**Next:** [About this document](http://texas.math.ttu.edu/%7Egilliam/ttu/mlhtml_1/symbolic/node1.html)

**MATLAB SYMBOLIC TOOLBOX**

1. The *Symbolic Toolbox* is different from all other Matlab Toolboxes. Some of the information in this tutorial is taken from *Mastering Matlab* by Duane Hanselman and Bruce Littlefield (Printice-Hall, The Matlab Curriculum Series), the Symbolic Toolbox User Guide, as well as our own experience.

Using a special version of the Maple Kernel, the Toolbox performs symbolic analysis using character strings, rather than numerical analysis on numerical arrays.

maple('sum(''1/k!'', ''k''=0..infinity)') %note the use of double quotes

The *Symbolic Toolbox* is a collection of *tools* for manipulating and solving symbolic expressions. When you pass a command to Matlab to perform a symbolic operation, it asks Maple to do it and return the result to the Matlab command window.

1. Here is a noteworthy example:

Using the Matlab Toolbox we can write

H=sym(15,15,'1/(i+j-1)');

x=sym(15,1,'1');

b=symmul(H,x);

flops(0);

linsolve(H,b);

flops

On the other hand, in Matlab syntax we have

H=hilb(15);x=ones(15,1);

b=H\*x; flops(0);

x=H\b;

flops

cond(H)

What do you learn from a direct comparison of exact and floating point solutions of this linear system?

1. Here is another example. First, in Matlab syntax we compute the eigenvalues of the tex2html_wrap_inline118Hilbert matrix
2. H=hilb(3)

mlr=eig(H)

Now we redo the calculation using Maple

H=maple('hilbert','3')

r=eigensys(H)

From a practical point of view this answer is useless, since we don't really have any idea what is the size of these numbers. Lets convert to floating point.

mar=numeric(r)

maple('evalf',r,50)

Notice this last answer, while correct to any realistic digits, is still rather strange since *H* is symmetric and we would expect to be returned real numbers like Matlab did. The point is that sometimes floating point answers are better than exact or floating point approximations of exact answers.

1. *Symbolic Expressions* are Matlab Strings, or arrays of character strings, that can represent numbers, functions, operators, and variables. The variables are not required to have predefined values. *Symbolic Equations* are symbolic expressions containing an equal sign. *Symbolic Arithmetic* is the pratice of solving symbolic symbolic equations. *Symbolic Matrices* are arrays whose elements are symbolic expressions.

Here are some examples:

diff('cos(x)') % differentiate cos(x) with respect to x

M=sym('[a,b;c,d]') % create a symbolic matrix M

determ(M) % find the determinant of the matrix M

1. The maple toolbox is primarily used for symbolic computation and exact integer arithmetic but is not confined to these uses. Usually though, if you wanted to do a numerical calculation, you would just write a program in matlab.
2. Symbolic expressions in matlab are given as character strings, e.g.,

'cos (x)'

or

'1/(1+x^2)'

1. There is a wide range of operations that can be performed on such expressions including many from differential and integral calculus, linear algebra and differential equations.
2. Examples include
   1. diff('cos(t)')

in simple cases the program can guess the variable to differentiate with respect to without being told.

* 1. int('1/(1+x^2)')

1. Suppose *f* is something like

f='x^n'

or

f='sin(a\*t+b)'

then more general syntax for diff and int are

* 1. diff(f,'x')

or

diff(f,'n')

Note the difference

* 1. diff(f,'x',k)

where *k* is an integer differentiates *f*, *k* times.

* 1. int(f,'t')

finds the indefinite integral.

* 1. int(f,'t',a,b)

finds the definite integral from *a* to *b*.

* 1. The following integral demonstrates the usefullness of exact integrals

displaymath122

For example try

int('2\*x^2\*(19+12\*x^2)/(7\*(x^2+1))','x',0,1)

numeric

The first ans gives a better answer from a mathematical point of view.

Here are some exercises:

* 1. differentiate the following functions
     1. x^2\*sin(x^3)\*e^{2x}
     2. x^6\*log(cos(x^2)\*x)
     3. x^3\*e^{x^4}
     4. (ax+b)/(cx+d)
     5. atan(x^2+1)
  2. try to integrate the answers you got in the last problem
  3. Integrate the following functions
     1. x^7
     2. 1/x
     3. log(x)\*sqrt(x)
     4. x^7 % on (0,1)
     5. 1/x % on (1,2)
     6. sqrt(x)\*log(x) % on (0,1)
     7. exp (-x^2) % on (0,infinity)
  4. Find the derivatives (give it a try by hand first) and simplify:
     1. tex2html_wrap_inline124
     2. tex2html_wrap_inline126

1. You must be a little careful about names of symbolic variables using diff and int without specifying a variabe. In particular, the variable must be a single character in the range `a':`z' excluding `i' and `j'. More complex variable names are allowed, however, you must explicitly describe the variable:
3. f='exp(i\*theta)'

diff(f,'theta',2)

gives

-exp(i\*theta)

1. You can write these commands in a matlab m-file and run them just like any matlab program.
2. Here is some syntax with an example. Type the statement and hit return.
4. f='1/(5+4\*cos(x))'
5. ezplot(f)
6. f2=diff(f,2)
7. ezplot(f2)

axis([-2\*pi 2\*pi -2 2])

here *axis* is a matlab command to set an axis range. Here is another example

f3=diff(f2)

pretty(f3)

Lets try to simplify this

f4=simple(f3)

pretty(f4)

the command *simple* tries to simplify a complex expression. It will try many different Maple options. Lets look at root finding in the toolbox:

z=solve(f3)

the command *solve* finds zeros of the function. The answer *z* is a symbolic matrix. To find numeric values issue the next command.

x=numeric(z)

*numeric* without an argument numerically evaluates the last expression. Sometimes you might want to pick off, say, the second answer in *z*. The next line shows one way to do this

s=sym(z,2,1)

*sym* is the symbol manipulator. Note that *z* is a *3* by *1* matrix so we type the pair *2,1* to indicate that entry. Often you want to substitute a value into a symbolic expression. This can be done in several ways. One way is as follows:

subs(f3,s)

Now lets try to recover *f*

g=int(int(f2))

Note this does not look like *f*. To check our work lets take *f-g* and see what we get. Now *f* and *g* are strings in matlab with different lengths, so you cannot simply take *f-g*. To carry out symbolic difference (*symsub* for symbolic subtraction), you would type the following:

d=symsub(f,g)

Note the answer is not zero but, rather, as we know from calculus the answer can differ by a constant.

simple(ans)

You should use *mhelp* to look at the following related topics *symadd*, *symmul*, *symdiv*, *sympow*, *symop*, *symrat*, *symvar*, *symsize*.

1. Now lets turn to symbolic algebra where we will look at just a few simple tools for algebraic manipulation.
2. f='(x+y)^10\*(2\*u+3\*v)^7\*(x-2\*y)^3'

g=expand(f)

the command *expand* multiplies all the terms out. As simple examples, try expanding

(x-1)\*(x-2)\*(x-3)

x\*(x\*(x-6)+11)-6

a\*(x+y)

exp(a+b)

n\*(n-1)!

Now consider the reverse process *factoring*:

factor(g)

Try factoring

x^3-6\*x^2+11x-6

x^6+1

x^4+4

x^4-2\*x^2+1

1. The next examples show how Matlab loops can be combined with maple syntax. Consider the four lines:
2. for n=1:9
3. p=symadd(sympow('x',n),1);
4. disp([p '= ' factor(p)])

end

1. The *factor* command used above can also factor numbers:
2. for n=1:17
3. N=setstr('1'\*ones(1,n));
4. fn=factor(N);
5. disp([N,' = ',fn])

end

1. Besides factoring you can also simplify expressions:

simplify(f)

here *f* must be a maple expression. Try letting *f* be the following expressions and use *simplify*

(1-x^2)/(1-x)

(1/a^3+6/z^2+12/a+8)^{1/3}

e^xe^y

cos^2(x)+sin^2(x)

1. A related tool that tries many types of simplification methods (not a maple command) included in the symbolic toolbox is *simple(f)*. try the following expressions for *f*:
2. 2\*cos^2(x)-sin^2(x)
3. cos^2(x)-sin^2(x) cos(x)+i\*sin(x)

cos(3\arccos(x))

1. Finding a sequence of integrals can be done using a Matlab loop (we do this same example below in part 21. several other different ways.) The answers are the Fourier coefficients of the function tex2html_wrap_inline128.
2. n=10
3. f1='2\*sin(j\*pi\*x)\*x^2\*(1-x)^2';
4. for j=1:n
5. f=subs(f1,j,'j');
6. ss=['x=0','..','1'];
7. vvv=(maple('int',f,ss));
8. y(j)=eval(vvv)

end

1. Lets compute the first six Tchebyshev polynomials
2. clear
3. n=6;
4. v=ones(1,n);
5. t=sym(v);
6. f='cos(k\*acos(x))';
7. for k=1:n
8. t=sym(t,1,k,subs(f,k,'k'))

end

Note that these don't look like polynomials. Lets fix that using simple and expand

tcheb=sym(ones(1,n));

for j=1:n

%tcheb=sym(tcheb,1,j,simple(sym(t,1,j)))

tcheb=sym(tcheb,1,j,expand(sym(t,1,j)))

end

The Tchebshev polynomials can also be built by recursion. Lets do this using matlab polynomials which, you may recall are given by vectors, i.e., if tex2html_wrap_inline130. The recusion formula for the Tchebyshev polynomilas is

displaymath132

Note that the polynomial tex2html_wrap_inline134is given in matlab by [1, 2, 3]. In order to express the polynomial *xp*(*x*) we would write [1, 2, 3,0]. Using this information we have

t0=1;

t1=[1 0];

for j=1:(n-1)

eval(['t',int2str(j+1),' = [2\*t',int2str(j),' 0]-[0 0 t',int2str(j-1),']'])

end

If you really like to see the *x*'s we can a symbolic toolbox feature that turns Matlab polynomials into regular variable polynomials.

for j=1:n

eval(['tchebp',int2str(j),' = poly2sym(t',int2str(j),')'])

end

Now lets plot our answers: First lets plot the matlab polynomials *t*1, *t*2, *t*3 *t*4 *t*5 *t*6

clear y

xvec=0:.05:1;

for j=1:6

eval(['y(:,',int2str(j),')=polyval(t',int2str(j),',xvec)'';'])

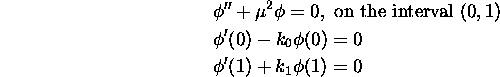
end

plot(xvec,y)

axis([0 1 -1.5 1.5]);

Before ending this topic we should mention that many important special functions including the standard orthogonal polynomials are built into Maple but they are only available with the Extended symbolic Toolbox.

1. Finding zeros of a function. One place that Matlab is not so good is in finding zeros of functions. You can use *fzero* but it is not so good. We will now demonstrate the use of Maple's *solve* to find zeros using the Symbolic Toolbox. The next example arises in find the eigenvalues of the following Sturm-Louiville boundary value problem:



The Characteristic equation for this problem can be written as

displaymath146

Consider the case tex2html_wrap_inline148and tex2html_wrap_inline150. Note that, using *x* for tex2html_wrap_inline154this equation can also be written in terms of sin's and cos's. Plotting the resulting function

ezplot('(1\*2-x^2)\*(sin(x)/x)+(1+2)\*cos(x)',[.0001,4\*pi])

we easily see that there are infinitely many zeros interlacing with multiples of tex2html_wrap_inline156.

Lets find the first 10 zeros.

k0=1

k1=2

f=subs('(k0\*k1-x^2)\*(sin(x)/x)+(k0+k1)\*cos(x)',k0,'k0');

f1=subs(f,k1,'k1');

for j=1:10

ss=['x=Pi\*',int2str(j-1),'..',int2str(j),'\*Pi'];

vvv=(maple('fsolve',f1,ss));

mu(j)=eval(vvv);

end

mu'

1. Here is an example of a problem that arises in the theory of ill-posed inverse problems. My goal in this example is several fold: 1) show how you can represent the solution of some PDE's, 2) show how hard solving inverse problems can be, 3) provide a platform from which to learn other techniques using Matlab and the Symbolic Toolbox together.

Consider the one dimensional heat equation on the interval *0* to *1*:



The solution to this problem can be written in terms of a Dirichlet series:

displaymath160

where

displaymath162

Imagine that we can sample the temperature at a single point tex2html_wrap_inline164on the rod at a sequence of times tex2html_wrap_inline166to obtain output tex2html_wrap_inline168. Using this information we want to approximate the Fourier coefficients tex2html_wrap_inline170and hence approximate the initial condition tex2html_wrap_inline172. Defining the matrices

displaymath174

we arrive at a system of equations:

displaymath176

Assuming that tex2html_wrap_inline164is irrational, then the matrix *P* is invertible and, as a generalized Vandermonde matrix, *A* is invertible. Therefore, in principle, we can solve for *X* and obtain:

displaymath186

With this we obtain an approximation to *f*, namely,

displaymath188

Lets consider an example:

displaymath190

In a physical experiment we would measure the outputs using some type of sensing device but in the present case we will simply approximate the solution using the Dirichlet series formula for the solution. To do this we must compute the Fourier coefficients of the initial data. Lets do it using the symbolic toolbox and using matlab:

First lets use Maple

nn=20

for j=1:nn

f=subs('2\*x^2\*(1-x)^2\*sin(j\*pi\*x)',j,'j');

ss=['x=0','..','1'];

vvv=(maple('int',f,ss));

cma(j)=eval(vvv);

end

We could also have typed

maple('g:=(j)->int(2\*x^2\*(1-x)^2\*sin(j\*pi\*x),x=0..1)')

sg=maple('[seq(g(j),j=1..20)]')

cma=numeric(sg) % these are the Fourier coefficients of f

Now lets use Matlab: We can build a function file SPMquotcoefn.m", for example,

function y=coefn(x)

global jj

y=2\*x.^2.\*(1-x).^2.\*sin(jj\*pi\*x);

Now write the program SPMquotfcml"

global jj

nn=20

% use Tchebyshev quadrature

nc=50; % the number of Tchebyshev nodes

xc=cos((2\*(1:nc)-1)\*pi/(2\*nc)); %nodes for Tchebyshev quadrature

% you can use any interval with (a , b) for the integration, here we use

a=0;

b=1;

for j=1:nn

jj=j;

cml(j)= ((b-a)\*pi)/(2\*nc)\* sqrt(1-xc.^2)\*coefn((b+a+(b-a)\*xc)/2)';

end

cml' % these are the Fourier coefficients of f

Notice that the result was almost instantaneous.

Lets compare the results

format long

[cma' cml']

Check the Fouier coefficients

nn=8;

x=0:.01:1;

J=(1:nn);

cmly=cml(1:nn)\*sin(pi\*J'\*x);

plot(x,cmly,x,(x.\*(1-x)).^2)

Looks like a good approximation

Lets look at numerical solution of the heat problem on tex2html_wrap_inline192

n=20;

t=[0:.05:2]';

x=0:.05:1;

J=(1:n);

v=cml'; % build a column vector of Fourier coefs

vv=v(:,ones(size(x))); %this is a neat trick to build v\*ones(size(x))

Y=exp(-pi^2\*t\*J.^2)\*(vv.\*sin(J'\*pi\*x));

surf(x,t,Y)

That's pretty easy. Okay back to solution of our inverse problem

Now suppose we sample the temperature at *n*=8 time values, say at the times tex2html_wrap_inline196, at the *x* value tex2html_wrap_inline200.

x\_0=1/pi;

nn=20;

n=8;

T=(1:n).^2/(n^2);

J=(1:nn);

Y=exp(-pi^2\*T'\*J.^2)\*(cml(1:nn).\*sin(J\*pi\*x\_0))'

n=8;

x\_0=1/pi;

V=1:n;

A=exp(-pi^2\*(T)'\*(V.^2));

P=diag(sin(V\*pi\*x\_0));

AP=A\*P;

X=AP\Y

nn=8;

x=0:.01:1;

J=(1:nn);

y=X'\*sin(pi\*J'\*x);

plot(x,y,x,(x.\*(1-x)).^2)

Looks bad, lets try a few less coefficients

nn=3;

x=0:.01:1;

J=(1:nn);

y=X(1:nn)'\*sin(pi\*J'\*x);

plot(x,y,x,(x.\*(1-x)).^2)

Well that looks a bit more like tex2html_wrap_inline128.

Now lets give Maple a chance: We could rebuild the above matrices using the Maple toolbox, for example,

maple('f:=(i,j)->exp(-pi^2\*i^2\*j^2/8^2)')

SA=maple('matrix(8,20,f)')

vsg=maple('matrix(20,1,[seq(g(j)\*sin(j),j=1..20)])')

symmul(SA,vsg)

We find that results cannot be to long.

A simpler way is as follows

SA=sym(A);

SP=sym(P);

SY=sym(Y);

XS=symmul(inverse(SP),linsolve(SA,SY))

%SX=maple('evalf',XS)

SX=numeric(XS)

Lets try our plot with *nn*=3 as above

nn=3;

x=0:.01:1;

J=(1:nn);

sy=SX(1:nn)'\*sin(pi\*J'\*x);

plot(x,sy,x,(x.\*(1-x)).^2)

Lets try *nn*=2

nn=2;

x=0:.01:1;

J=(1:nn);

sy=SX(1:nn)'\*sin(pi\*J'\*x);

plot(x,sy,x,(x.\*(1-x)).^2)

Well it might be okay for government work but not really very good.

Once again the exact answer is of no use and an attempt to obtain a floating point approximation to it fails. The theory of ill-posed inverse problems would suggest that we ``regularize'' the problem and solve as follows:

n=8;

x\_0=1/pi;

V=1:n;

A=exp(-pi^2\*(T)'\*(V.^2));

P=diag(sin(V\*pi\*x\_0));

RX=inv(P)\*((A'\*A +.000065\*eye(n))\(A'\*Y));

[cml(1:n)' X SX RX]

nn=8;

x=0:.01:1;

J=(1:nn);

sy=RX(1:nn)'\*sin(pi\*J'\*x);

plot(x,sy,x,(x.\*(1-x)).^2)

The pictures better but at least we can use all the approximate Fourier coefficients in our approximation.

To be fair in this problem please redo the whole thing using the following sample times

n=8;

T=(1:n).^2/(pi^2\*n^2);

and

T=(1:n).^2/(n^4);

You will see that for the last *T* you can use all eight approximate coefficients. But the regularized solution does very bad. This is because you must adjust the parameter used to multiply times the identity (in this case change .000065 to zero.

1. As a final example, let us consider a problem in quadrature. Consider the seemingly harmless analytic function

f='sinh(a\*sqrt(sqrt(-1)\*x))/sinh(sqrt(sqrt(-1)\*x))'

on tex2html_wrap_inline212. We are really only interested in the real part which is oscillatory but decays like tex2html_wrap_inline214. From results in Fourier Analysis it can be shown that the real part of the integral of *f* from zero to infinity is zero.

First, lets use Matlab's *quad8* to approximate the integral. To this end build two *function files* ``yr.m"

function y=yr(x)

global aa

z=x+eps; % we put in eps to take care of the 0/0 when x=0

y=real(sinh(aa\*sqrt(sqrt(-1)\*z))./sinh(sqrt(sqrt(-1)\*z)));

and ``yi.m''

function y=yi(x)

global aa

z=x+eps;

y=imag(sinh(aa\*sqrt(sqrt(-1)\*z))./sinh(sqrt(sqrt(-1)\*z)));

Now build an m-file SPMquotint\_y.m"

global aa

aa=input(' global variable aa in (0,1) = ');

b(1)=0;

int\_yr(1)=0;

int\_yi(1)=0;

disp(' b int\_yr int\_yi')

disp('--------------------------------------------')

for j=2:7

b(j)=10^j;

int\_yr(j)=quad8('yr',b(j-1),b(j))+int\_yr(j-1);

int\_yi(j)=quad8('yi',b(j-1),b(j))+int\_yi(j-1);

fprintf('%8.0f \t %8.8f\t %8.8f\n',b(j),int\_yr(j),int\_yi(j))

end

Return to Matlab and type SPMquotint\_y" return. Try ``aa=.5'' and ``aa=.9''

Well we might guess that the integral of the real part is zero and the imag part is minus two hundred when ``aa=.9''.

Now lets try the symbolic toolbox: First the real part

maple('evalf(Re(Int(2\*sinh(.9\*i^.5\*v)/sinh(i^.5\*v)\*v,v=0..10000)),8)')

and now the imaginary part

maple('evalf(Im(Int(2\*sinh(.9\*i^.5\*v)/sinh(i^.5\*v)\*v,v=0..10000)),8)')

Now the fact is that the integral

displaymath218

can be expressed in terms of the Zeta function by

displaymath220

So it is clear that for every ``aa'' the real part is zero and we can compute the result using the symbolic toolbox to obtain

maple('-i\*(Zeta(0,2,1/2\*(1-.9))-Zeta(0,2,1/2\*(1+.9)))/2')

* [About this document ...](http://texas.math.ttu.edu/%7Egilliam/ttu/mlhtml_1/symbolic/node1.html" \l "SECTION00010000000000000000)

**Next:** [About this document](http://texas.math.ttu.edu/%7Egilliam/ttu/mlhtml_1/symbolic/node1.html)

*David Sexton Gilliam   
Tue Apr 9 12:41:14 CDT 1996*