### Symbolic Math Toolbox: Quick Reference Sheet

#### Symbolic Variables

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
</table>
| `syms`  | Create symbolic variables: `syms x;`  
Create arrays of symbolic scalar variables: `syms M [2 3];`  
Create symbolic matrix variables: `syms A [2 3] matrix;`  
| `symmatrix2sym` | Convert symbolic matrix variable to array of symbolic scalar variables: `syms A B [2 3] matrix; X = A + B; Y = symmatrix2sym(X)` |

#### Calculus

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>diff</code></td>
<td>Differentiation: <code>diff(sin(x^2+t),x)</code></td>
</tr>
<tr>
<td><code>int</code></td>
<td>Definite and indefinite integrals: <code>int(x/(1 + z^2),z)</code></td>
</tr>
<tr>
<td><code>release</code></td>
<td>Evaluate integrals: <code>F = int(cos(x),'Hold',true); G = release(F)</code></td>
</tr>
<tr>
<td><code>limit</code></td>
<td>Compute limit of symbolic expression: <code>limit(1/x,x,0,'left')</code></td>
</tr>
<tr>
<td><code>taylor</code></td>
<td>Taylor series: <code>x; taylor(exp(-x))</code></td>
</tr>
<tr>
<td><code>series</code></td>
<td>Puiseux series expansion: <code>x; series(1/sin(x),x)</code></td>
</tr>
<tr>
<td><code>symsum</code></td>
<td>Sum of a series: <code>x; symsum(1/x^2,1)</code></td>
</tr>
<tr>
<td><code>gradient</code></td>
<td>Gradient vector of scalar function: <code>syms x y z; gradient(x*y + 2*z*x,[x y z])</code></td>
</tr>
<tr>
<td><code>jacobian</code></td>
<td>Jacobian matrix: <code>syms x y z u v; jacobian([x*y*z; y; x+z],[x y z])</code></td>
</tr>
<tr>
<td><code>hessian</code></td>
<td>Hessian matrix of scalar function: <code>syms x y z; hessian(x*y + 2*z*x,[x y z])</code></td>
</tr>
<tr>
<td><code>laplacian</code></td>
<td>Laplacian of scalar function: <code>syms x y z; laplacian(1/x + y^2 + z^3,[x y z])</code></td>
</tr>
<tr>
<td><code>divergence</code></td>
<td>Divergence of vector field: <code>syms x y z; divergence([x^2 2*y z],[x y z])</code></td>
</tr>
</tbody>
</table>

#### Algebra

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>double</code></td>
<td>Convert symbolic values to double precision: <code>symN = sym(pi); doubleN = double(symN)</code></td>
</tr>
<tr>
<td><code>vpa</code></td>
<td>Control precision of computations with variable-precision arithmetic: <code>x; p = sym(pi); piVpa = vpa(p)</code></td>
</tr>
<tr>
<td><code>subs</code></td>
<td>Symbolic substitution: <code>syms a b; subs(a^3+b,[a,b],[2,sym('e'))</code></td>
</tr>
<tr>
<td><code>solve</code></td>
<td>Equations and systems solver: <code>syms a b u v; S = solve(u+v==a, u-v==b)</code></td>
</tr>
<tr>
<td><code>dsolve</code></td>
<td>Solve differential equations: <code>syms y(t) a; eqn = diff(y,t)==a*y; S = dsolve(eqn)</code></td>
</tr>
<tr>
<td><code>pdeCoefficients</code></td>
<td>Extract PDE Coefficients: <code>syms u(x,y); pdeeq = laplacian(u,[x y])== -3; coeffs = pdeCoefficients(pdeeq,u)</code></td>
</tr>
<tr>
<td><code>isolate</code></td>
<td>Isolate variable or expression in equation: <code>syms a b c x; isolate(a*x^2+b*x+c==0,x)</code></td>
</tr>
<tr>
<td><code>lhs</code></td>
<td>Left side (LHS) of equation: <code>syms x y; lhs(x^2 &gt;= y^2)</code></td>
</tr>
<tr>
<td><code>rhs</code></td>
<td>Right side (RHS) of equation: <code>syms x y; rhs(x^2 &gt;= y^2)</code></td>
</tr>
<tr>
<td><code>simplify</code></td>
<td>Algebraic simplification: <code>syms x; simplify(sin(x)^2 + cos(x)^2)</code></td>
</tr>
<tr>
<td><code>rewrite</code></td>
<td>Rewrite expression in terms of another function: <code>syms x; rewrite(tan(x)/cos(x),'sin')</code></td>
</tr>
<tr>
<td><code>resultant</code></td>
<td>Resultant of two polynomials: <code>syms x y; p = x^2+y; q = x-2*y; resultant(p,q)</code></td>
</tr>
</tbody>
</table>
## Symbolic Math Toolbox: Quick Reference Sheet

### Graphics

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>fplot</code></td>
<td>Plot symbolic expression or function:</td>
</tr>
<tr>
<td></td>
<td><code>syms x; f(x) = sin(x)/x; fplot(f)</code></td>
</tr>
<tr>
<td><code>fplot3</code></td>
<td>Plot 3-D parametric curve:</td>
</tr>
<tr>
<td></td>
<td><code>syms x; fplot3(sin(x),cos(x),log(x))</code></td>
</tr>
<tr>
<td><code>fsurf</code></td>
<td>Plot 3-D surface, mesh or contour:</td>
</tr>
<tr>
<td></td>
<td><code>syms x y; f(x,y)=x*exp(-x^2-y^2); fsurf(f)</code></td>
</tr>
<tr>
<td><code>fmesh</code></td>
<td>Plot 3-D mesh:</td>
</tr>
<tr>
<td></td>
<td><code>syms x y; f(x,y)=x*exp(-x^2-y^2); fmesh(f)</code></td>
</tr>
<tr>
<td><code>fcontour</code></td>
<td>Plot contours:</td>
</tr>
<tr>
<td></td>
<td><code>f(x,y)=x*exp(-x^2-y^2); fcontour(f)</code></td>
</tr>
<tr>
<td><code>fimplicit</code></td>
<td>Plot implicit symbolic equation or function:</td>
</tr>
<tr>
<td></td>
<td><code>syms x y; fimplicit(y^2-x^2*(x+1),[-2 2])</code></td>
</tr>
<tr>
<td><code>fimplicit3</code></td>
<td><code>syms x y z; fimplicit3(x^2*y+z+y^3-z^3)</code></td>
</tr>
<tr>
<td><code>fanimator</code></td>
<td>Create stop-motion animation object:</td>
</tr>
<tr>
<td></td>
<td><code>syms y t; fanimator(@fplot,sin(x+t),[0 t]); playAnimation</code></td>
</tr>
</tbody>
</table>

### Functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>symfun</code></td>
<td>Create Symbolic Functions:</td>
</tr>
<tr>
<td></td>
<td><code>syms x y; f = symfun(x+y,[x y]); f(1,2)</code></td>
</tr>
<tr>
<td><code>piecewise</code></td>
<td>Piecewise defined expression or function:</td>
</tr>
<tr>
<td></td>
<td><code>g(x) = piecewise(x&lt;0,-1,x&gt;=0,2); g(3)</code></td>
</tr>
<tr>
<td><code>matlabFunction</code></td>
<td>Convert symbolic expression to function handle or file:</td>
</tr>
<tr>
<td></td>
<td><code>syms x y; f = sqrt(x^2 + y^2); g = matlabFunction(f)</code></td>
</tr>
<tr>
<td><code>matlabFunctionBlock</code></td>
<td>Convert symbolic expression to MATLAB function block for Simulink:</td>
</tr>
<tr>
<td></td>
<td><code>new_system('my_system'); open_system('my_system'); syms x y z; f = x^2 + y^2 + z^2; matlabFunctionBlock('my_system/my_block',f)</code></td>
</tr>
<tr>
<td><code>simscapeEquation</code></td>
<td>Convert symbolic expression to Simscape equations:</td>
</tr>
<tr>
<td></td>
<td><code>syms t x(t) y(t); phi = diff(x) + 5*y + sin(t); simscapeEquation(phi)</code></td>
</tr>
</tbody>
</table>

### Learn More

[www.mathworks.com/products/symbolic](http://www.mathworks.com/products/symbolic)

### Related Products

**Control System Toolbox.** Design and analyze control systems.

**Financial Toolbox.** Analyze financial data and develop financial models.

**Optimization Toolbox.** Solve linear, quadratic, conic, integer, and nonlinear optimization problems.