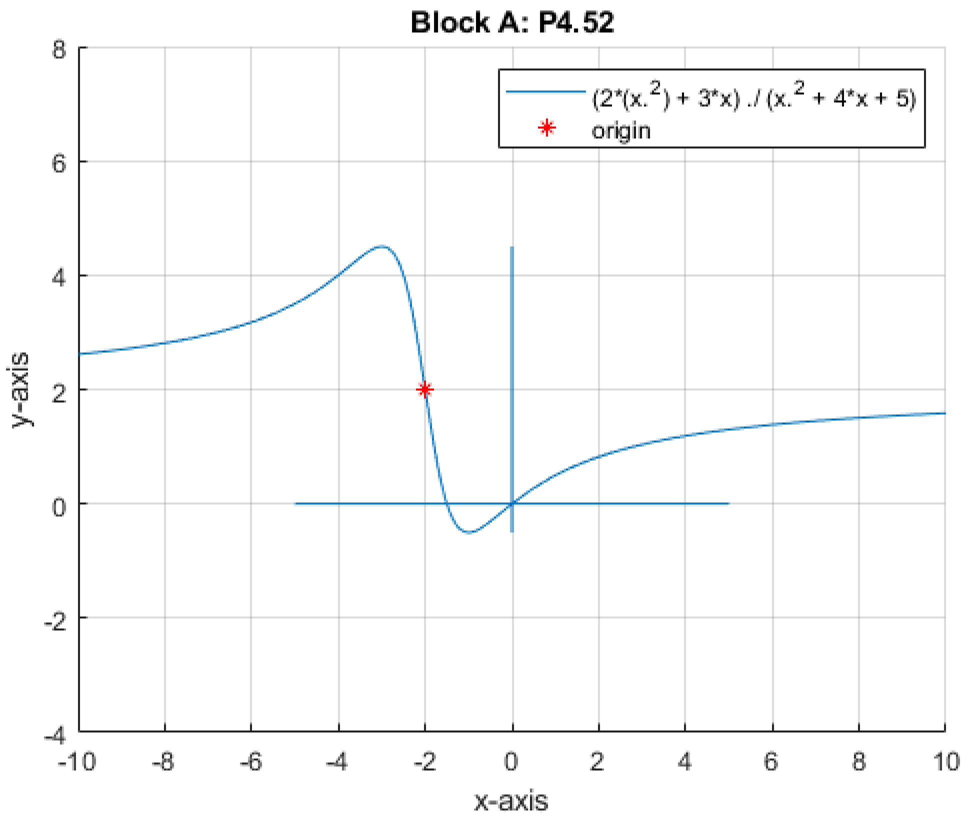
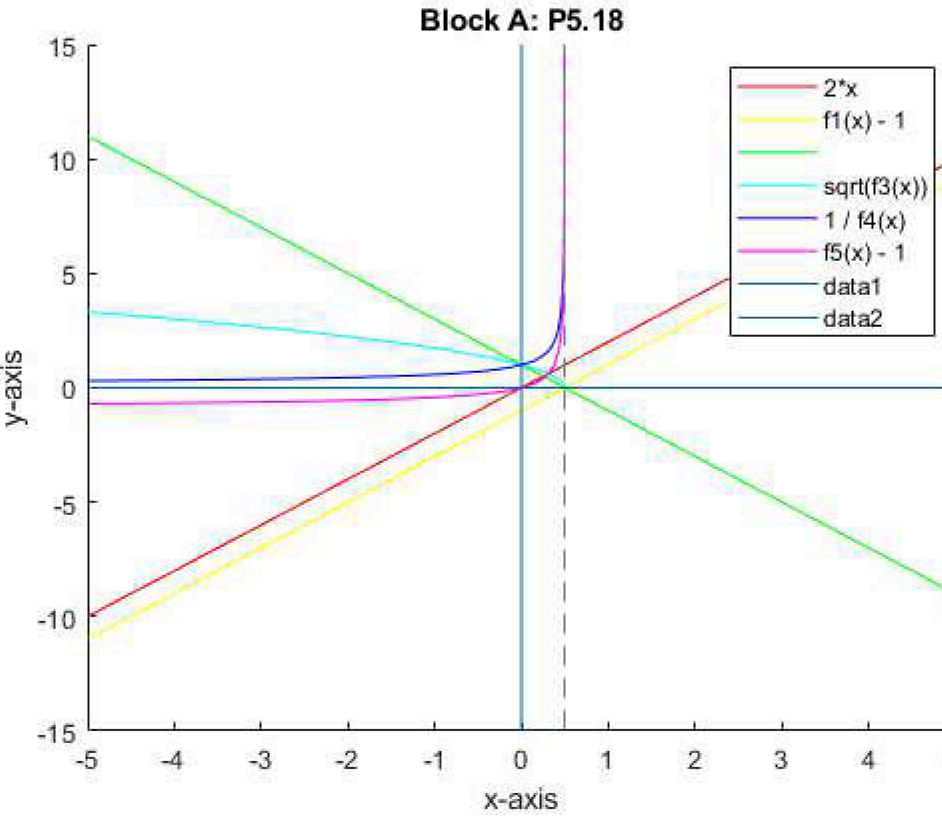
```
% plot function
f = @(x) (2*(x.^2) + 3*x) ./ (x.^2 + 4*x + 5);
hold on
fplot(f)
plot(-2, f(-2), 'r*') % origin
line(xlim, [0, 0])
line([0, 0], ylim)
hold off

% style plot
grid on
axis([-10 10 -4 8])
legend("(2*(x.^2) + 3*x) ./ (x.^2 + 4*x + 5)", "origin");
title("Block A: P4.52")
xlabel("x-axis")
ylabel("y-axis")
```



Published with MATLAB® R2020b

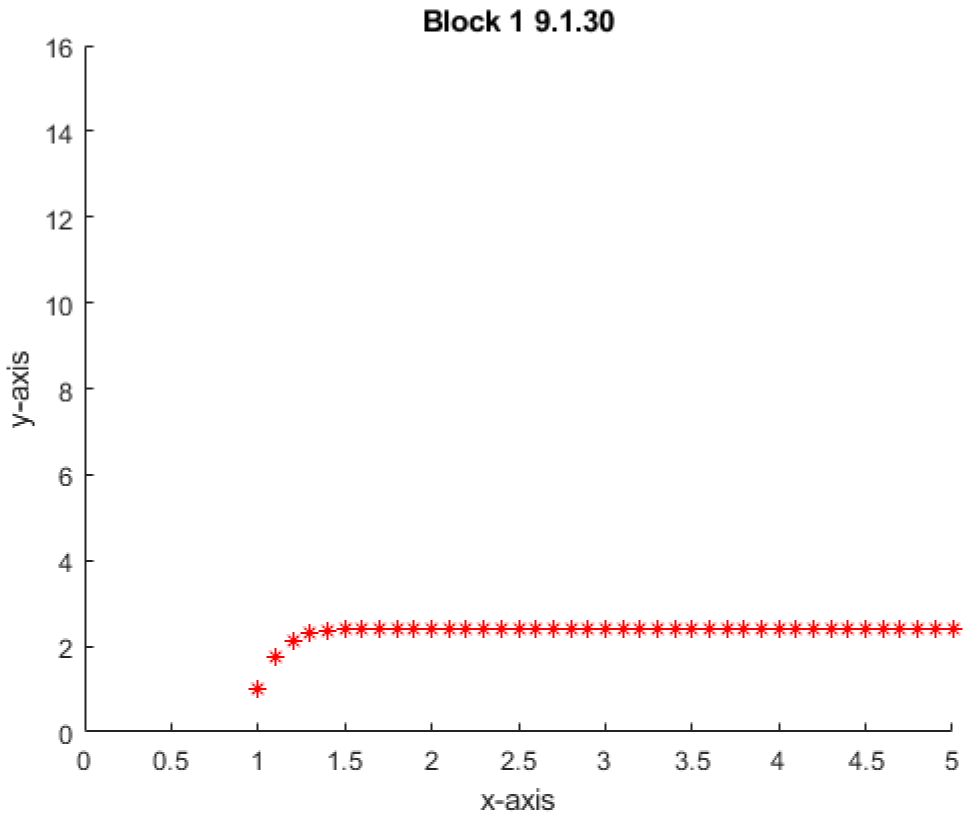
```
% functions
f1 = @(x) 2*x;            % start; no change
f2 = @(x) f1(x) - 1;      % shift graph done 1 unit on y-axis
f3 = @(x) 1 - f1(x);      % invert f2
f4 = @(x) sqrt(f3(x));    % change shape and domain
f5 = @(x) 1 ./ (f4(x));      % change shape of function
f6 = @(x) f5(x) - 1;      % shift f5 down

% plot functions
hold on
fplot(f1, "r")
fplot(f2, "y")
fplot(f3, "g")
fplot(f4, "c")
fplot(f5, "b")
fplot(f6, "m")
hold off

% style
legend("2*x", "f1(x) - 1", "", "sqrt(f3(x))", "1 / f4(x)", "f5(x) -
 1")
title("Block A: P5.18")
line(xlim, [0, 0])
line([0, 0], ylim)
ylabel("y-axis")
xlabel("x-axis")
```



Published with MATLAB® R2020b

```
%Show that {a_n} is increasing and has an upper bound, conclude that
%the sequence converges and find its limit.
%Let a_1 = 1 and a_n+1 = sqrt(1+2a_n) (n = 1,2,3,...)

i=1; %set startvalue of loop
a=0; %set startvalue of a
axis([0 5 0 16]) %set zoomlevel
hold on
while i < 15 %loop enough times so upper bound is visible
    a = sqrt(1+(a*2));
    plot(i,a, "r*") %set points in graph
    %fprintf("Loop %f: %f\n", i, a)
    i=i+0.1; %increase with each loop i
end
hold off
xlabel('x-axis')
ylabel('y-axis')
title('Block 1 9.1.30')
fprintf("final value of a is %f\n", a)
%While the upper bound goes to 3, the value of y does not exceed
 2.4142
```

```
final value of a is 2.414214
```



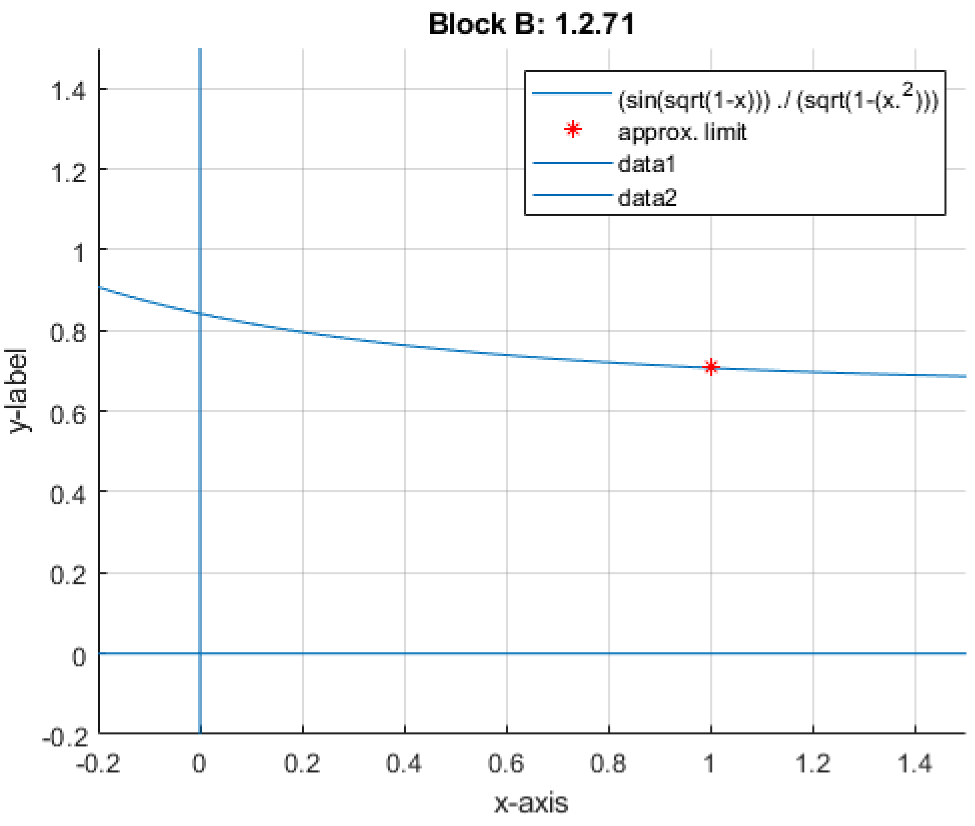
```
% Block B: D.1.31

% array of number 1,2,3,...,30
arr = 0:29;

% Solution 1
% create a list of twos with equal length to arr and for each element
% raise to opposing element in arr
arr2 = 2.*ones([1, 30]);
arr = arr2.^arr;

% Solution 2
% loop through and change each number to the power of two for that
 index
% (double each iteration)
%for i = 1:30
%   arr(i) = 2^(i-1);
%end

% Solution 3
% same as solution 1 but with a slick one-liner
% note: initial arr needs to be commented out
%arr = (2.*ones([1, 30])).^(0:29);

% add all list values togheter and print
s = sum(arr);
fprintf("Final value is %i", s)
```

```
Final value is 1073741823
```

Published with MATLAB® R2020b

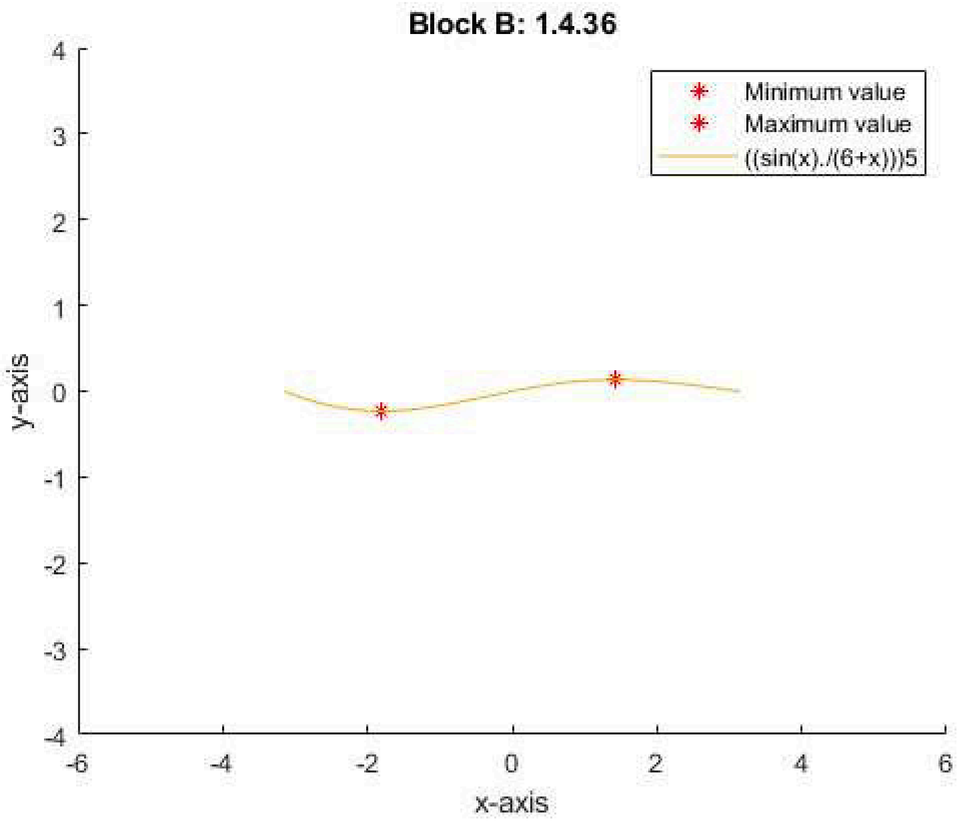
```
format long % enable 4 decimal place precision

% plot
f = @(x) (sin(sqrt(1-x))) ./ (sqrt(1-(x.^2))); % "sin" is radians by
 default
hold on
fplot(f);
plot(0.999, f(0.999), 'r*') % read x value from graph plot
hold off

% print result
fprintf("when x->1-  then y reaches %f", f(0.999))

% style
grid on
axis([-0.2 1.5 -0.2 1.5])
legend("(sin(sqrt(1-x))) ./ (sqrt(1-(x.^2)))", "approx. limit")
title("Block B: 1.2.71")
line(xlim, [0, 0])
line([0, 0], ylim)
xlabel("x-axis")
ylabel("y-label")
```

```
when x->1-  then y reaches 0.707166
```

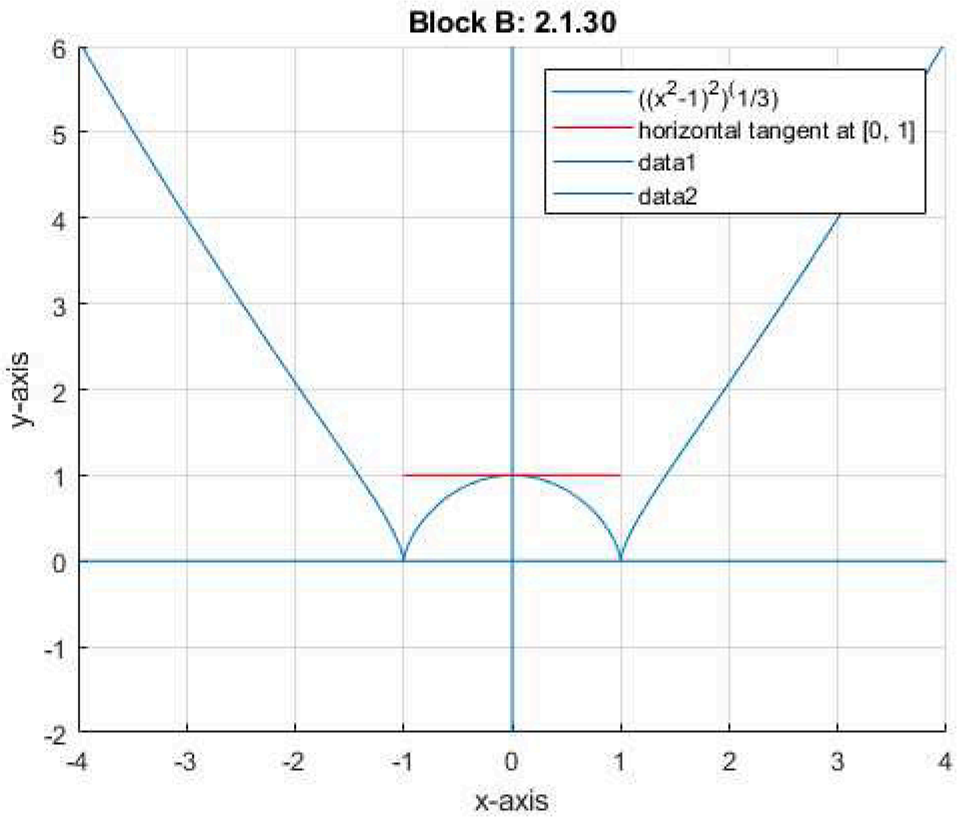


```
% plot graph
hold on
f = @(x) ((sin(x)./(6+x)));      %initial function
g = @(x) -(f(x));                %flip graph
minb = fminbnd(f, -pi, pi);      %values of x,y + set minimum
maxb = fminbnd(g, -pi, pi);      %values of x,y + set maximum
plot(minb, f(minb), 'r*')        %set red star to min
plot(maxb, f(maxb), 'r*')        %same but max
fplot(f,[-pi,pi]);               %draw the curve
hold off

% print points
fprintf("min: [%f, %f]\n", minb, f(minb))
fprintf("max: [%f, %f]\n", maxb, f(maxb))

% style
axis([-6 6 -4 4])
title('Block B: 1.4.36')
legend("Minimum value", "Maximum value", "((sin(x)./(6+x)))5")
xlabel('x-axis')
ylabel('y-axis')
```

```
min: [-1.804811, -0.231871]
max: [1.437121, 0.133261]
```



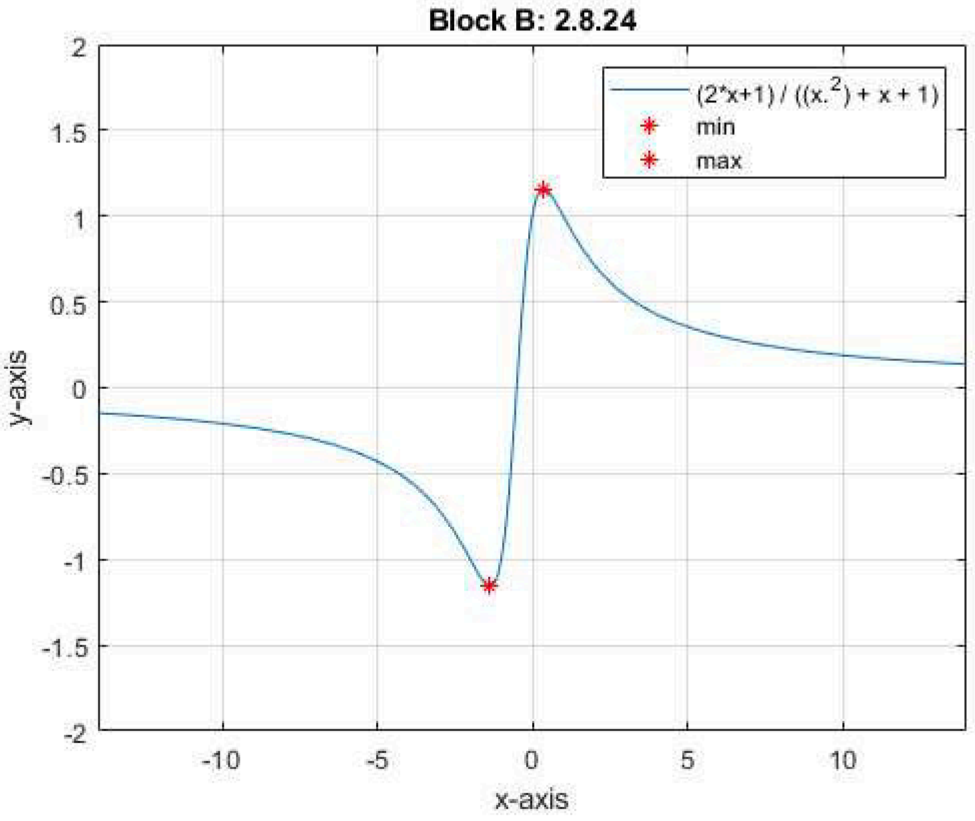
```
% plot function
hold on
grid on %draw grid
f = @(x) ((x.^2-1).^2).^(1./3);
fplot(f)
line ([-1 1],[1 1],'color','r') %draw tangent from x=-1 to x=1 on y=1
hold off

% style
title('Block B: 2.1.30')
legend("((x^2-1)^2)^(1/3)", "horizontal tangent at [0, 1]")
xlabel('x-axis')
ylabel('y-axis')
axis([-4 4 -2 6]) %zoom factor
line ([0 0], ylim); %draw x-axis
line (xlim, [0 0]); %draw y-axis
```



Published with MATLAB® R2020b

```
format long % 6 decimal places

% plot function
f = @(x) (2*x+1) ./ ((x.^2) + x + 1);
g = @(x) -(f(x)); % inverse f
fplot(f)
grid on
axis([-14 14 -2 2]) % assuming trend continues, critical points
 ( f'(x)=0 )
% should all be in the open interval (-14, 14)

% calculate critical points
% function has a minimum and a maximum value, the 2 critical points
bnd = 14;
fmin = fminbnd(f, -bnd, bnd);
fmax = fminbnd(g, -bnd, bnd);
fprintf("Min: [%f, %f]\n", fmin, f(fmin))
fprintf("Max: [%f, %f]\n", fmax, f(fmax))

% mark out critical points
hold on
plot(fmin, f(fmin), 'r*')
plot(fmax, f(fmax), 'r*')
hold off

% style
title("Block B: 2.8.24")
legend("(2*x+1) / ((x.^2) + x + 1)", "min", "max")
xlabel("x-axis")
ylabel("y-axis")
```

```
Min: [-1.366026, -1.154701]
Max: [0.366025, 1.154701]
```



Published with MATLAB® R2020b

```
format long

% y = g^-1(x) <=> x = g(y):
% =>  g^-1(x) = y = where g(y) = 2
% y^-1' = 1 / g'(y)

% calculate g^-1(x)=2
g = @(y) 2*y + sin(y); % increases; one-to-one; invertable
% g(y) = 2 => g(y) - 2 = 0  =>  2y + siny - 2 = 0
g2 = @(x) g(x) - 2;
z = fzero(g2, 1);
fprintf("g^-1(x)=2 when y=%f\n", z)

% calculate derivative of g^-1(2)
% ... = 1/g'(y)
gp = @(y) 2 + cos(y); % derivative of g
dv = 1 / (gp(z));
fprintf("derivative at y=2 is %f\n", dv)
```

```
g^-1(x)=2 when y=0.684037
derivative at y=2 is 0.360357
```

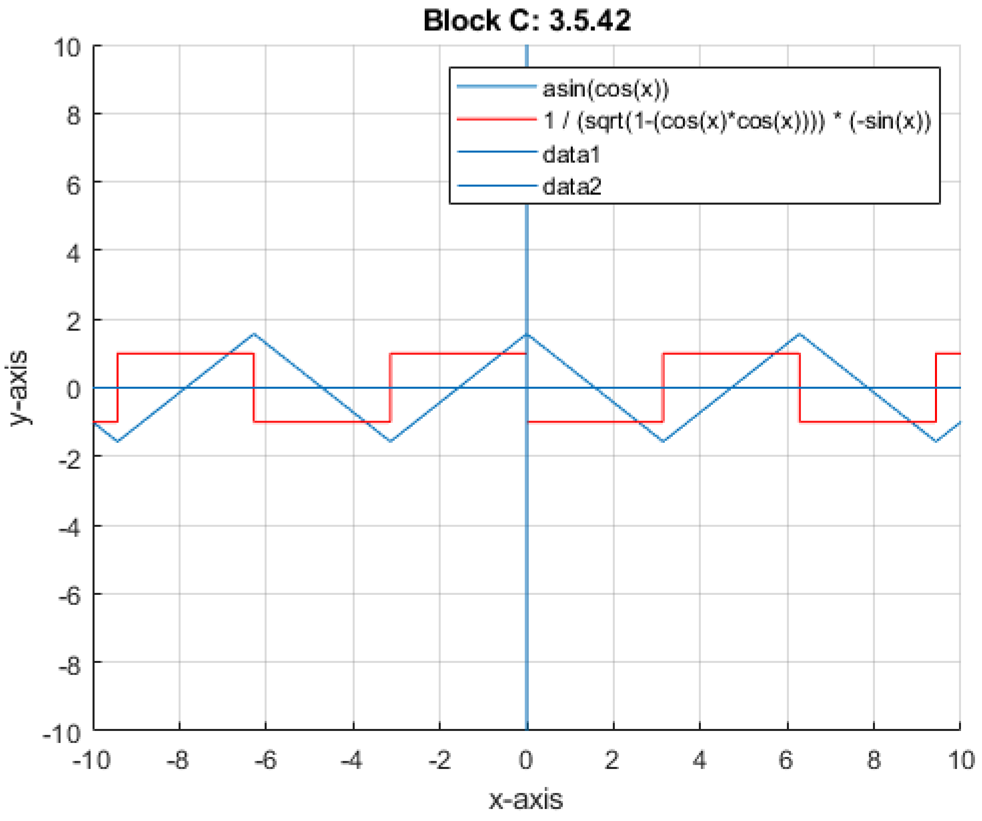
Published with MATLAB® R2020b

```
% x^sqrt(2) = 3  =>  sqrt(2)*log_10*x = log_10 * 3
% => x = 10^(log_10*3 / sqrt*2)
x = 10.^(log10(3)./sqrt(2));
fprintf('Value is %d', x)
%prints answer: x = 2.174581
```

```
Value is 2.174581e+00
```

Published with MATLAB® R2020b

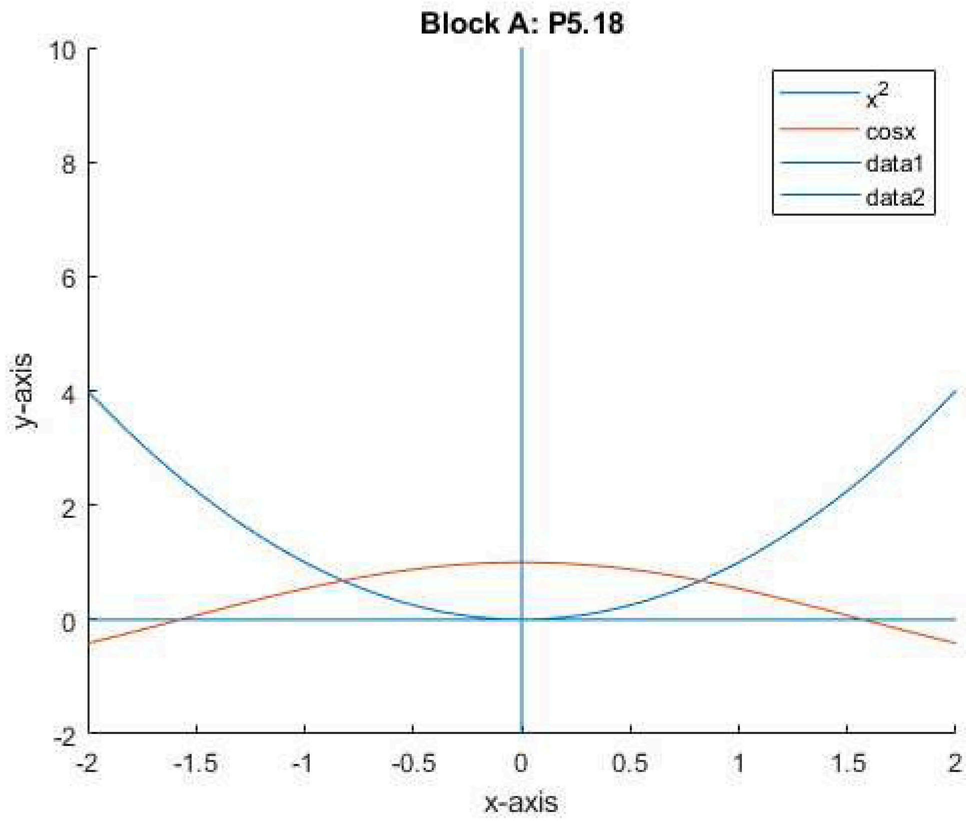
```
f = @(x) asin(cos(x));

% D asin(cos(x)) = 1/(sqrt(1-cos^2(x))) * (- sin(x))
fp = @(x) 1 ./ (sqrt(1-(cos(x).*cos(x)))) .* (-sin(x));

% plot function
hold on
grid on
fplot(f);
fplot(fp, 'r')
hold off

% since sin is continous on interval -1 to 1 and stays the same for
% all real numbers it stays continous for the whole real line

% derivate is undefined when sqrt(1-cos^2x) equals 0
% => when 1-cos^2x = 0
% => cos^2x = 1
% since cos^2x is just cosx * cosx, cosx needs to be 1 (1^2 = 1)
% cosx = 1 when n*pi, n is a positive integer

%style
title('Block C: 3.5.42')
legend("asin(cos(x))", "1 / (sqrt(1-(cos(x)*cos(x)))) * (-sin(x))")
xlabel("x-axis")
ylabel("y-axis")
axis([-10 10 -10 10])
line([0 0], ylim)
line(xlim, [0 0])
```



Published with MATLAB® R2020b

```
% cosx = x^2  => x^2 - cosx = 0
f = @(x) x^2 - cos(x);

% find derivative of f
fp = @(x) 2 + sin(x);

% plot out cos x and x^2, analyze for a suggested solution
% when both graphs meet (x^2=cosx) we see a value of ~0.8
hold on
fplot(@(x) x.^2)
fplot(@(x) cos(x))
hold off

% solve through newton-raphson; repeat this function
nr = @(x0) x0 - (f(x0) ./ (fp(x0)));
x0 = 0.8;
for i = 1:5
    x0 = nr(x0);
    %fprintf("%f\n", x0)
end

fprintf("After 5 iterations we get the result %f\n", x0)

% we can also note that the function will have another zero-value at
 x=-x0

% style
axis([-2 2 -2 10])
legend("x^2", "cosx")
title("Block A: P5.18")
line(xlim, [0, 0])
line([0, 0], ylim)
ylabel("y-axis")
xlabel("x-axis")
```

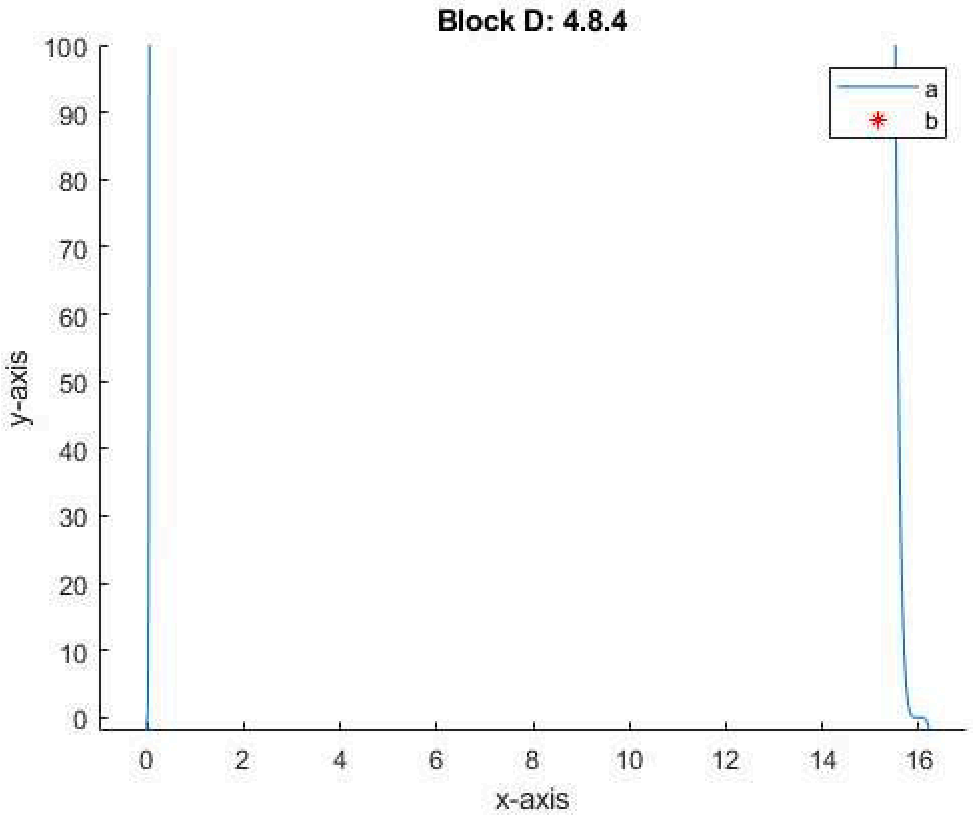
```
After 5 iterations we get the result 0.824131
```



Published with MATLAB® R2020b

```
% x + y = 16 ; f = y^3 * x^5
% => y = 16-x
% => f(x) = x^3 * (16-x)^5
f = @(x) x.^3 .* (16-x).^5;

% calculate f prime and decide maximum value
fp = @(x)  (x.^2)*((16-x).^4) * (48-(8*x)); % f prime; derivative of f
fpx = fzero(fp, 16/2); % start search at middle
fprintf("x: %d,  y: %d\n", fpx, f(fpx))

% plot function
hold on
fplot(f)
plot(fpx, f(fpx), 'r*')
hold off

% style
axis([-1 17 -2 100])
title("Block D: 4.8.4")
legend("a", "b")
xlabel("x-axis")
ylabel("y-axis")
```

```
x: 6,  y: 21600000
```



```
format long

% maclaurin gives us the following formula for e^x as x->0:
% 1 + x + x^2/2! + x^3/3! + ... + x^n/n!
% since we want 1/e, we can see that e^-1 (x=1) gives 1(e), therefore:
%   x = 1
% we can also note that for e^-x the sign of the iteration will vary
Pn = @(n) ((-1)^n) * (1 / factorial(n));

% find iteration count n for a set precision
n = 1;
p = 1;
while p > 0.000005
   n = n + 1;
   p = 1/factorial(n+1);
end
fprintf("number of iterations is %d\n", n)

% use n to iterate over f
% note that since for our function e^-x, Pn(x) = 1 - 1 + 1/2! - ...
% the first two terms equal 0, and the loop should start at 2
r = 0;
for i = 2:n+1
    r = r + Pn(i);
    %fprintf("%d: %f\n", i, r)
end
fprintf("Final value is %0.5f\n", r)
```

```
number of iterations is 8
Final value is 0.36788
```

Published with MATLAB® R2020b