MATLAB – HW5 Upload Date: 4 Dec. 2020

Dorreen Rostami – 97243034
Zahra Hashemi – 97243072

6-1)

	Syntax	Description
assume	assume(condition)	states that condition is valid. When used, it automatically deletes all previous assumptions on the variables in condition.
	assume(expr,set)	states that expr belongs to set. assume deletes previous assumptions on variables in expr.
	assume(expr,'clear')	clears all assumptions on all variables in expr.

	Syntax	Description
assumptions	assumptions(var)	returns all assumptions that affect variable var. If var is an expression or function, returns all assumptions that affect all variables in var.
	assumptions	returns all assumptions that affect all variables in the workspace.

```
1 -    clc;clear;
2
3 -    syms x
4 -    assume(x/3,'integer')
5 -    solve(x>10,x<20,x)
6
7 -    assume(x,'positive')
8 -    assumeAlso(x<10)
9 -    assump = assumptions(x)
```

```
Command Window

ans =

 12
 15
 18


assump =

[ 0 < x, x < 10]
```

7- deletes last assumption

6-2)

	Syntax	Description
double(s)	double(s)	converts the symbolic value s to double precision. useful when a function does not accept symbolic values

```
clc;clear;

x = sym([1/10,2/10])
d = double(x)
```

```
Command Window

x =

[ 1/10, 1/5]


d =

    0.1000    0.2000
```

6-3)

	Syntax	Description
symvar	C = symvar(expr)	searches the expression, expr, for identifiers other than i, j, pi, inf, nan, eps, and common functions.

```
clc;clear;

syms x
f1 = cos(x) + i*sin(x);
symvar(f1)

syms y
f2 = exp(i*x);
symvar(f2)
```

```
Command Window

ans =

x


ans =

x
```

6-4)

	Syntax	Description
diff	Y = diff(X)	calculates differences between adjacent elements of X along the first array dimension.
	Y = diff(X,n)	applies the diff(X) operator recursively n times.
	Y = diff(X,n,dim)	returns the nth difference calculated along the dimension specified by dim.
	diff(F)	differentiates F with respect to the variable determined by symvar(F,1).
	diff(F,var)	differentiates F with respect to the variable var.
	diff(F,n)	computes the nth derivative of F with respect to the variable determined by symvar.
	diff(F,var,n)	computes the nth derivative of F with respect to the variable var.
	diff(F,var1,...,varN)	differentiates F with respect to the variables var1,...,varN.

So, Diff(f,n) computes the 3rd derivative of the function f.

The functions written above the double horizontal lines (the top 3 ones) are like differentiation in a discreet space and the ones below differentiate symbolic expressions or functions so they are in a continuous space.

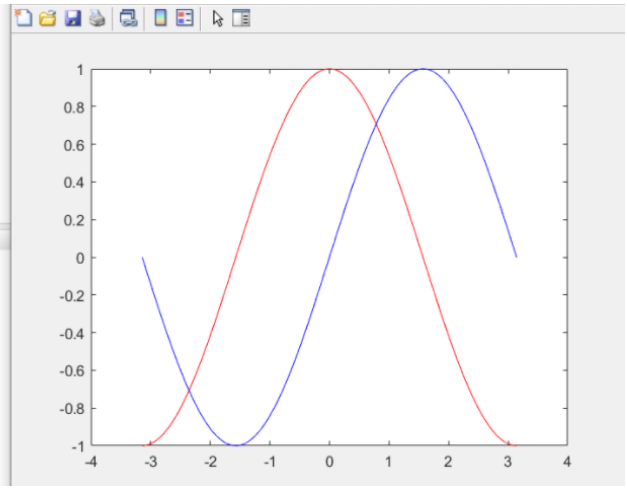
```
1 -  clc;clear;
2
3 -  X = [1 1 2 3 5 8 13 21];
4 -  Z = diff(X)
5
6 -  h = 0.001;
7 -  X = -pi:h:pi;
8 -  f = sin(X);
9 -  Y = diff(f)/h;
10 - plot(X(:,1:length(Y)),Y,'r',X,f,'b')
```

```
Command Window

  Z =

       0     1     1     2     3     5     8

fx >>
```



6-5)

```
clc;clear;

syms x y

z = x*y^3 + x^2;
d1 = diff(z, x)  % y^3 + 2x
d1 = diff(diff(z,x), x)  % 2
d3 = diff(z, y)  % 3xy^2
d4 = diff(diff(z,y), y)  % 6xy

w = exp(3*x*y) + cos(x/5) - sin(7*y)^2;
d5 = diff(w, x)  % 3y*exp(3xy) - (1/5)sin(x/5)
d6 = diff(diff(w,x), x)  % 9(y^2)*exp(3xy) - (1/25)cos(x/5)
d7 = diff(w, y)  % 3x*exp(3xy) - 14sin(7y)cos(7y)
d8 = diff(diff(w,y), y) % 98*sin(7*y)^2 - 98*cos(7*y)^2 + 9*x^2*exp(3*x*y)
```

Command Window

```
d1 =

y^3 + 2*x


d1 =

2


d3 =

3*x*y^2


d4 =

6*x*y


d5 =

3*y*exp(3*x*y) - sin(x/5)/5


d6 =

9*y^2*exp(3*x*y) - cos(x/5)/25


d7 =

3*x*exp(3*x*y) - 14*cos(7*y)*sin(7*y)


d8 =

98*sin(7*y)^2 - 98*cos(7*y)^2 + 9*x^2*exp(3*x*y)
```

6-6)

	Syntax	Description
int	int(expr,var)	computes the indefinite integral of expr with respect to the symbolic scalar variable var (optional, If not specified, int uses the default variable determined by symvar). If expr is a constant, then the default variable is x.
	int(expr,var,a,b)	computes the definite integral of expr with respect to var from a to b. *(same thing as above about var being optional and expr being a constant)*
	int(___,Name,Value)	specifies additional options using one or more Name,Value pair arguments.

```
1 -    clc;clear;
2
3 -    syms x
4
5 -    int1 = int(6*exp(-4*x), -3, 5)
6
7 -    int2 = int(exp(-x), 0, inf)
8
9 -    int3 = int(exp(-(x^2)), -inf, inf)
```

Command Window

```
int1 =

(3*exp(-20)*(exp(32) - 1))/2


int2 =

1


int3 =

pi^(1/2)
```

6-7)

	Syntax	Description
limit	limit(f,var,a)	returns the bidirectional limit of the symbolic expression f when var approaches a.
	limit(f,a)	uses the default variable found by symvar.
	limit(f)	returns the limit at 0.
	limit(f,var,a,'left')	returns the left-side limit of f as var approaches a.
	limit(f,var,a,'right')	returns the right-side limit of f as var approaches a.

```
1 -   clc;clear;
2
3 -   syms x
4 -   f = ((x^2 - 4)^2)/((x-2)*(x-3));
5 -   right2 = limit(f,x,2,'right')
6 -   left2  = limit(f,x,2,'left')
7 -   right3 = limit(f,x,3,'right')
8 -   left3  = limit(f,x,3,'left')
```

Command Window

```
right2 =

0


left2 =

0


right3 =

Inf


left3 =

-Inf
```

6-8)

	Syntax	Description
finverse	g = finverse(f)	returns the inverse of function f, such that f(g(x)) = x.
	g = finverse(f,var)	uses the symbolic variable var as the independent variable, such that f(g(var)) = var.

```
1 -    clc;clear;
2
3 -    syms x y
4 -    finverse(exp(x-y),x)
5
6 -    syms u v
7 -    finverse(exp(tan(u) + v), u)
```

Command Window

```
ans =

y + log(x)


ans =

-atan(v - log(u))
```