ":\n");
    disp(mat);
    fprintf('Coefficient Vector:\n');
    disp(coeff);
    fprintf('Polynomial of Matrix:\n');
    fprintf(poly_string);
    fprintf('\n\nInverse of Matrix:\n');
    disp(inv);
end
```

*Published with MATLAB® R2019b*