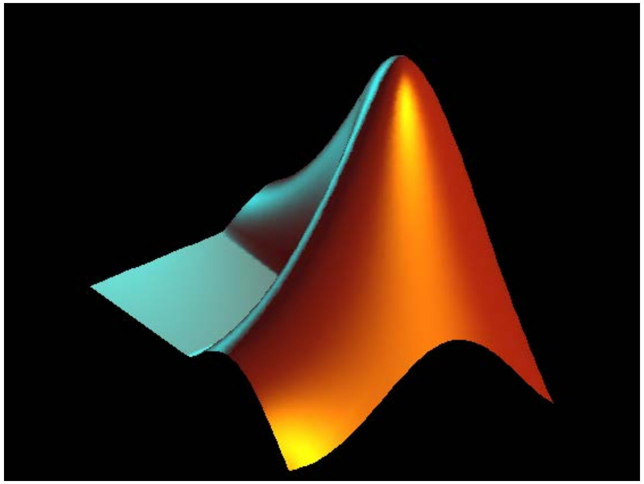
Computational Mathematics with MATLAB

Topic 5



Symbolic Math Toolbox

outline

- symbolic data type
- defining symbolic numbers, variables and functions
- manipulating algebraic expressions
- plotting with ezplot and ezsurf
- polynomial division example (quorem)
- partial fraction decomposition example (partfrac)
- differentiation and integration of symbolic functions
- integration problem (to be solved in class/at home)
- list of useful symbolic functions

introduction

The **Symbolic Math Toolbox** allows users to perform exact mathematical calculations with symbolic variables, e.g.

- expanding and simplifying algebraic expressions
- solving algebraic or matrix equations (if possible)
- calculating the formula for the derivative or antiderivative of a function
- calculating the exact value of a definite integral and expressing the answer in terms of constants
- determining the analytical solutions of differential equations (if possible)

The Symbolic Math Toolbox now uses the **MuPAD language** (older versions are based on MAPLE).

Symbolic derivations can be documented with embedded text, graphics, and typeset math in the **MuPAD Notebook APP.**

symbolic numbers

Symbolic numbers must be declared with the sym function.

```
>> x=sym(1/3)
x =
1/3   % notice that symbolic output is not indented!
>> sqrt(x)
ans =
3^(1/2)/3
```

Compare **x** to its numeric (floating point) equivalent **y**

```
>> y=1/3
y =
    0.3333
>> sqrt(y)
ans =
    0.5774
```

x is in exact rational form while **y** is a decimal approximation

symbolic numbers

A few more examples:

```
>> a=sym(2)^(4/3)
a =
2*2^(1/3)

% now move the ^ symbol inside the bracket
>> b=sym(2^(4/3))
b =
2837089985411397/1125899906842624

% evaluate b to (the default) 32 significant digits
>> vpa(b)
ans =
2.5198420997897459372438788705044

% evaluate b to 10 significant digits
>> vpa(b,10)
ans =
2.5198421
```

symbolic variables

Symbolic variables can be created with the sym or syms commands.

The following commands are equivalent

```
>> x=sym('x');
>> syms x
```

do NOT do this

```
>> sym x;  % check the Workspace – x does not appear!
>> disp(x)
Undefined function or variable 'x'.
```

Use the syms to create several variables at once

```
>> syms x y z
```

but use the sym to create a numbered sequence of variables

```
>> l=sym('x', [1 10])
l =
[ x1, x2, x3, x4, x5, x6, x7, x8, x9, x10]
```

notice that only vector `l` is listed in the Workspace.

symbolic functions

```
>> syms a b c x  % create symbolic variables a, b, c and x

% create a symbolic function with or without the sym
>> f = sym('a*x^2 + b*x + c'); % string inside sym
>> f = a*x^2 + b*x + c;

>> expand(f^2) % using f in an algebraic expression
ans =
a^2*x^4 + 2*a*b*x^3 + 2*a*c*x^2 + b^2*x^2 + 2*b*c*x + c^2

>> subs(f,x,2) % substitute one of the variables
ans =
4*a + 2*b + c

>> subs(f,[a, b,c], [1, -4 1]) % multiple substitutions
ans =   % old and new values must be listed as vectors
x^2 - 4*x + 1

>> solve(ans,x) % solving f=0 with the chosen coefficients
ans =
 2 - 3^(1/2),  3^(1/2) + 2
```

symbolic functions

We can (attempt to) solve algebraic and differential equations and systems of equations with the solve function.

```
>> disp(f)
a*x^2 + b*x + c
>> solve(f,x)          % solve for x
ans =
 -(b + (b^2 - 4*a*c)^(1/2))/(2*a)
 -(b - (b^2 - 4*a*c)^(1/2))/(2*a)
>> solve(f,a)
ans =
-(c + b*x)/x^2         % solve for a
```

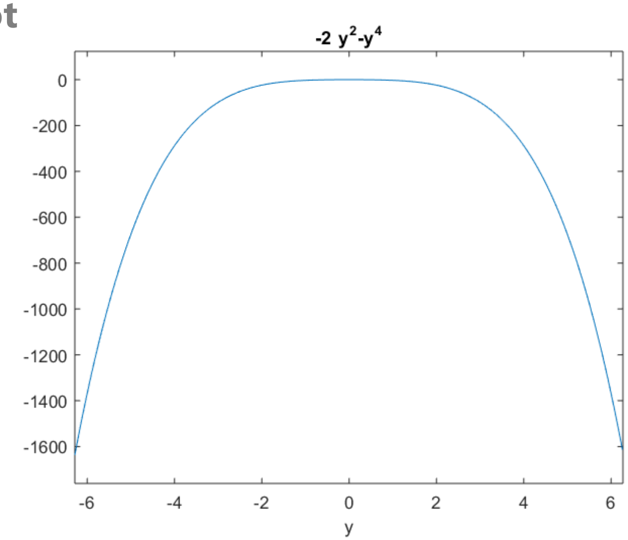
```
>> f1=subs(f,[a, b,c], [1, -4 1]); disp(f1)
x^2 - 4*x + 1              % define a parabola
>> l1=x-10; disp(l1)       % define a line
x - 10
>> solve(f1-l1)            % look for intersection
5/2 - (19^(1/2)*1i)/2
(19^(1/2)*1i)/2 + 5/2      % no real intersection
```

ezplot and ezsurf

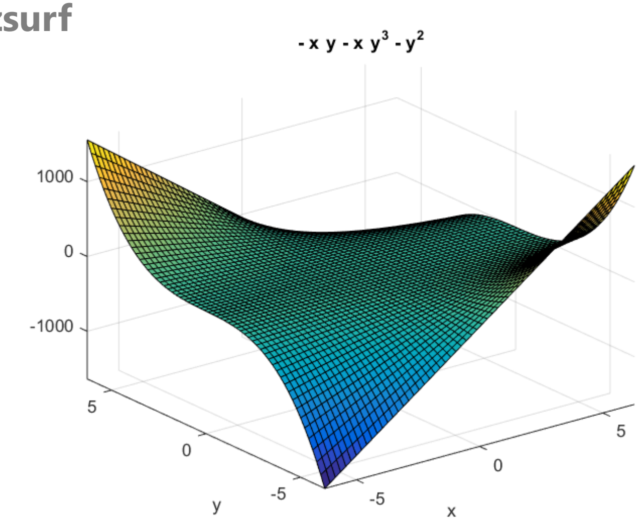
The ezplot and ezsurf functions can be used plot symbolic functions.
The default axis range is [-2π, 2π].

```
% define an explicit function of 1 variable
>> f2=sym('-2*y^2-y^4');
% define an explicit function of 2 variables
>> f3=y^2-x*y-x*y^3;

>> ezplot(f2) % ezplot('-2*y^2-y^4') also works
>> figure, ezsurf(f3)
```

ezplot



ezsurf



manipulating algebraic expressions

The following examples demonstrate the use of the pretty, expand, collect, simplify and factor functions

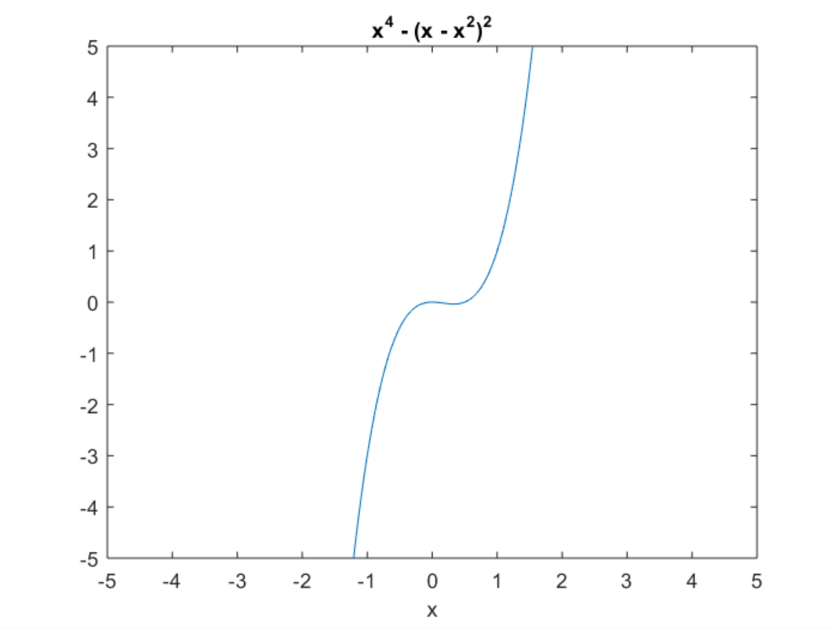
```
>> p1 = x^2*(z - y)^2 + (z - x^2*y + 2)*(x + y + z^2);

>> pretty(p1) % create a more readable output
  2                2              2         2
(z  + x + y) (- y x  + z + 2) + x  (y - z)

>> expand(p1) % expand the expression
ans =
2*x + 2*y + x^2*z^2 + x*z + y*z - x^3*y + 2*z^2 + z^3 -
2*x^2*y*z - x^2*y*z^2

>> collect(p1,z) % rewrite p1 in terms of the powers of z
ans =
z^3 + (x^2 - x^2*y + 2)*z^2 + (- 2*y*x^2 + x + y)*z +
x^2*y^2 - (y*x^2 - 2)*(x + y)
```

manipulating algebraic expressions

```
% p1 is a linear function of variable y
>> collect(p1,y)
ans =
(z - 2*x^2*z - x^2*(z^2 + x) + 2)*y + x^2*z^2 + (z^2 + x)*(z + 2)


% so there is only 1 solution to p1(y)=0
>> solve(p1,y)
ans =
-(x^2*z^2 + (z^2 + x)*(z + 2))/(z - 2*x^2*z - x^2*(z^2 + x) + 2)


% p1 cannot be factorised so this is the same as expand(p1)
>> factor(p1)
ans =
2*x + 2*y + x^2*z^2 + x*z + y*z - x^3*y + 2*z^2 + z^3 -
2*x^2*y*z - x^2*y*z^2
```

manipulating algebraic expressions

```
% now try a simple polynomial expression p2=p2(x)

>> p2=x^4-(x^2-x)^2; disp(p2)
x^4 - (- x^2 + x)^2

>> disp(collect(p2))
2*x^3 - x^2

>> disp(simplify(p2))
x^2*(2*x - 1)

>> disp(factor(p2))
[ x, x, 2*x - 1]

>> solve(p2)
ans =
   0
   0
 1/2

>> ezplot(p2,[-5,5,-5,5])
```



the factor function

```
% integer input x – factor returns the prime factorization of x
>> disp(factor(270480))
      2      2      2      2      3      5      7      7     23

% symbolic variable input
>> disp(factor(x^2-4))
[ x - 2, x + 2]

% factor only allows for integer coefficients
>> disp(factor(x^2-1/4))
[ 1/4, 2*x - 1, 2*x + 1] % factorisation includes the common factor ¼

% so neither of the following will work
>> factor(x^2-2)  % irrational roots
>> factor(x^2+1)  % complex roots

% in such cases use the solve function to find the non-integer factors
>> disp(solve(x^2+1))
-1i
 1i
```

example: polynomial (long) division

Consider the following example of polynomial division

$$\frac{x^3 - 7x^2 + 23x - 9}{x - 2} = \underbrace{\mathbf{x^2 - 5x + 13}}_{\text{Quotient}} + \frac{\overbrace{\mathbf{17}}^{\text{Remainder}}}{x - 2}$$

[Q,R] = quorem(u,v) divides u by v and returns the quotient Q and remainder R of the division, such that u = Q*v + R.

```
% Example

>> u=x^3 - 7*x^2 + 23*x - 9;
>> v=x-2;
>> [Q, R]=quorem(u,v)
Q =
x^2 - 5*x + 13
R =
17
```

example: partial fractions

Consider the following example of partial fraction decomposition

$$\frac{2x-3}{x^3-x^2} = \frac{A}{x-1} + \frac{B}{x} + \frac{C}{x^2}$$

We could bring the RHS to the common denominator and then use an appropriate substitution to determine the constants A, B and C. Alternatively we can get the answer using the partfrac function.

```
% Example

>> g=(2*x-3)/(x^3-x^2);
>> disp(partfrac(g))
1/x - 1/(x - 1) + 3/x^2

>> pretty(partfrac(g))
1     1      3
- - ----- + --
x   x - 1    2
            x
```

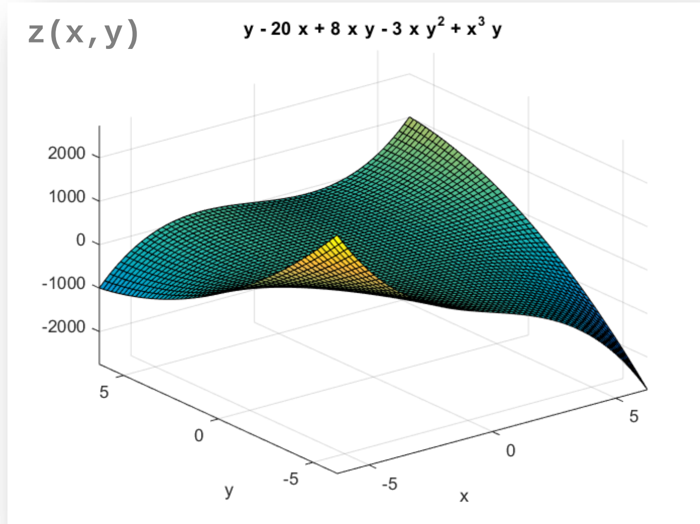
differentiation example

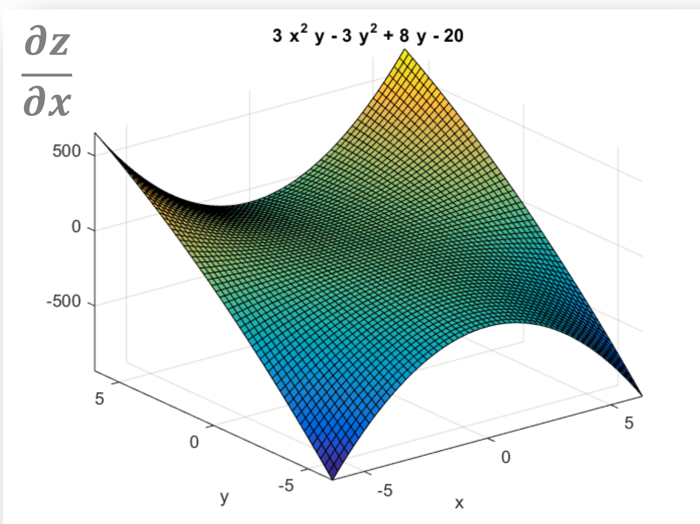
```
% Example 1:  w=w(x), function of 1 variable


>> w=x^4 - 3*x^3 + 8*x^2 - 20; % quartic polynomial

>> w1=diff(w); disp(w1)           % 1st derivative
4*x^3 - 9*x^2 + 16*x

>> w2=diff(w,2); disp(w2)          % 2nd derivative
12*x^2 - 18*x + 16

>> w3=diff(w,3); disp(w3)         % 3rd derivative
24*x – 18

>> w4=diff(w,4); disp(w4)         % 4th derivative
24

>> w5=diff(w,5); disp(w5)         % 5th derivative
0
```

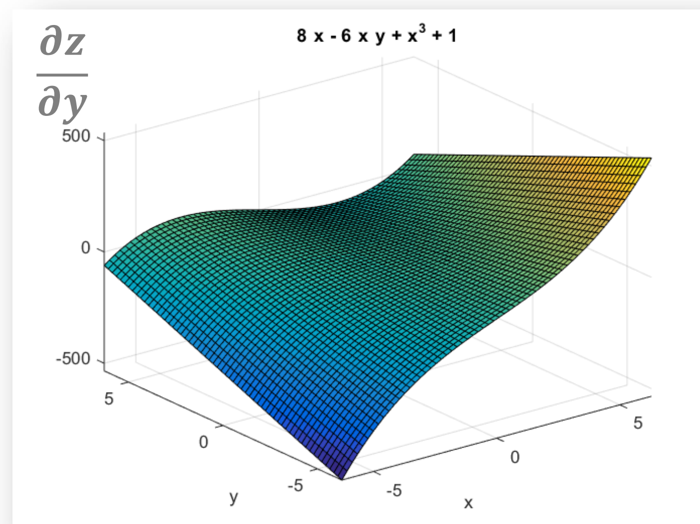
differentiation: partial derivatives

```
% z=z(x,y), function of 2 variables

% define a surface
>> z = x^3*y - 3*x*y^2 + 8*x*y - 20*x + y;

% list of symbolic variables in z
>> disp(symvar(z))
[ x, y]

% if the differentiation variable is not specified
% diff uses the 1st (default) variable in symvar

>> disp(diff(z)) % same as diff(z,x)
3*x^2*y - 3*y^2 + 8*y – 20


>> disp(diff(z,y))
8*x - 6*x*y + x^3 + 1
```

z(x,y) and its first partial derivatives

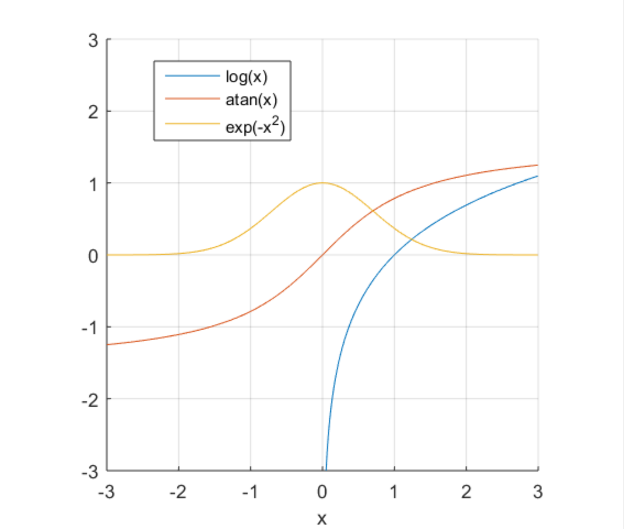






indefinite and definite integrals

```
% Example functions
>> y1=log(x); y2=atan(x); y3=exp(-x^2);

% indefinite integrals (constant of integration NOT included!)
>> disp(int(y1))
x*(log(x) - 1)
>> disp(int(y2))
x*atan(x) - log(x^2 + 1)/2
>> disp(int(y3))
(pi^(1/2)*erf(x))/2
```




```
% definite integral examples
>> int(y1,0.5,1.5); disp(vpa(ans,3))
-0.0452
>> int(y2,0,1); disp(vpa(ans,3))
0.439
>> int(y3,-inf,inf); disp(ans)
pi^(1/2)
```

area integration problem

Find the area between the curves $f(x) = -2x^2 + 12x - 12$ and $g(x) = x^2 - 10x + 27$.
Create a plot showing the area in question (see example)

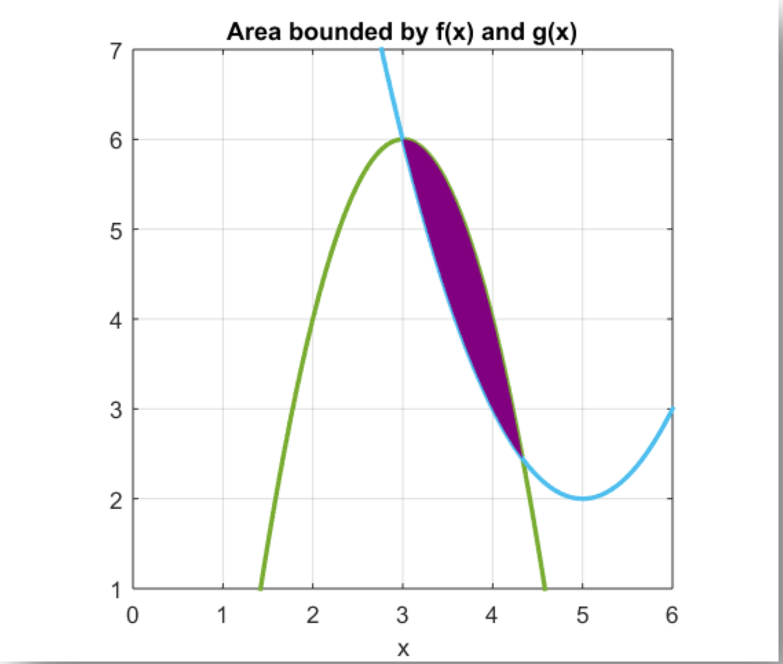
Suggestions

- define f(*x*) and g(*x*) as symbolic functions and use **ezplot** to check the graphs
- use **solve** to find the intersection point
- use **int** to calculate the area

plotting numerical data is easier so consider changing the variables from symbolic to numeric (double) type

- use **fill** to calculate the area
- make sure you understand the geometry (i.e. know how to determine the x and y coordinate (vectors) of the polygon in **fill**

Feel free to use a different approach!



useful symbolic functions

function name	type
sym (syms)	format
vpa	format
pretty	format
symvar	format
subs	equation/function
solve	equation/function
expand	algebra
collect	algebra
simplify	algebra
factor	algebra

function name	type
ezplot	line plot
ezsurf	surface plot
ezmesh	surface plot
quorem	algebra
partfrac	algebra
diff	calculus
int	calculus
symsum*	calculus
limit*	calculus
taylor*	calculus

* the functions **symsum**, **limit** and **taylor** do not appear in these notes