Rutgers University
School of Engineering

Fall 2011

14:440:127 - Introduction to Computers for Engineers

Sophocles J. Orfanidis
ECE Department
orfanidi@ece.rutgers.edu

week 10

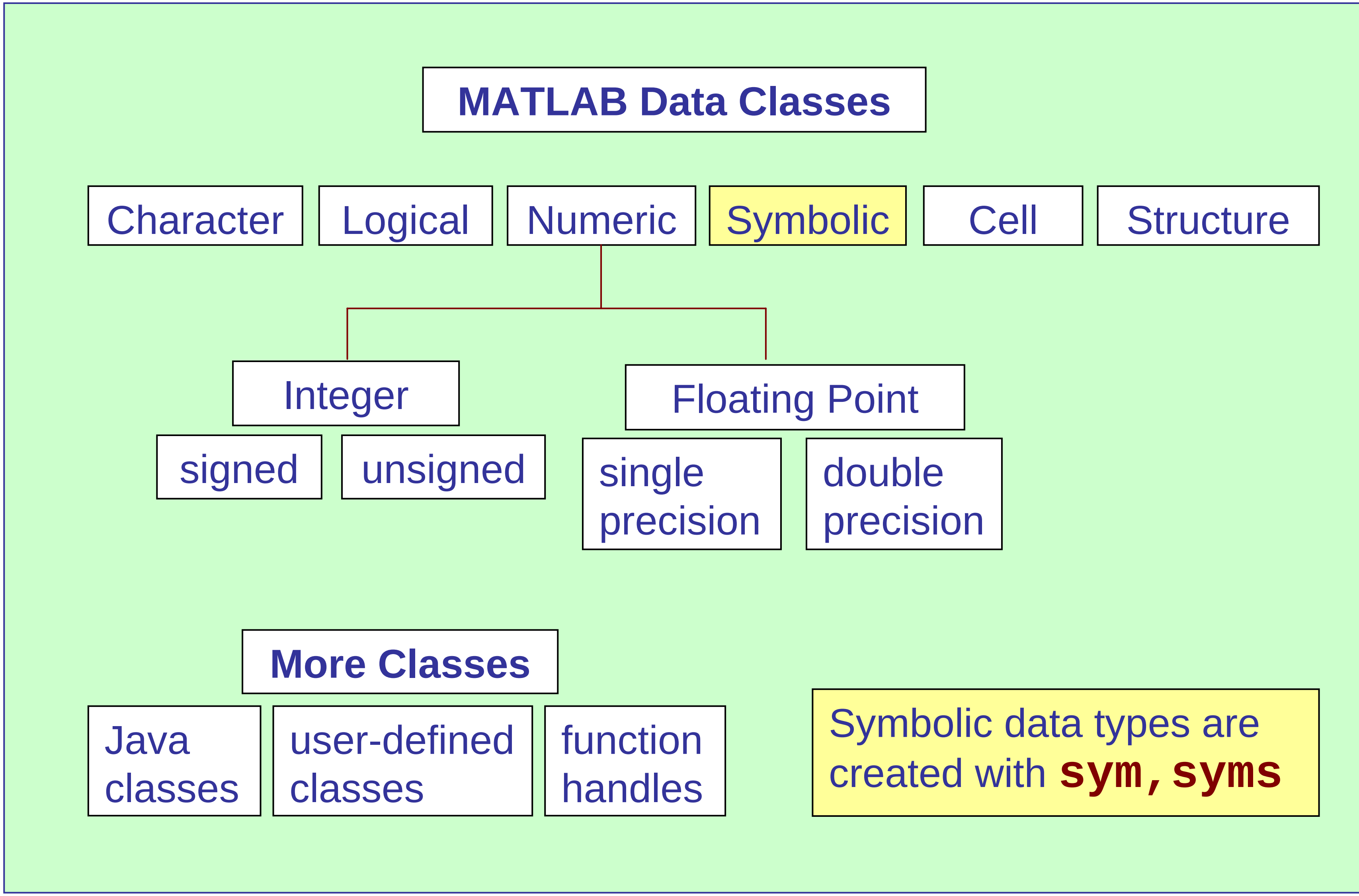
Weekly Topics

Week 1 - Basics – variables, arrays, matrices, plotting (ch. 2 & 3)
Week 2 - Basics – operators, functions, program flow (ch. 2 & 3)
Week 3 - Matrices (ch. 4)
Week 4 - Plotting – 2D and 3D plots (ch. 5)
Week 5 - User-defined functions (ch. 6)
Week 6 - Input-output processing (ch. 7)
Week 7 - Program flow control & relational operators (ch. 8)
Week 8 - Matrix algebra – solving linear equations (ch. 9)
Week 9 - Strings, structures, cell arrays (ch. 10)
Week 10 - Symbolic math (ch. 11)
Week 11 - Numerical methods – data fitting (ch. 12)
Week 12 – Selected topics

Textbook: H. Moore, *MATLAB for Engineers*, 2nd ed., Prentice Hall, 2009

Symbolic Math Toolbox

- Creating and manipulating symbolic variables
- Factoring and simplifying algebraic expressions
- Solving symbolic equations
- Linear algebra operations
- Performing summation of infinite series
- Taylor series expansions, limits
- Differentiation and integration
- Solving differential equations
- Fourier, Laplace, Z-transforms and inverses
- Variable precision arithmetic


MATLAB Data Classes
Character
Logical
Numeric
Symbolic
Cell
Structure
Integer
Floating Point
signed
unsigned
single precision
double precision
More Classes
Java classes
user-defined classes
function handles
Symbolic data types are created with sym,syms

```
>> syms x y z

>> syms x y z real
>> syms x y z positive

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

```
>> help symbolic
>> doc symbolic
>> doc sym
>> doc syms
```

```
>> f = x^6-1
f =
x^6 - 1

>> g = x^3 +1
g =
x^3 + 1
```

```
>> sym(sqrt(2))
ans =
2^(1/2)

>> sqrt(sym(2))
ans =
2^(1/2)

>> double(ans)
ans =
    1.4142
```

```
>> a = [0, pi/6, pi/4, pi/2, pi];

>> cos(a)
ans =
    1    0.8660    0.7071    0    -1

>> c = cos(sym(a))
c =
[ 1,  3^(1/2)/2,  2^(1/2)/2,  0,  -1]
```

```
>> symdisp(c)
```

display symbolic expression using LaTeX – on sakai

$$\begin{pmatrix} 1 & \frac{\sqrt{3}}{2} & \frac{\sqrt{2}}{2} & 0 & -1 \end{pmatrix}$$

```
>> pretty(c)
 +-                               -+
 |       1/2    1/2                |
 |      3      2                   |
 |  1, ----, ----, 0, -1           |
 |       2      2                  |
 +-                               -+
```

```
>> f = x^6-1;
>> g = x^3 +1;

>> r = f/g          % x^6-1 = (x^3-1)*(x^3+1)
r =
(x^6 - 1)/(x^3 + 1)

 >> pretty(r)   % pretty print

    6
   x    - 1
   -------
    3
   x  + 1

 >> symdisp(r)
```

$$\boxed{\frac{x^6 - 1}{x^3 + 1}}$$

Functions for factoring and simplification of algebraic expressions:

```
simplify     - Simplify.
expand       - Expand.
factor       - Factor.
collect      - Collect.
simple       - Search for shortest form.
numden       - Numerator and denominator.
subs         - Symbolic substitution.
```

factor

```
>> simplify(r)
ans =
x^3 - 1

>> factor(simplify(r))          % x^3 - 1
ans =
(x - 1)*(x^2 + x + 1)

>> factor(g)                    % x^3 + 1
ans =
(x + 1)*(x^2 - x + 1)

>> factor(f)                    % x^6 - 1
ans =
(x - 1)*(x + 1)*(x^2 + x + 1)* (x^2 - x + 1)
```

expand

```
>> syms x a b;
>> expand((x+1)*(x+2))
ans =
x^2 + 3*x + 2

>> expand(exp(a+b))
ans =
exp(a)*exp(b)

>> expand(cos(a+b))
ans =
cos(a)*cos(b) - sin(a)*sin(b)

>> expand(cosh(a+b))
ans =
sinh(a)*sinh(b) + cosh(a)*cosh(b)
```

expand

```
>> A = [sin(2*x), sin(3*x)
        cos(2*x), cos(3*x)]
A =
[ sin(2*x), sin(3*x)]
[ cos(2*x), cos(3*x)]

>> B = expand(A)
B =
[2*cos(x)*sin(x),   3*cos(x)^2*sin(x)-sin(x)^3]
[cos(x)^2-sin(x)^2, cos(x)^3-3*cos(x)*sin(x)^2]

>> expand((a+b)*(a^2+2*a*b+b^2))
ans =
a^3 + 3*a^2*b + 3*a*b^2 + b^3
```

collect

```
>> collect((x+1)*(x+2))
ans =
x^2 + 3*x + 2

>> collect((a+b)*(a^2 + 2*a*b + b^2), a)
ans =
a^3 + (3*b)*a^2 + (3*b^2)*a + b^3

>> collect((a+b)*(a^2 + 2*a*b + b^2), b)
ans =
b^3 + (3*a)*b^2 + (3*a^2)*b + a^3

>> factor(a^3 + 3*a^2*b + 3*a*b^2 + b^3)
ans =
(a + b)^3
```

simplify

```
>> simplify(a^3 + 3*a^2*b + 3*a*b^2 + b^3)
ans =
(a + b)^3

>> simplify(cos(b)*sin(a) - cos(a)*sin(b))
ans =
sin(a - b)

B =
[2*cos(x)*sin(x),   3*cos(x)^2*sin(x)-sin(x)^3]
[cos(x)^2-sin(x)^2, cos(x)^3-3*cos(x)*sin(x)^2]

>> simplify(B)
ans =
[ sin(2*x), sin(3*x)]
[ cos(2*x), cos(3*x)]
```

numden

```
>> [num,den] = numden(1 + 2*x + 3/(x^2+5))

num =
2*x^3 + x^2 + 10*x + 8
den =
x^2 + 5

>> symdisp(1 + 2*x + 3/(x^2+5))
```

$$2\,x + \frac{3}{x^2+5} + 1$$

```
>> symdisp(num/den)
```

$$\frac{2\,x^3 + x^2 + 10\,x + 8}{x^2+5}$$

subs

```
>> syms x a b
>> y = a*x+b
y =
b + a*x
```

x,a,b class sym

```
>> y1 = subs(y,a,3)
y1 =
b + 3*x
>> y2 = subs(y,b,5)
y2 =
a*x + 5

>> y3 = subs(y,{a,b},{3,5})
y3 =
3*x + 5
```

a,b taken from workspace

```
>> a=2; b=4; y4 = subs(y)
y4 =
2*x + 4
```

a,b class double

Functions for solving algebraic and differential equations:

```
solve        - solve algebraic equations.
dsolve       - solve differential equations.
finverse     - Functional inverse.
compose      - Functional composition.
```

Solving symbolic equations

```
>> syms x a b c;
>> f = a*x^2 + b*x + c;

>> xsol = solve(f)
xsol =
 -(b + (b^2 - 4*a*c)^(1/2))/(2*a)
 -(b - (b^2 - 4*a*c)^(1/2))/(2*a)

>> xsol(1), xsol(2)
ans =
-(b + (b^2 - 4*a*c)^(1/2))/(2*a)
ans =
-(b - (b^2 - 4*a*c)^(1/2))/(2*a)
```

```
symdisp(xsol(1));
```

$\longrightarrow$

$$-\frac{b+\sqrt{b^2-4\,a\,c}}{2\,a}$$

```
>> g = a*(x-xsol(1))*(x-xsol(2))
g =
a*(x + (b + (b^2 - 4*a*c)^(1/2))/(2*a))*(x +
(b - (b^2 - 4*a*c)^(1/2))/(2*a))

>> simplify(g)
ans =
a*x^2 + b*x + c


>> a*xsol(1)^2 + b*xsol(1) + c
ans =
 c + (b + (b^2 - 4*a*c)^(1/2))^2/(4*a) -
(b*(b + (b^2 - 4*a*c)^(1/2)))/(2*a)

>> simplify(a*xsol(1)^2 + b*xsol(1) + c)
ans =
 0
```

some variations

```
>> xsol = solve('a*x^2 + b*x + c');
>> xsol = solve('a*x^2 + b*x + c=0');
>> xsol = solve('a*x^2 + b*x + c','x');
>> xsol = solve(a*x^2 + b*x + c, x);

>> syms z;
>> xsol = solve(a*z^2 + b*z + c)
xsol =
 -(b + (b^2 - 4*a*c)^(1/2))/(2*a)
 -(b - (b^2 - 4*a*c)^(1/2))/(2*a)

>> bsol = solve(a*x^2 + b*x + c, b)
bsol =
-(a*x^2 + c)/x
```

solving equations

```
>> syms a b p u w y z
>> solve(a+b+p), solve(a+b+w)
ans =
- a - b
>> solve(u+w+z)
ans =
- u - z
>> solve(w+y+z)
ans =
- w - z
```

how does it know which variable to solve for?

```
>> syms x a b c;
>> f = a*x^2 + b*x + c;

>> xsol = solve(f)
xsol =
 -(b + (b^2 - 4*a*c)^(1/2))/(2*a)
 -(b - (b^2 - 4*a*c)^(1/2))/(2*a)
```

solving equations

```
>> f1 = subs(f,{a,b,c},{1,1,-1})
f1 =
x^2 + x – 1

>> x1 = solve(f1)
x1 =
   5^(1/2)/2 - 1/2
 - 5^(1/2)/2 – 1/2

>> subs(xsol,{a,b,c},{1,1,-1})
ans =
   -1.6180
    0.6180

>> phi = simplify(1./x1)
phi =
 5^(1/2)/2 + 1/2
 1/2 - 5^(1/2)/2
```

x1 class sym

class double

golden ratio

Fibonacci numbers, $F(n) = F(n\text{-}1)+F(n\text{-}2)$
$= [0, 1, 1, 2, 3, 5, 8, 13, \ldots]$

```
>> syms n
>> F = (phi(1)^n – phi(2)^n)/(phi(1)-phi(2));

>> simplify(F - subs(F,n,n-1) - subs(F,n,n-2))
ans =
0

>> limit(F/subs(F,n,n-1), n, inf)
ans =
2/(5^(1/2) - 1)

>> simplify(limit(F/subs(F,n,n-1), n,inf))
ans =
5^(1/2)/2 + 1/2
```

golden ratio

solving equations

```
>> eq1 = 'x^2 + x -1=0';
>> solve(eq1)
ans =
   5^(1/2)/2 - 1/2
 - 5^(1/2)/2 - 1/2

>> eq2 = '1/x = x/(1-x)';
>> solve(eq2)
ans =
   5^(1/2)/2 - 1/2
 - 5^(1/2)/2 - 1/2

>> solve('1/x = x/(1-x)');
```

```
>> eq1 = 'x^2 + y - 1=0';
>> eq2 = 'x+y=0';
>> [x,y] = solve(eq1,eq2)

x =
 5^(1/2)/2 + 1/2
 1/2 - 5^(1/2)/2
y =
 - 5^(1/2)/2 - 1/2
   5^(1/2)/2 – 1/2

>> eval(x)   % or, double(x)
ans =
    1.6180
   -0.6180
```

```
>> [x,y] = solve('x+y=1', 'x^2+y^2=1');
>> [x,y]
ans =
[ 1, 0]
[ 0, 1]

>> S = solve('x+y=1', 'x^2+y^2=1')
S =
    x: [2x1 sym]
    y: [2x1 sym]

>> [S.x, S.y]
ans =
[ 1, 0]
[ 0, 1]
```

returned as a structure of syms

linear equations

```
>> A = [ 5  1  3  0
         1  4  1  1
        -1  2  6 -2
         1 -1  1  4];
```

from homework-9

```
>> A = sym(A)
A =
[  5,  1, 3,  0]
[  1,  4, 1,  1]
[ -1,  2, 6, -2]
[  1, -1, 1,  4]

>> inv(A)
ans =
[  53/274, -2/137, -12/137, -11/274]
[ -21/548, 34/137,  -3/274, -37/548]
[  13/548, -8/137,  41/274,  49/548]
[ -35/548, 11/137,  -5/274, 121/548]
```

```
>> b = [12; -3; 11; 10];

>> x = inv(A)*b      % or, A\b

x =
   1
  -2
   3
   1

>> class(x)
ans =
sym
```

```
>> f = 5*x^4 - 2*x^3 + x^2 + 4*x + 3
f =
5*x^4 - 2*x^3 + x^2 + 4*x + 3

>> p = sym2poly(f)
p =
     5    -2     1     4     3

>> poly2sym(p)
ans =
5*x^4 - 2*x^3 + x^2 + 4*x + 3

>> poly2sym(p,'t')
ans =
5*t^4 - 2*t^3 + t^2 + 4*t + 3
```

polynomials

poly2sym
sym2poly
coeffs
quorem

roots
poly

```
>> [q,mono] = coeffs(f)
q =
[ 5, -2, 1, 4, 3]
mono =
[ x^4, x^3, x^2, x, 1]

>> p = sym2poly(f)
p =
     5    -2     1     4     3

>> z = roots(p)
z =
   0.7393 + 0.8967i
   0.7393 - 0.8967i
  -0.5393 + 0.3916i
  -0.5393 - 0.3916i
```

polynomials

poly2sym
sym2poly
coeffs
quorem

roots
poly

```
>> p = sym2poly(f)
p =
     5    -2     1     4     3

>> z = roots(p)
z =
   0.7393 + 0.8967i
   0.7393 - 0.8967i
  -0.5393 + 0.3916i
  -0.5393 - 0.3916i

>> P = poly(z)
P =
    1.0   -0.4    0.2    0.8    0.6
>> 5*P
ans =
     5    -2     1     4     3
```

polynomials

poly2sym
sym2poly
coeffs
quorem

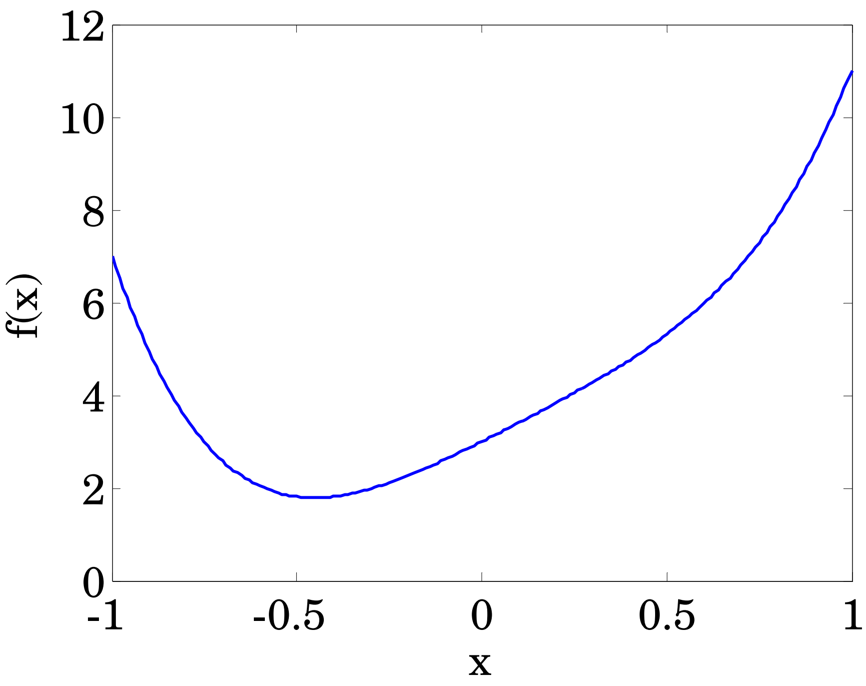
roots
poly

```
>> f = 5*x^4 - 2*x^3 + x^2 + 4*x + 3;

>> x1 = linspace(-1,1,201);

>> f1 = subs(f,x1);    % or, f1 = subs(f,x,x1);

>> plot(x1,f1,'b');
>> xlabel('x');
>> ylabel('f(x)');
```



symsum

$$\sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{\pi^2}{6} \qquad \sum_{n=1}^{\infty} \frac{1}{n^4} = \frac{\pi^4}{90}$$

```
>> syms n

>> symsum(1/n^2, n, 1, inf)
ans =
pi^2/6

>> symsum(1/n^4, n, 1, inf)
ans =
pi^4/90
```

symsum

```
>> syms x n N

>> symsum(x^n, n, 0, N-1)
piecewise([x = 1, N], [x <> 1,...
 (x^N - 1)/(x - 1)]])

>> symsum(x^n, n, 0, inf)
piecewise([1 <= x, Inf],...
[abs(x) < 1, -1/(x - 1)]])

>> symsum(x^n/sym('n!'), n, 0, inf)
ans =
exp(x)

>> symsum(n^2, n, 1, N)
ans =
(N*(2*N + 1)*(N + 1))/6
```

finite & infinite geometric series

Taylor series

```
>> syms x

>> taylor(exp(x))
ans =
x^5/120 + x^4/24 + x^3/6 + x^2/2 + x + 1

>> taylor(exp(x),4)
ans =
x^3/6 + x^2/2 + x + 1

>> taylor(sin(x),4)
ans =
x - x^3/6

>> taylor(sin(x)/x)
ans =
x^4/120 - x^2/6 + 1
```

limits

```
>> syms x

>> limit(exp(-x), x, inf)
ans =
0

>> limit(sin(x)/x, x, 0)
ans =
1

>> limit(cos(pi*cos(x)/2)/x, x, 0)
ans =
0

>> taylor(cos(pi*cos(x)/2)/x,2)
ans =
(pi*x)/4
```

$$\frac{\cos\left(\frac{\pi\ \cos(x)}{2}\right)}{x}$$

simple calculus

differentiation
integration

```
>> syms x

>> diff(x^2)
ans =
2*x

>> diff(x^3)
ans =
3*x^2

>> diff(cos(x))
ans
-sin(x)
```

```
>> syms x

>> int(x^2)
ans =
x^3/3

>> int(cos(x))
ans =
sin(x)

>> int(cos(x)^2, 0, pi)
ans =
pi/2
```

$$\int_0^{\pi} \cos^2(x)\,dx = \frac{\pi}{2}$$

$$\boxed{\frac{dv(t)}{dt} = a - C\,v^2(t)}$$

acceleration with air drag

dsolve

$$v_a = \sqrt{\frac{a}{C}}, \quad \tau_a = \frac{1}{\sqrt{aC}}$$

terminal velocity and time constant

$$\frac{dv(t)}{dt} = \frac{v_a}{\tau_a}\left[1 - \frac{v^2(t)}{v_a^2}\right]$$

solve with initial velocity $v(0) = v_0$

$$v(t) = v_a \frac{v_0 + v_a \tanh(t/\tau_a)}{v_a + v_0 \tanh(t/\tau_a)}$$

$$v(t) = v_a \tanh\left[\frac{t}{\tau_a} + \operatorname{atanh}\left(\frac{v_0}{v_a}\right)\right]$$

exact solution

$$v(0) = v_0$$

$$v(\infty) = v_a$$

D = derivative

dsolve

```
>> syms v t va ta v0

>> diffeq = 'Dv=va/ta*(1-v^2/va^2)';

>> v = dsolve(diffeq, 'v(0)=v0')
v =
-va*tan(-va*(- t/(ta*va) +
(atan((v0*i)/va)*i)/va)*i)*i

>> v = simplify(v)
v =
va^2/v0 +
(va*(v0^2 - va^2))/(v0*(va + v0*tanh(t/ta)))

>> [num,den] = numden(v);
```

dsolve

```
>> v = num/den
v =
(tanh(t/ta)*va^2 + v0*va)/(va + v0*tanh(t/ta))

>> symdisp(v);
```

$$\longrightarrow \quad \frac{\tanh\left(\frac{t}{\mathrm{ta}}\right)\ \mathrm{va}^2 + \mathrm{v0}\,\mathrm{va}}{\mathrm{va} + \mathrm{v0}\ \tanh\left(\frac{t}{\mathrm{ta}}\right)}$$

```
>> u = va*tanh(t/ta + atanh(v0/va));
>> simplify(v-u)
ans =
0
```

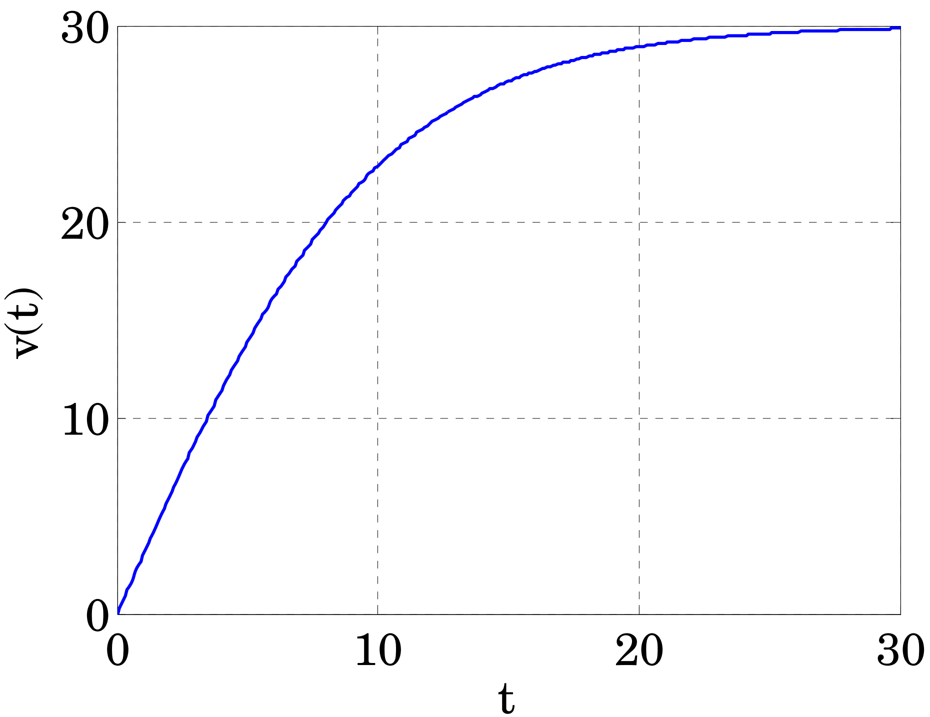
$$v(t) = v_a \frac{v_0 + v_a \tanh(t/\tau_a)}{v_a + v_0 \tanh(t/\tau_a)}$$

$$v(t) = v_a \tanh\left[\frac{t}{\tau_a} + \operatorname{atanh}\left(\frac{v_0}{v_a}\right)\right]$$

dsolve

```
>> subs(v,t,0)
ans =
v0

>> t1 = linspace(0,30,301);
>> v1 = subs(v, {va,ta,v0,t}, {30,10,0,t1});
>> plot(t1,v1,'b');
>> xlabel('t');
>> ylabel('v(t)');
```



dsolve

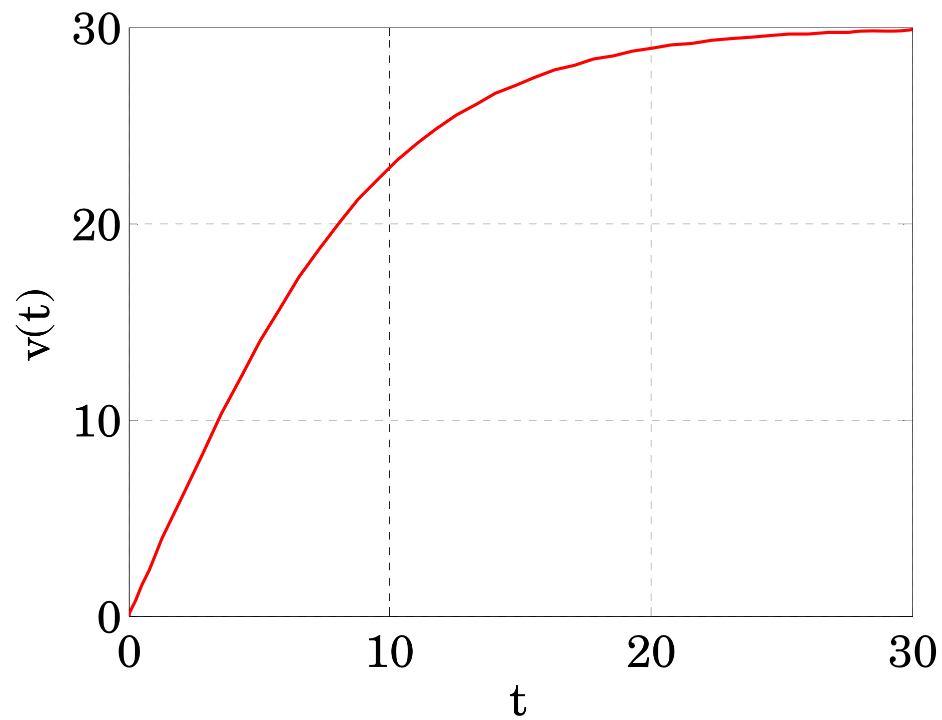
```
v0 = 0; va = 30; ta = 10;
tspan = [0,30];

vdot = @(t,v) va/ta * (1 - v.^2/va^2);

[t2,v2] = ode45(vdot, tspan, v0);

plot(t2,v2,'r-')
xlabel('t');
ylabel('v(t)');
```

compare exact and numerical solutions using **ode45**



compare exact and numerical solutions using forward Euler method, which replaces derivatives by forward differences

dsolve

$$\frac{v(n+1)-v(n)}{T} = \frac{v_a}{\tau_a}\left[1 - \frac{v^2(n)}{v_a^2}\right]$$

$$v(n+1) = v(n) + T\,\frac{v_a}{\tau_a}\left[1 - \frac{v^2(n)}{v_a^2}\right]$$

dsolve

```
Tspan = 30; N = 60; T = Tspan/N;
tn = 0:T:Tspan;
vn(1) = v0;

for n=1:N,
   vn(n+1) = vn(n) + T*va/ta*(1-vn(n)^2/va^2);
end

plot(t1,v1,'b');
hold on;
plot(tn,vn,'r--')
```

compare exact and numerical solutions using forward Euler