Weibull Distribution

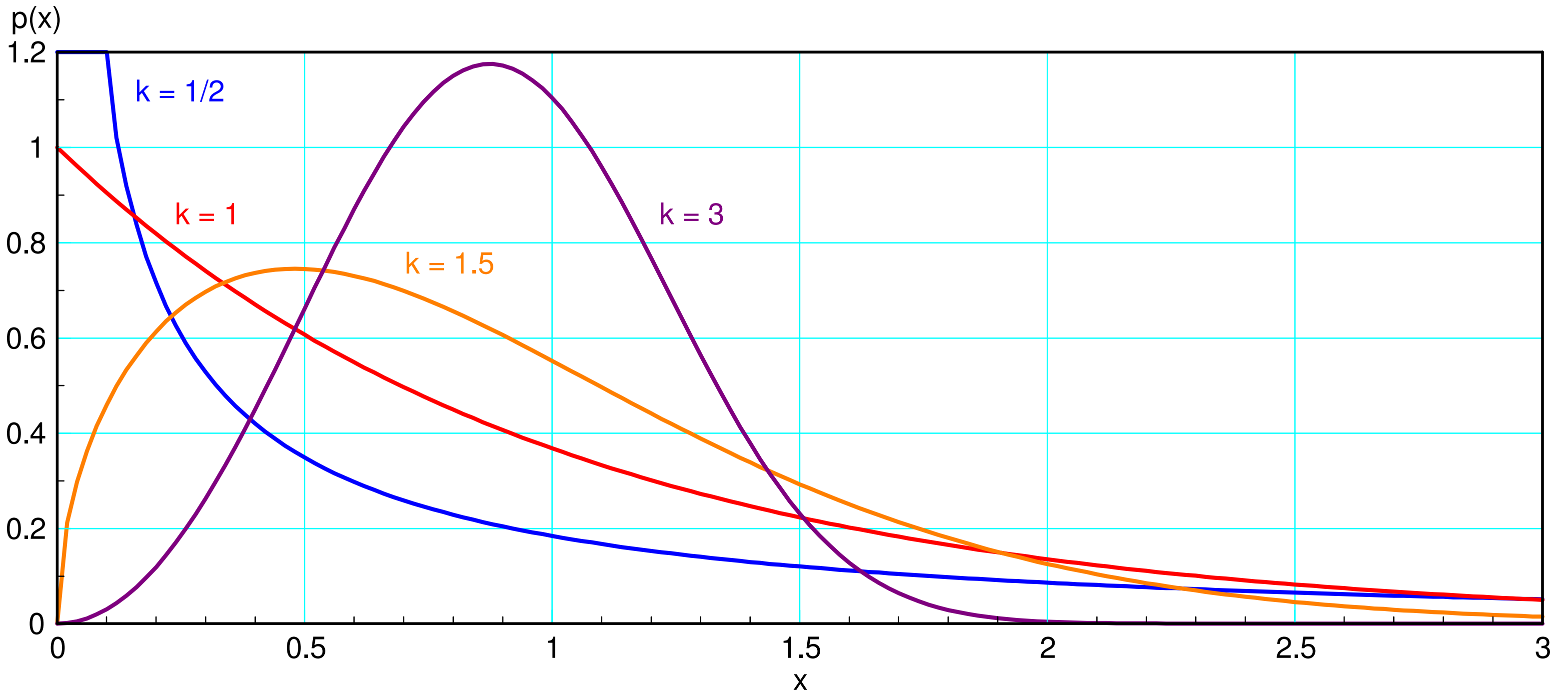
ECE 341: Random Processes

Lecture #16

note: All lecture notes, homework sets, and solutions are posted on www.BisonAcademy.com

Weibull Distribution

- No theoretical underpinnings.
- It's just able to approximate a large number of probability density functions fairly well
- Only uses two parameters: λ and k.



pdf and cdf:

pdf:

$$f(x;\lambda,k) = \frac{k}{\lambda}\left(\frac{x}{\lambda}\right)^{k-1} e^{-(x/\lambda)^k} \cdot u(x)$$

cdf:

$$F_x(\lambda,k) = \left(1 - e^{-(x/\lambda)^k}\right) u(x)$$

To determine λ and k, we'll use the function *fminsearch* in Matlab

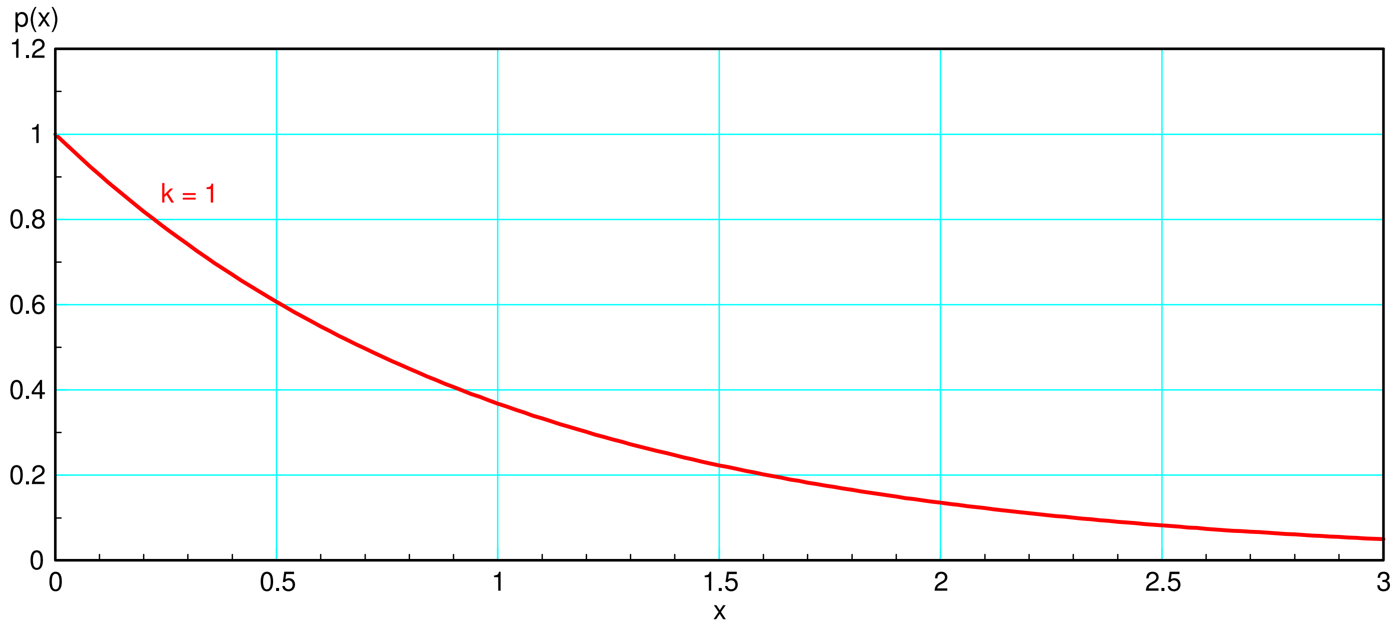
Example 1: Weibull approximation for an exponential pdf

The pdf for an exponential distribution is

$$f(x) = a\, e^{-ax}\, u(x)$$

The Weibull distribution can match this exactly (k = 1, λ = 1/a)

$$f(x) \approx \frac{k}{\lambda}\left(\frac{x}{\lambda}\right)^{k-1} e^{-(x/\lambda)^k}\, u(x)$$



Matching an Gamma distribution:

- Geometric = Time until the next customer
- Gamma = Time until the kth customer (Wikipedia)

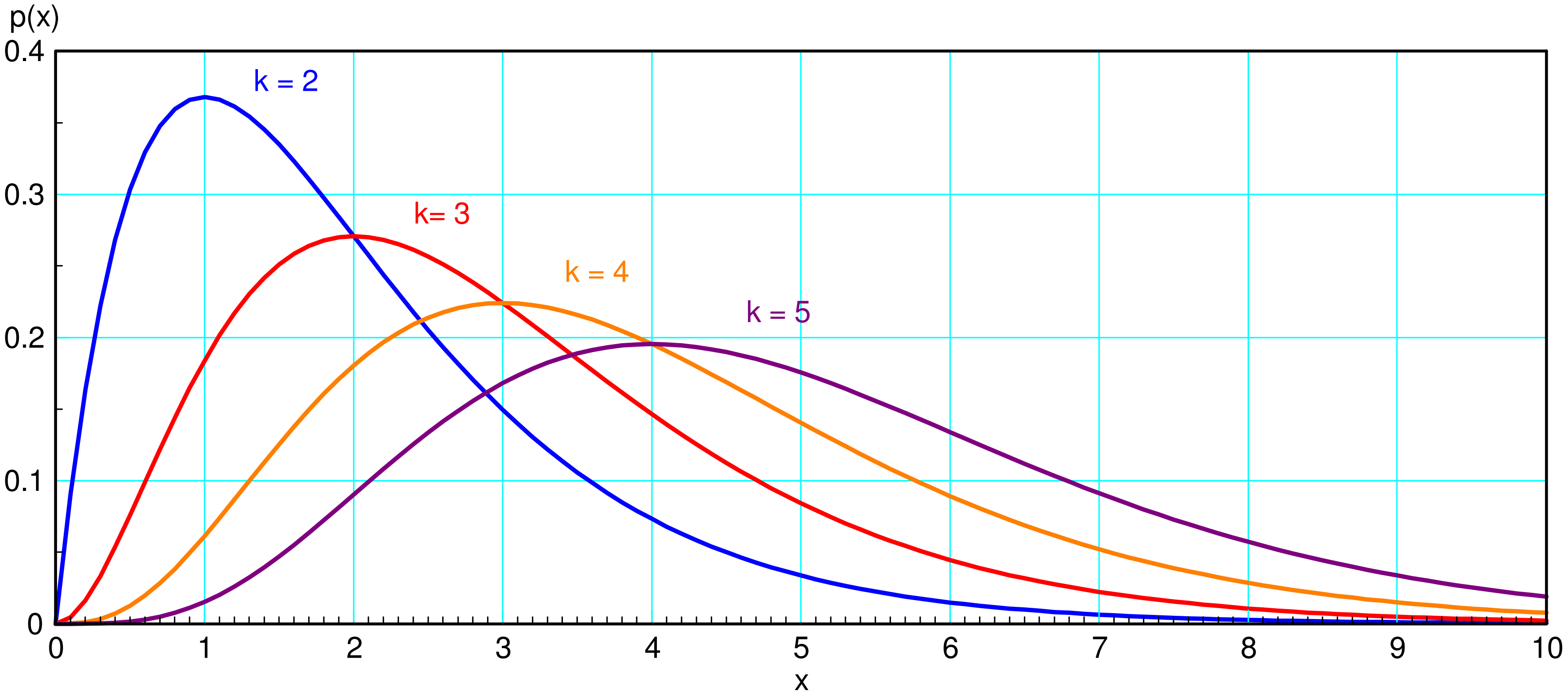
$$f(x) = \left(\frac{1}{(k-1)!\ \theta^k}\right)\ x^{k-1}\ e^{-x/\theta}$$

where

- k is the number of number of customers,
- θ is the average time between customers arriving, and
- x is the time it takes for k customers to arrive

Example:

- Average time between customers arriving is 1 minute
- The pdf for a Gamma distribution is:



Let k = 5 (time until the 5th customer arrives)

$$f(x) = \left(\frac{1}{4!}\right) x^4 \, e^{-x} \qquad \textit{gamma}$$

$$f(x) \approx \frac{k}{\lambda}\left(\frac{x}{\lambda}\right)^{k-1} e^{-(x/\lambda)^k} \qquad \textit{Weibull}$$

Using fminsearch in Matlab, you can optimize the parameters for a Weibull distribution:

fminsearch()

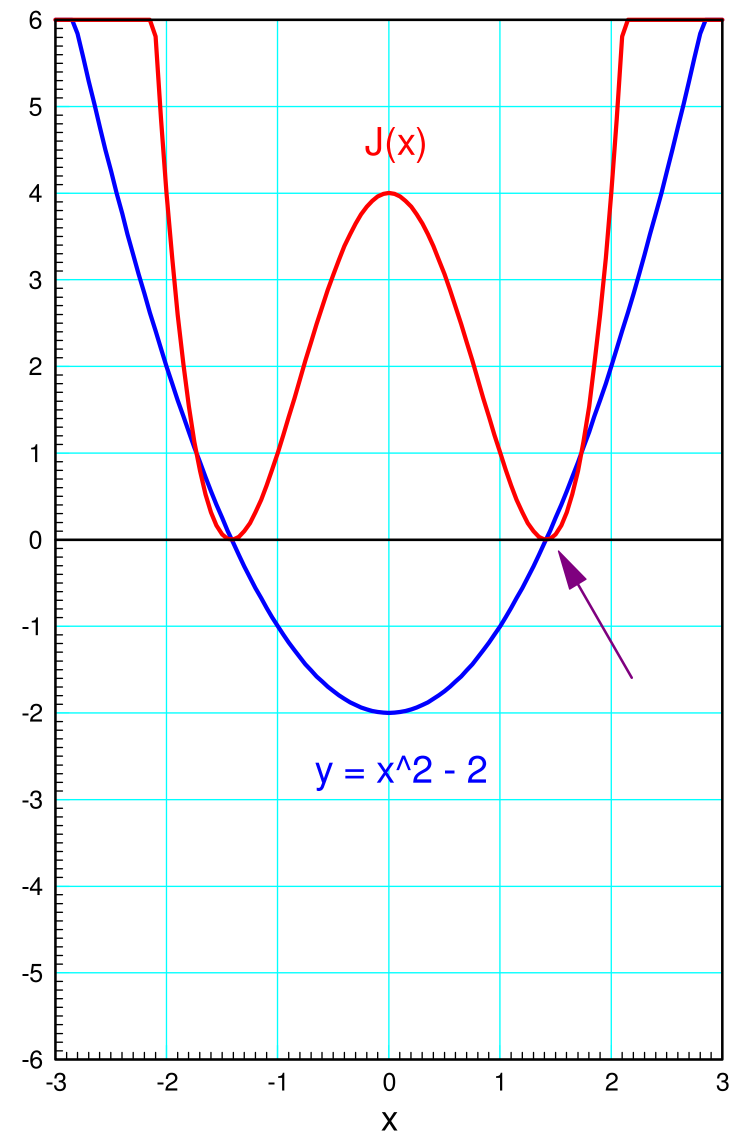
- Really useful Matlab function
- Finds the minimum of a function

Example: Find $\sqrt{2}$

```
function [ J ] = cost( z )
   e = z*z - 2;
   J = e^2;
end
```

Minimize in Matlab

```
>> [a,b] = fminsearch('cost',4)

a =     1.4143
b =     1.5665e-008
```



Example: Shape of a hanging chain

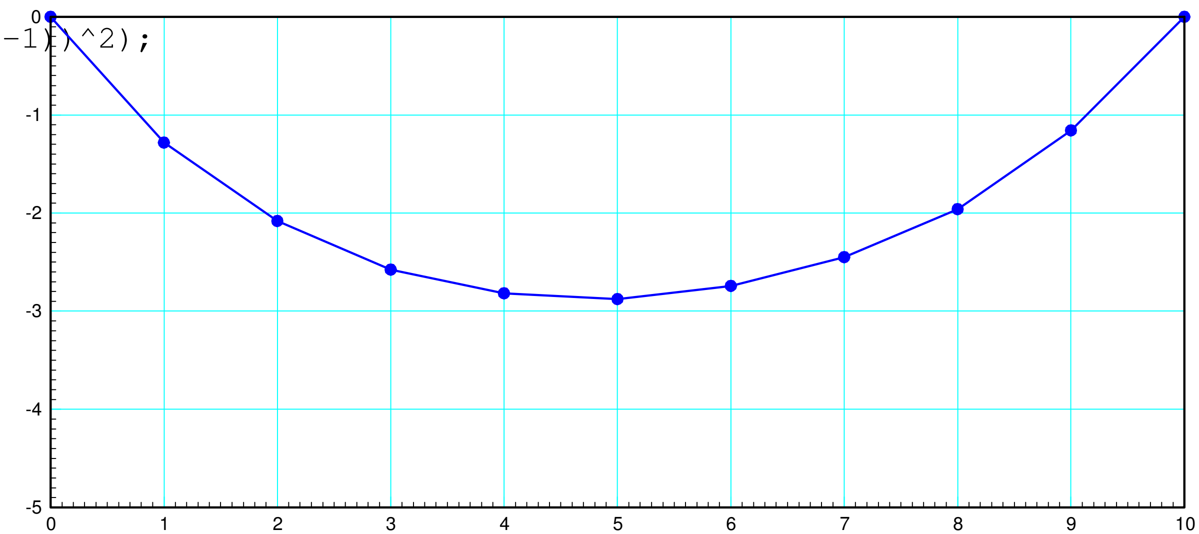
Minimize the potential energy

$$PE = mg(y_1 + y_2 + ... + y_9)$$

Constrain the length to be 12 meters (ish)

$$J = PE + \alpha(12 - L)^2$$

```
function [ J ] = cost_chain( Z )

   Y = [0;Z;0];
   PE = sum(Y);
   L = 0;
   for i=2:11
      L = L + sqrt(1 + (Y(i) - Y(i-1))^2);
   end

   E = (12 - L);
   J = PE + 100*E^2;

   plot([0:10],Y,'.-');
   ylim([-5,1]);
   pause(0.01);

   end


y = i .* (i-10);
[a,b] = fminsearch('cost',0.2*y)
```



Filter Design with fminsearch:

$$|G_d(s)| = \begin{cases} 1 & \omega < 3 \\ 0 & \omega > 3 \end{cases}$$

Step 1: Assume the form of the filter

$$G(s) = \left(\frac{a}{\left(s^2+bs+c\right)\left(s^2+ds+e\right)} \right)$$

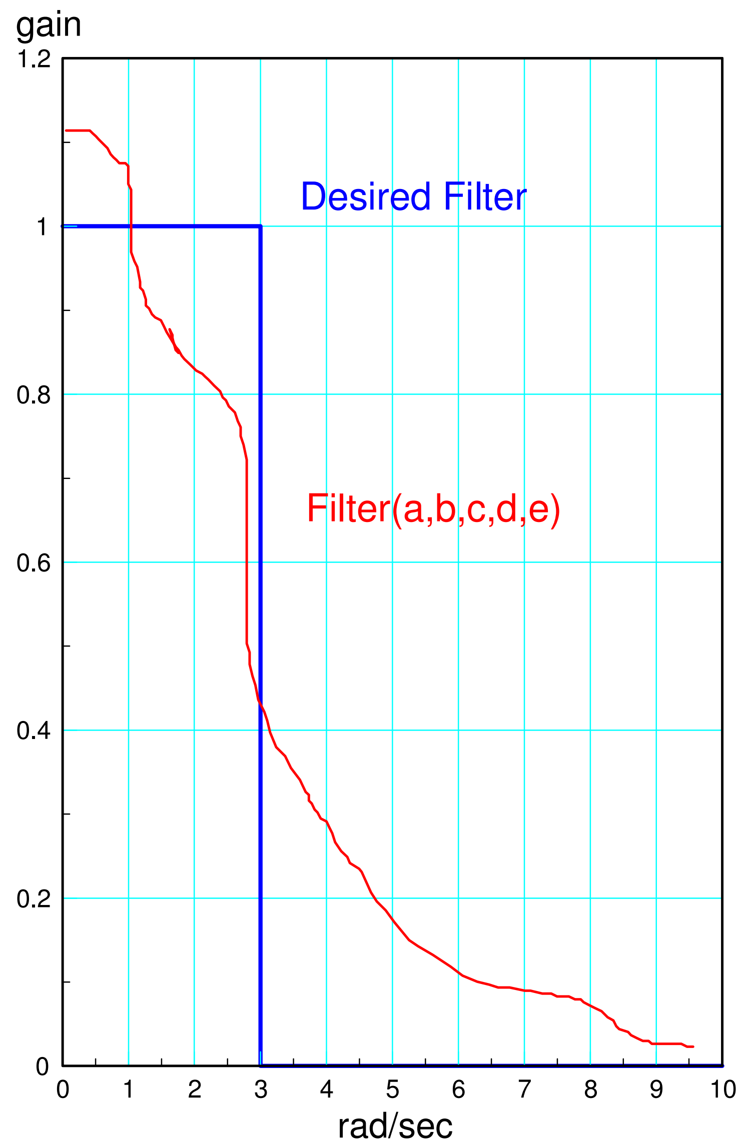
Define the cost (J)

- Minimum is when G(s) = desired filter

$$E(s) = |G(s)| - |G_d(s)|$$

$$J = \sum E^2$$

```
Guess {a, b, c, d, e} to minimize J
```

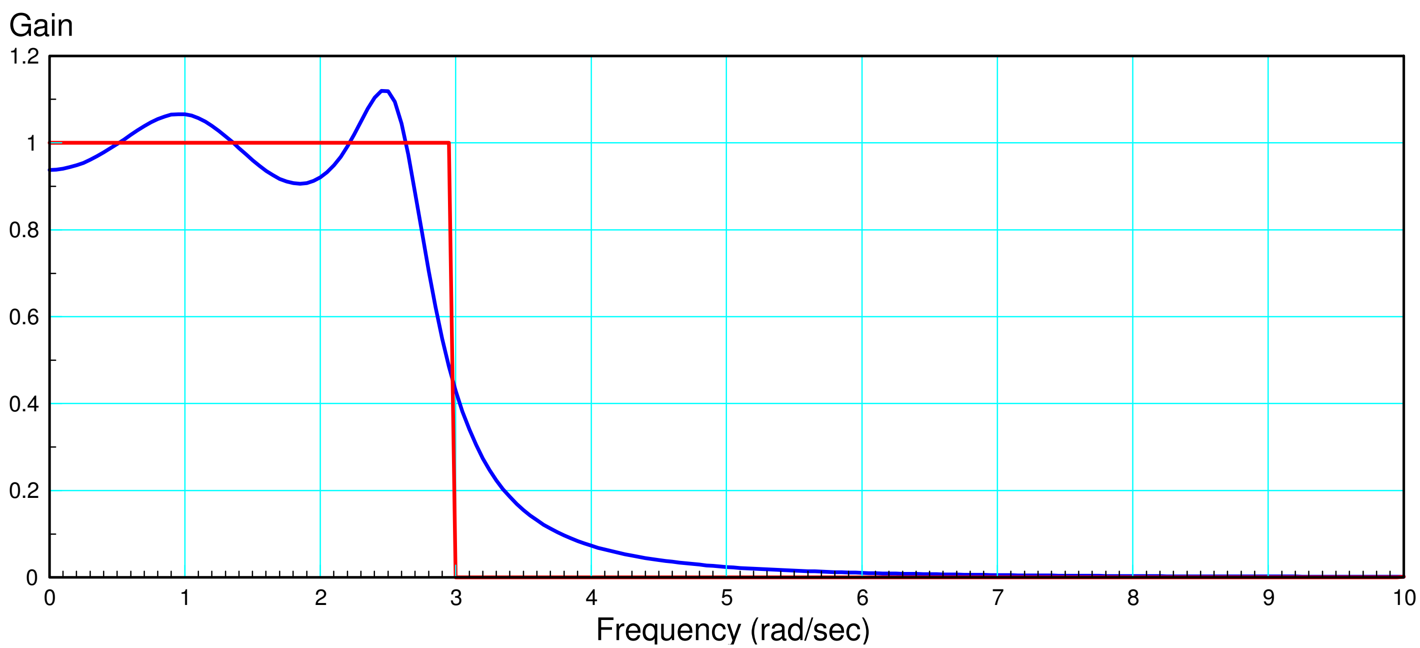


```
function [ J ] = costF( z )

   a = z(1);
   b = z(2);
   c = z(3);
   d = z(4);
   e = z(5);

   w = [0:0.1:10]';
   s = j*w;

   Gideal = 1 * (w < 3);


   G = a ./ ( (s.^2 + b*s + c).*(s.^2 + d*s + e) );

   e = abs(Gideal) - abs(G);

   J = sum(e .^ 2);

   plot(w,abs(Gideal),w,abs(G));
   ylim([0,1.2]);
   pause(0.01);

end
```

Call fminsearch with an initial guess for (a,b,c,d)

```
>> [Z,e] = fminsearch('costF',[1,2,3,4,5])
             a          b          c          d          e
Z =    10.9474    1.6224    1.7317    0.6141    6.7413
e =     0.9575
```

$$G(s) = \left(\frac{10.9474}{\left(s^2+1.6224s+1.7317\right)\left(s^2+0.6141s+6.7413\right)} \right)$$



fminsearch() & Weibull Approximation for a Gamma pdf

First, create a cost function

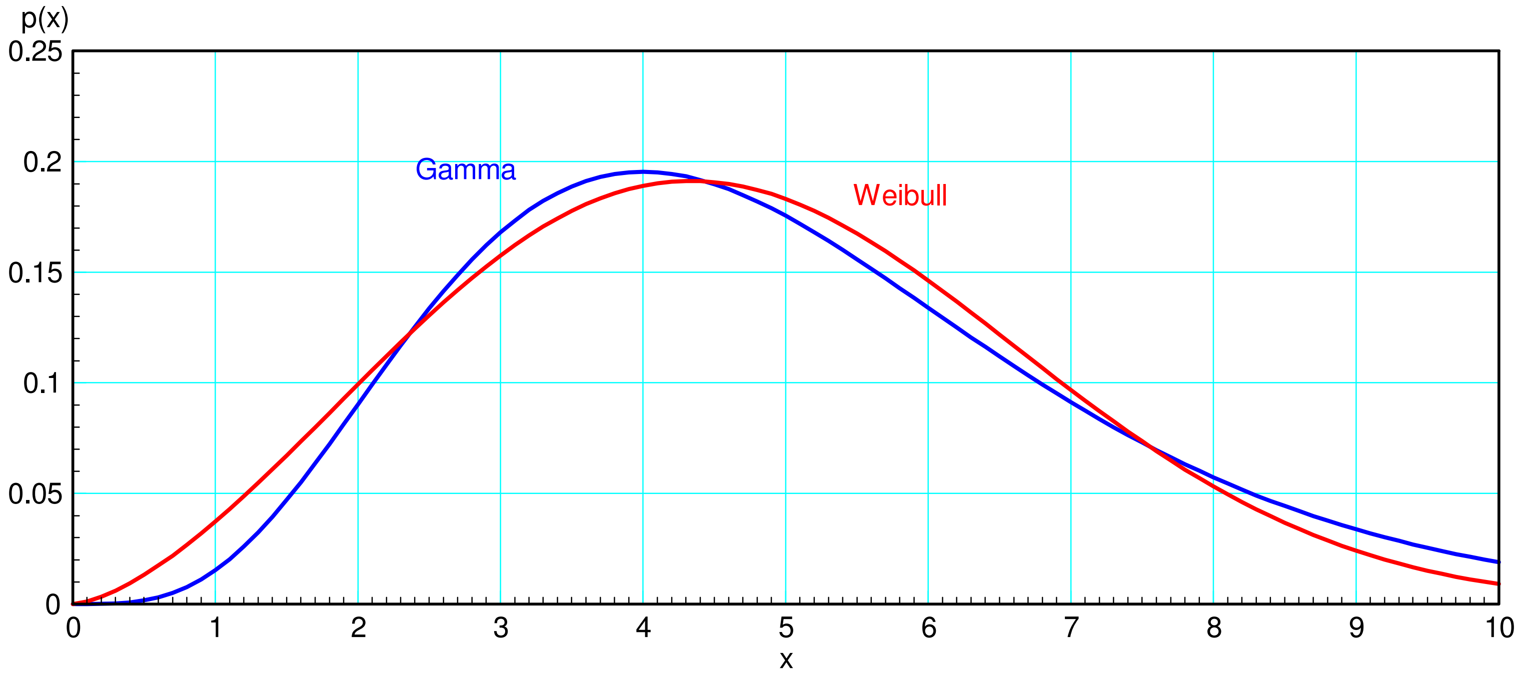
- The desired pdf (Gamma distribution),
- The approximate pdf (Weibull distribution), and
- The sum squared difference in the two

```
function [J] = cost14(z)
L = z(1);
k = z(2);
x = [0:0.1:10]';

% Gamma
G = ( 1 / factorial(4) ) * x.^4 .* exp(-x);

%Weibull
W = (k/L) * (x/L) .^ (k-1) .* exp( -(x/L).^k );

E = G - W;
J = sum(E.^2);
plot(x,G,x,W);
pause(0.01);
end
```

Now minimize the sum squared error in the pdf:

```
[Z,e] = fminsearch('cost',[1,1])

Z =     5.3043    2.5146
e =     0.0110
```

This tells you that a Weibull distribution with $\lambda = 5.3042$, $k = 2.5146$



Weibull Approximation for a Binomial Distribution (Poisson):

As a second example, approximate a binomial distribution with

n = 500

p = 0.01

Approximate this as a Poisson distribution:

$$f(x) = \frac{1}{x!} \cdot \lambda^x e^{-\lambda}$$

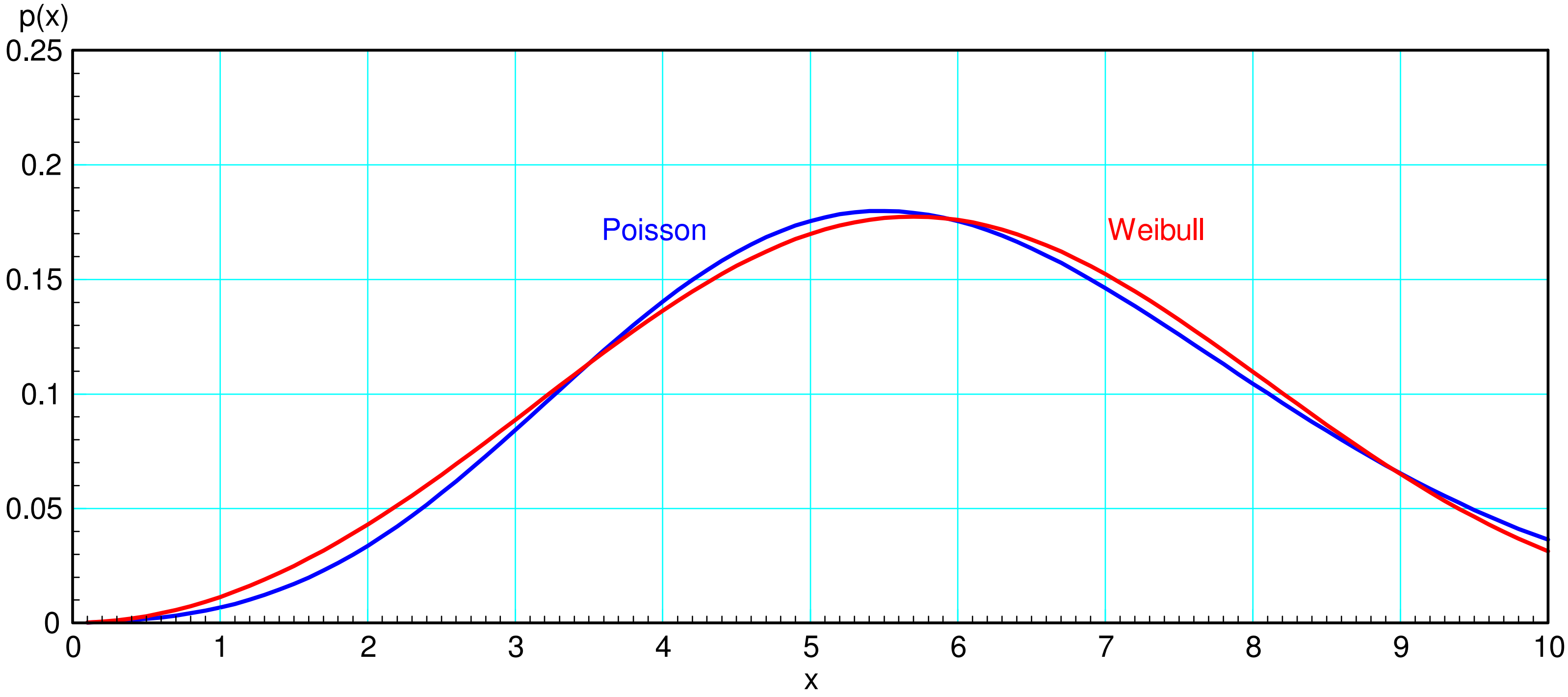
where $\lambda = np = 5$.

Repeating the previous procedure, define

```
function y = cost(z)
% y = cost(z)
% Weibull distribution curve fit

   k = z(1);
   L = z(2);

   x = [0.1:0.1:20]';
   np = 5;

   f = 0.2 * (1 ./ (gamma(x) ) ) .* (np .^ x) * (exp(-np));

   W = (k/L) * ( (x/L) .^ (k-1) ) .* exp( -( (x/L) .^ k ) );

   e = f - W;
   plot(x,f,x,W);
   pause(0.01);

   y = sum(e.^2);

end
```

Calling Routine:

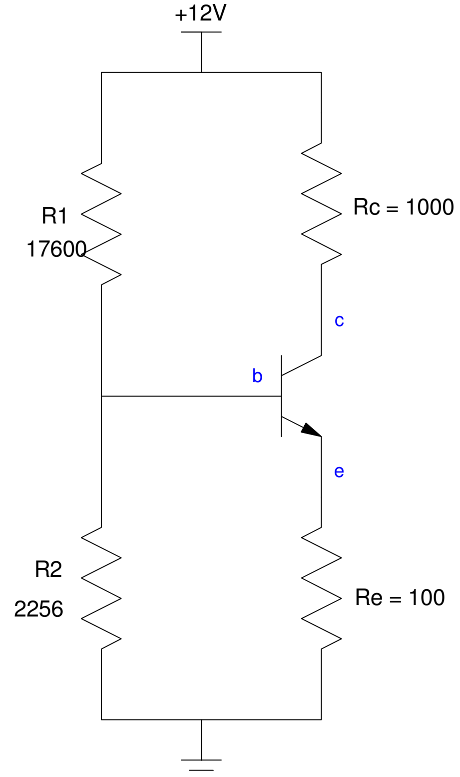
```
[Z,e] = fminsearch('cost',[1,2])

 Z  =     2.9585655    6.5479469

 e  =     0.0000109
```

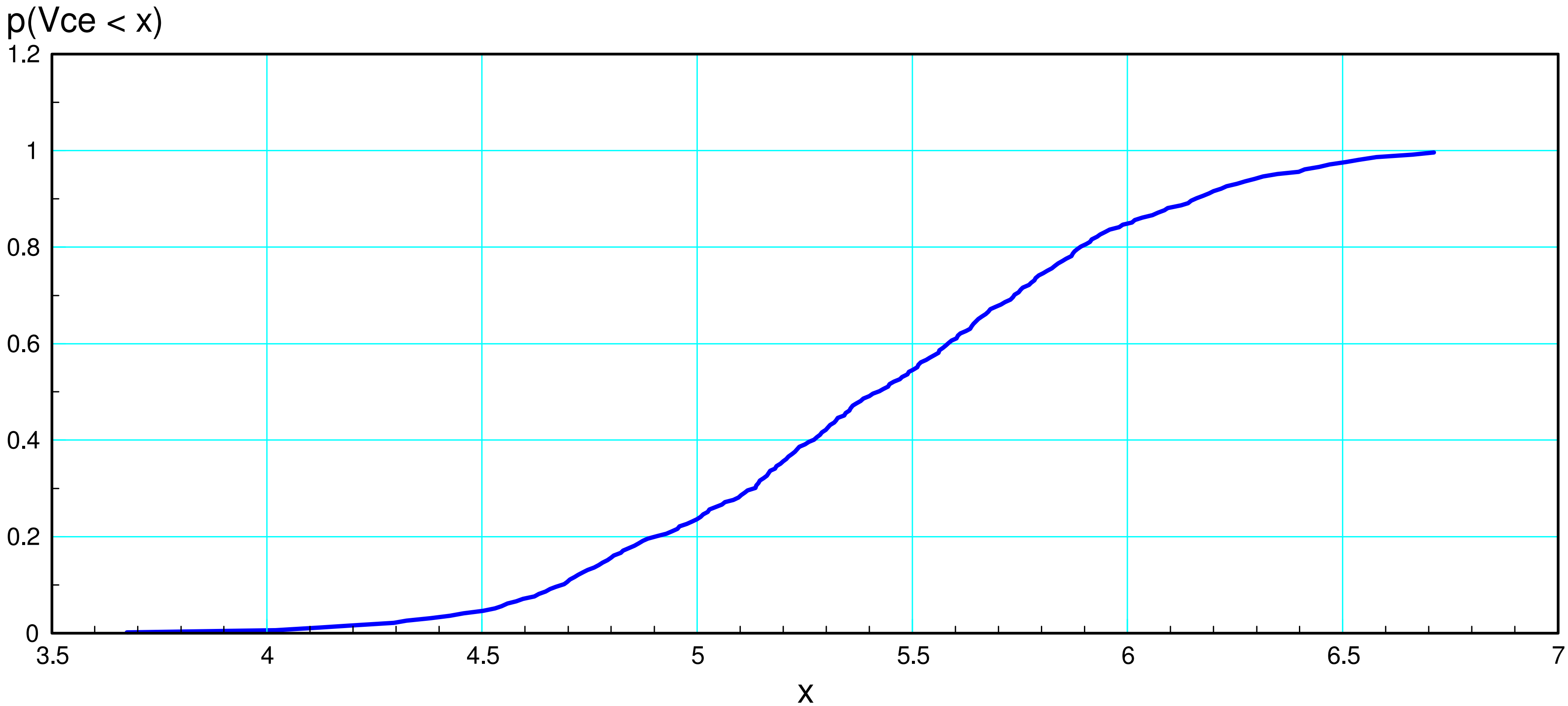


Weibull Approximation for Circuit Voltage

Since the Weibull distribution is so versatile, it can be used when you don't really know what the distribution really is. For example, consider the following circuit where the components have 5% tolerance:

```
DATA = [];

for i=1:1000
 R1 = 17600 * (1 + (rand()*2-1)*0.05);
 R2 = 2256 * (1 + (rand()*2-1)*0.05);
 Rc = 1000 * (1 + (rand()*2-1)*0.05);
 Re = 100 * (1 + (rand()*2-1)*0.05);
 Beta = 200 + 100*(rand()*2-1);
 Vb = 12*(R2 / (R1+R2));
 Rb = 1/(1/R1 + 1/R2);
 Ib = (Vb-0.7) / (Rb + (1+Beta)*Re);
 Ic = Beta*Ib;
 Vce = 12 - Rc*Ic - Re*(Ic+Ib);
 DATA = [DATA; Vce];
end
DATA = sort(DATA);
```





cdf for the voltage, Vce

Determine a Weibull distribution to approximate this data.

$$F_x(\lambda, k) = \left(1 - e^{-(x/\lambda)^k}\right) u(x)$$

```
function y = cost21(z)
% y = cost(z)
% Weibull distribution curve fit

   k = z(1);
   L = z(2);
   X0 = z(3);

% data to curve fit: Vce and p
DATA = [
    3.6745    0.0010
    4.1150    0.0110
    ...
    6.5369    0.9810
    6.6619    0.9910
    ];

   Vce = DATA(:,1);
   p = [1:length(DATA)]' / length(DATA);

   x = Vce - X0;
```

```
    x = max(0,x);

% p(Vce) = target

    W = 1 - exp( -( (x/L) .^ k ) );

    e = p - W;

    plot(Vce,p,Vce,W);
    pause(0.01);

    y = sum(e.^2);

end
```

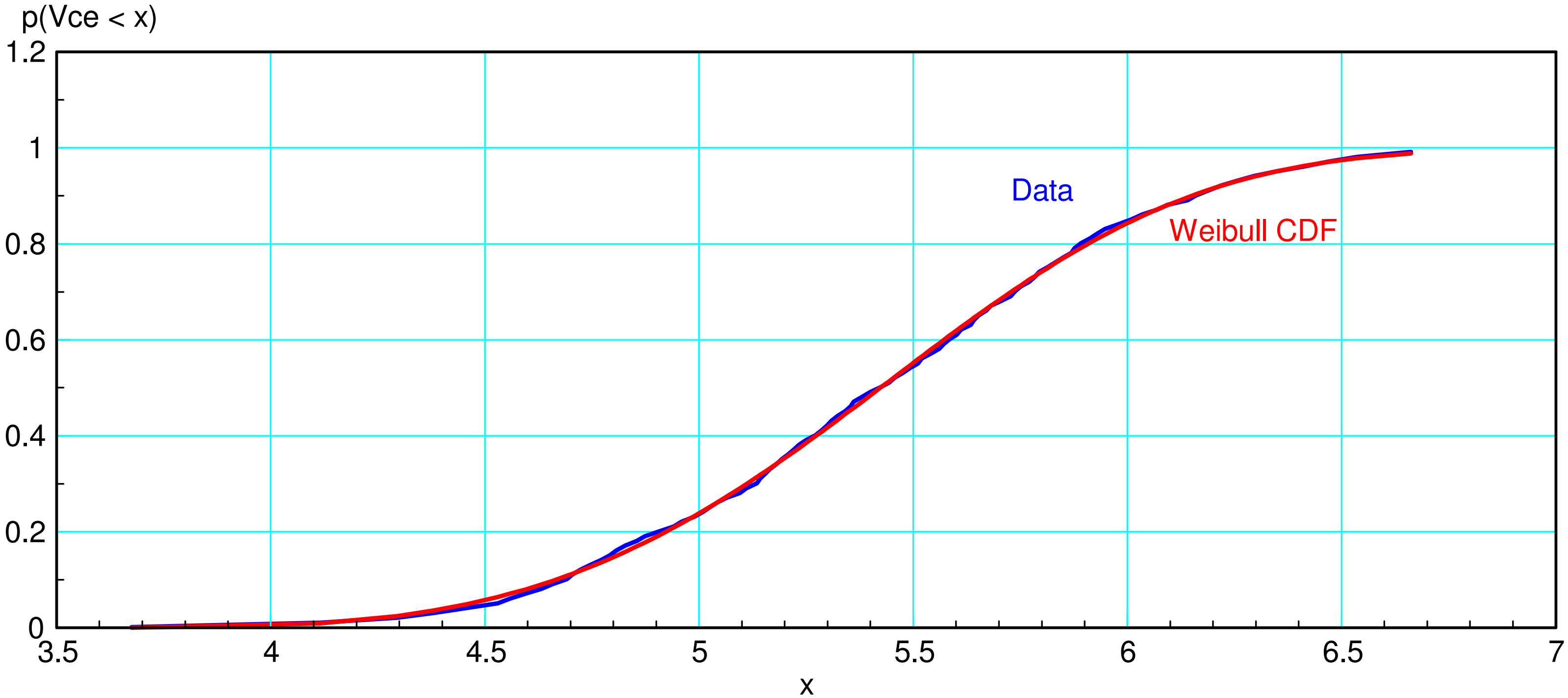
Now optimize with fminsearch

```
[Z,e] = fminsearch('cost',[1,2,3])

Z =    3.7155    2.1034    3.5173

e =    0.0040
```

which tells you that the pdf for Vce is approximately

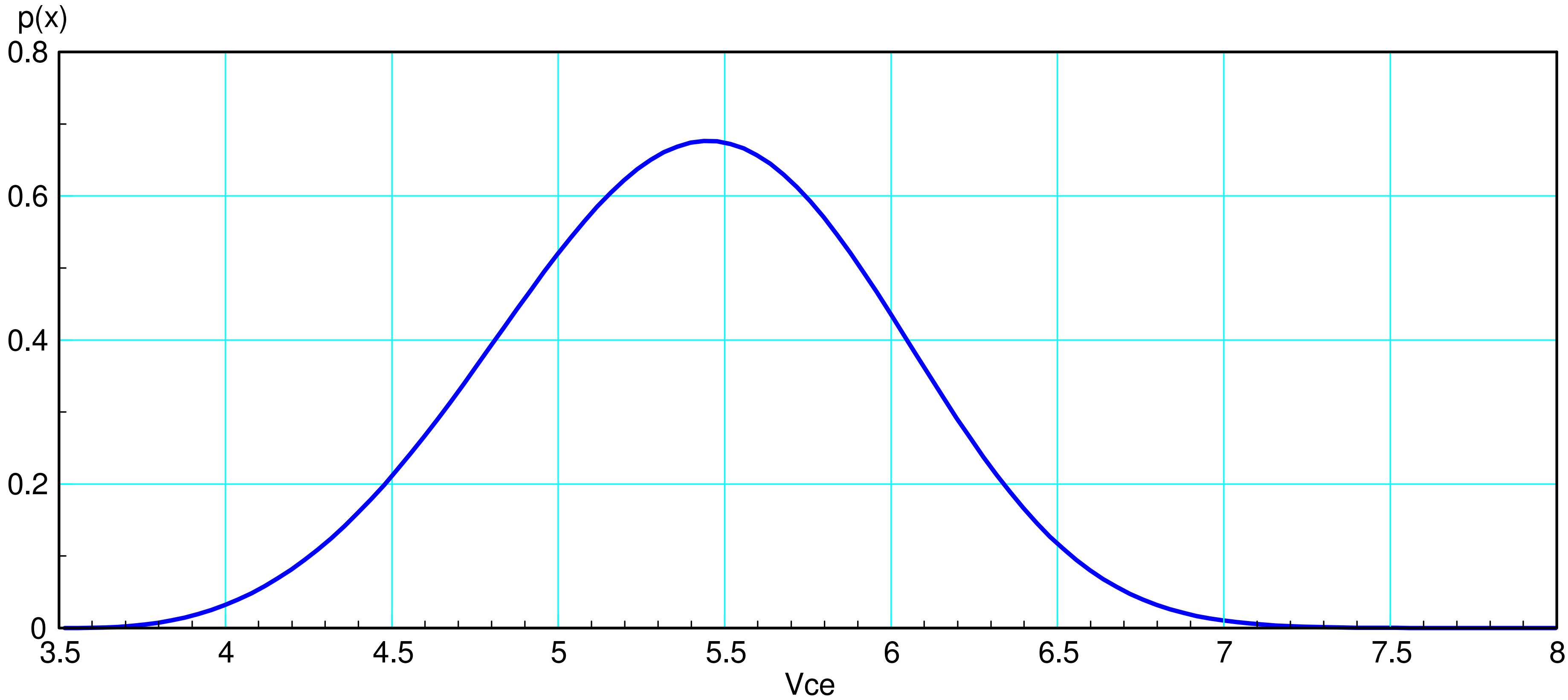
$$F_x(\lambda, k) = \left(1 - e^{-((x-x_0)/\lambda)^k}\right) u(x - x_0)$$

$$k = 3.7155, \qquad \lambda = 2.1034, \qquad x_0 = 3.5173V$$



cdf for Vce and its Weibull approximation which then tells you the pdf is

$$f(x;\lambda,k) = \frac{k}{\lambda}\left(\frac{x-x_0}{\lambda}\right)^{k-1} e^{-((x-x_0)/\lambda)^k} \cdot u(x-x_0)$$



Summary

- fminsearch() is a really useful Matlab function
- Weibull disributions can fit almost any pdf fairly well