ENME 303 LAB

Week 13: Eigenvalue & Eigenvectors

Nameless Lab

Useful Functions

- Using `expand(S)` multiplies all parentheses in S, and simplifies

Ex.

```
clc; clear
syms x a
%%
f1 = (x-a)^3;
e = expand(f1)
```

Output:

```
e = - a^3 + 3*a^2*x - 3*a*x^2 + x^3
```

- Using `factor()` returns a row vector containing the prime factors

Ex.

```
clc; clear
syms x a
%%
f2 = x^3 + 4*x^2 - 11*x - 30;
f = factor(f2)
```

Output:

```
f = [x + 5, x - 3, x + 2]
```

Useful Functions

- Using `simplify()` performs algebraic simplification of the expression

Ex.

```
clc; clear
syms x a
%%
f3 = -2*tan(x)/(tan(x)^2 -1);
s = simplify(f3)
```

Output:

```
s = tan(2*x)
```

- Using `collect()` collects coefficients of the eqn

Ex.

```
clc; clear
syms x a
%%
f4 = (x + x^2)*(x +1)*x;
c = collect(f4)
```

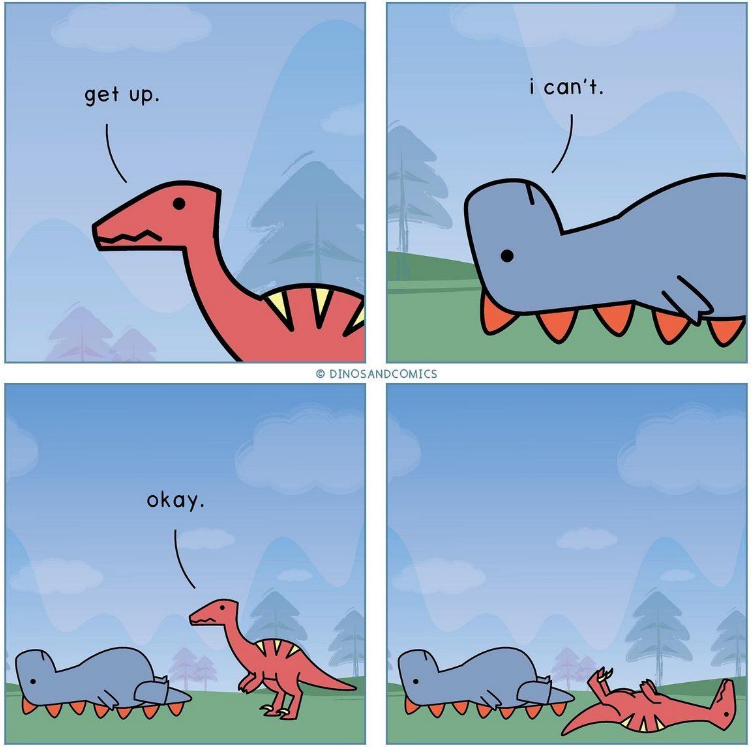
Output:

```
c = x^4 + 2*x^3 + x^2
```

Week 14: ⟶

I. Solving for Eigenvalues/Eigenfunctions in Matlab- one way:
   - **A. Eigenvalue/Eigenvector Function**

II. Solving for Eigenvalues/Eigenfunctions in Matlab- second way:
   - **A. Characteristic Polynomial Function**
   - **B. Polynomial Roots Function**



UMBC

Eigenvalues & Eigenvectors

1. **Solving for eigenvalue**

$$det(A - \lambda I) = 0$$

$$det(\begin{bmatrix} 3-\lambda & 1 \\ 0 & 2-\lambda \end{bmatrix}) = 0$$

$$det(\begin{bmatrix} 3-\lambda & 1 \\ 0 & 2-\lambda \end{bmatrix}) = (\lambda - 3)(\lambda - 2) = 0$$

$$\lambda = 3, 2$$

Check lecture slides for the definition of eigenvalues and eigenvectors

Characteristic Polynomial!

1. **Plugging in each eigenvalue to find set of eigenvectors**

$$\boxed{(A - \lambda I)\vec{v} = 0}$$

$$\lambda = 2:$$

$$\begin{bmatrix} 3-2 & 1 \\ 0 & 2-2 \end{bmatrix} \begin{bmatrix} \vec{x} \\ \vec{y} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Way #1 to Solve for Eigenvalue/Eigenvectors in Matlab

In Matlab, we can calculate the eigenvalues of a matrix, A, using:

eig(A)

It returns a column vector with all the eigenvalues of the matrix

```
%% Example 1: Eigenvalue
A=[3 1; 0 2];
Eigenvalues=eig(A);
fprintf('These are the eigenvalues of matrix A:\n');
disp(Eigenvalues)
```

```
These are the eigenvalues of matrix A:
     3
     2
```

Way #1 to Solve for Eigenvalue/Eigenvectors

We can use a variation of the same function:

Remember that a function can have more than one output!

[V, D] = eig(A)

To obtain a diagonal matrix D with eigenvalues and vector V containing the eigenvectors of matrix A.

```
%% Example 2: Eigenvalue + Eigenvectors
A=[3 1; 0 2];
[V,D] = eig (A);
eigvals=diag(D);
```

```
V =

    1.0000   -0.7071
         0    0.7071


D =

     3     0
     0     2
```

```
eigvals =

     3
     2
```

Way #2 to Solve for Eigenvalue/Eigenvectors In Matlab

Recall how eigenvalues are solved by hand:

$$det(A - \lambda I) = 0$$

$$det(\begin{bmatrix} 3-\lambda & 1 \\ 0 & 2-\lambda \end{bmatrix}) = 0$$

$$det(\begin{bmatrix} 3-\lambda & 1 \\ 0 & 2-\lambda \end{bmatrix}) = \boxed{(\lambda - 3)(\lambda - 2) = 0}$$

$$\boxed{\lambda = 3, 2}$$

Steps:

1. Determine your **characteristic polynomial**
2. Solve for the roots of that polynomial

Way #2 to Solve for Eigenvalue/Eigenvectors- Step 1

We can determine the characteristics polynomial in Matlab using:

[charpoly()]

It returns a vector of the coefficients of the characteristic polynomial

```
%% Example 3: Characteristic Polynomial

%Returns the COEFFICIENTS of your characteristic polynomial
A=[3 1; 0 2];
coeff=charpoly(A);
fprintf('These are the coefficients of the characteristic polynomial:\n');
disp(coeff)
```

```
These are the coefficients of the characteristic polynomial:
     1    -5     6
```

Way #2 to Solve for Eigenvalue/Eigenvectors- Step 1

A more useful way to display the characteristic polynomial using charpoly() would be to utilize our symbolic toolbox and create a polynomial in terms of x

```
%% Example 4: Characteristic Polynomial
%Returns the  of your characteristic polynomial in terms of x
syms x
A=sym([3 1; 0 2]);
poly=charpoly(A,x);
fprintf('This is the characteristic polynomial in terms of x:\n');
disp(poly)
```

```
This is the characteristic polynomial in terms of x:
x^2 - 5*x + 6
```

Way #2 to Solve for Eigenvalue/Eigenvectors- Step 2

Next, we have to solve for the roots of the characteristic polynomial i.e. the eigenvalues.

In Matlab, we can use the roots() built-in to do this, but only for the case of determining the coefficients:

```
%% Example 5: Solving for Roots of Coefficients
A=[3 1; 0 2];
coeff=charpoly(A);
fprintf('These are the coefficients of the characteristic polynomial:\n');
disp(coeff)
r = roots(coeff);
fprintf('These are the eigenvalues:\n');
disp(r)
```

```
These are the coefficients of the characteristic polynomial:
     1    -5     6

These are the eigenvalues:
    3.0000
    2.0000
```

Way #2 to Solve for Eigenvalue/Eigenvectors- Step 2

If using the symbolic toolbox, you must use solve() to determine the roots of the characteristic polynomial.

```
%% Example 6: Solving for Roots of Characteristic Polynomial
%Returns the  of your characteristic polynomial in terms of x
syms x
A=sym([3 1; 0 2]);
poly=charpoly(A,x);
fprintf('This is the characteristic polynomial in terms of x:\n');
disp(poly)
eval=solve(poly);
fprintf('These are the eigenvalues:\n');
disp(eval)
```

```
This is the characteristic polynomial in terms of x:
x^2 - 5*x + 6

These are the eigenvalues:
2
3
```

Way #2 to Solve for Eigenvalue/Eigenvectors- Step 3 **(Optional)**

Once you have the eigenvalues, we can plug them back into our equation to solve for the eigenvectors. We can use the null() function to ensure that we only get nontrivial (nonzero) solutions.

$$(A - \lambda I)\vec{v} = 0$$

```
%% Example 7: Solving for the eigenvectors

A = [3 1; 0 2];

coeffs = charpoly(A);

lambda = roots(coeffs); %these are our eigenvalues

%calculate the nontrivial solutions
vec1 = null((A-lambda(1)*eye(2)));
vec2 = null((A-lambda(2)*eye(2)));

display(vec1)
display(vec2)
```

```
vec1 =

    1.0000
    0.0000


vec2 =

   -0.7071
    0.7071
```

Acknowledgement

The lab slides you see are not made by one person. All the TA/TFs served for this course have contributed their effort and time to the slides. Below are the leading TFs for each semester:

- 2021 FA - Karla Negrete (GTA)
- 2022 SP - Justin Grahovac
- 2022 FA - Kelli Boyer, Yisrael Wealcatch, Noelle Ray (GTA)
- 2023 SP - Mahamoudou Bah and Matt Moeller
- 2024 SP - Mohammad Riyaz Ur Rehman and Michael Mullaney