EECE 231: Symbolic Math
Matlab help for symbolic math

OBJECTIVES

- Learn Matlab basics for symbolic computations
- Learn how to integrate, differentiate, solve
- Learn how to plot symbolic expressions
- Learn how to solve a system of equations
- Learn how to solve a system of ordinary differential equations
- Learn how to convert from symbolic to numeric
- Be able to find what you mind need later for symbolic computation with Matlab

OUTLINE

SYMBOLIC MATH TOOLBOX

- Solve symbolic math expressions
- Support for analytical derivative, integration, factorization, and equation solving
- Perform variable-precision arithmetic

SYMBOLIC VARIABLES IN MATLAB

- You declare symbolic variables with the `syms` keyword (may take few seconds)
- Then you use Matlab as usual to build the expressions

```
>> syms x
>> z = x^2 + cos(x);
>> f = int(z)
f =

sin(x) + x^3/3
```

- Matlab uses a kernel called MAPLE to execute the symbolic commands

FANCY STUFF

- Factorization, integration, solving systems of equations

```
>> factor((x^3 -y^3)/(x^4-y^4))
(x^2 + x*y + y^2)/((x + y)*(x^2 + y^2))

>> int(1/(1+x^2))
atan(x)

>> diff(x^2)
2*x

>> [a b] = solve('x+y=7','x-y=1');
[4 3]
```

SUBSTITUTING NUMERIC VALUES FOR SYMBOLIC VARIABLES

- `subs` substitutes symbolic variables with values
  - Values can be symbolic expressions or numeric values

```
>> syms a b c x;
>> f =a*x^2 + b*x + c;
>> subs(f,x,5)
ans=
  25*a + 5*b + c
>> subs(f,[x a b c], [ 5 1 2 3 ])
ans=
  38
```

SYMBOLIC AND MATRIX SYMBOLIC COMPUTATIONS

- Vector and matrix operations extend to symbolic variables
- For instance, `diff` computes the derivative of *A* with respect of *x* (derivative of each entry)

```
>> syms x;
>> A=x.^[1 2 ; 3 4]

A =

[   x, x^2]
[ x^3, x^4]

>> D=diff (A )

D =

[     1,   2*x]
[ 3*x^2, 4*x^3]
```

OUTLINE

PLOTTING SYMBOLIC EXPRESSIONS

- `ezplot` plots symbolic expressions
- check the double integral of a double differential with the plots
- Compute the difference $e = f - g$

```
syms x;
f = 1/(5+4*cos(x));
f1= diff(f);
f2= diff(f1);
ezplot(f);pause
ezplot(f1);pause
ezplot(f2);
g = int(int(f2));
ezplot(g);

e = f-g; ezplot(e);
```

PLOTTING SYMBOLIC FUNCTIONS IN 3D

```
syms x y;
f = sin(x);
ezsurf(f);

% something cool
g = cos(y);
ezsurf( f+g);

% more cool
ezsurf(real(atan(x+i*y)));
```

OUTLINE

SOLVING SYSTEMS OF EQUATIONS

- `solve` takes one or more formulae, and solves for the corresponding variables.
- It can solve for partial variables, i.e. express one variable in terms of others
- Other applications: to find local min and max of a function, use `diff` and then `solve`

See also: `linsolve`

```
>> syms x y;
>> [a b]=solve('x+y-3=0','x+2*y-6=0')
a = 0, b = 3
>> [a b] = solve(x+y-3, x+2*y-6);
a = 0, b = 3
>> f(1) = x + y -3; f(2)=x+2*y-6;
>> [a b] = solve(f);
a = 0, b = 3

>> y = 2*x + 4;
>> solve(y,x)
  -2 + (1/2)*y
```

OUTLINE

SOLVING ORDINARY DIFFERENTIAL EQUATIONS

- `dsolve` takes one or more ordinary differential equations, with initial conditions
- default variable is *t*
- Symbol *D* means $\frac{d}{dt}$
- Symbol *Dk* means $\frac{d^k}{dt^k}$

```
>> f = dsolve('Dx = -a*x')
C2/exp(a*t)
>> a =1;C2=3; g = subs(f); ezplot(g)

>> x = dsolve('Dx = -a*x','x(0) = 1')
x = 1/exp(a*t)

>> [S1 S2] = dsolve('Df = f + g',
                    'Dg = -f + g',
          'f(0) = 1','g(0) = 2');
>> S1
exp(t)*cos(t) + 2*exp(t)*sin(t)
>> S2
2*exp(t)*cos(t) - exp(t)*sin(t)

>> w = dsolve('D3w = -w','w(0)=1',
                'Dw(0)=0', 'D2w(0)=0')
w = 1/(3*exp(t)) +
    (2*exp(t/2)*cos((3^(1/2)*t)/2))/3
>> ezplot(w)
```

OUTLINE

LIMITS AND SUMMATIONS

- `limit(f,x,-inf)` finds the asymptotic limit of f as x goes to $-\infty$
- `symsum(f,start, end)` find a form for the sum of a sequence
- Similarly for `symprod`

```
>> syms x;
>> limit(1/x)
  NaN
% by default as x goes to zero
>> limit(x/x^2, inf)
  0

>> syms k;
>> symsum(1/k^2,1,inf)
  1/6*pi^2
>> symsum(x^k,k,0,inf)
  -1/(x-1)
```

OUTLINE

VARIABLE PRECISION ARITHMETIC

- Specify the number of precision digits with `digits`
- Use `vpa` to perform the precise computation
- Check the number of digits in factorial of 30 using symbolic toolbox

```
>>digits(70);
>>vpa(19/81)
0.2345679012345679012345679012345679012345679012345679012345679012
```

```
>>digits(781);
>>vpa(pi)
3.1415926535897932384626433832795028841971693993751058209749445923
```

```
>> digits(33);
>> vpa(factorial(30))
 265252859812191032188804700045312.0
```

CONVERSION FROM SYMBOLIC TO NUMERIC

double	Convert symbolic matrix to MATLAB numeric form
int8, int16, int32, int64	Convert symbolic matrix to signed integers
sym	Create symbolic objects
vpa	Variable-precision arithmetic

```
>> a = sym(sqrt(2)*20)
20*2^(1/2)
>> x = double(a)
   28.2843
>> y = int16(a)
     28
>> digits(15)
>> z = vpa(a)
28.2842712474619
>> v  = ceil(a*10)
283
% other similar functions: ceil, conj, floor,
%   imag, real, round
```

ONLINE REFERENCE OF SYMBOLIC FUNCTIONS

Check the online reference from Mathworks
http://www.mathworks.com/help/symbolic/functionlist.html