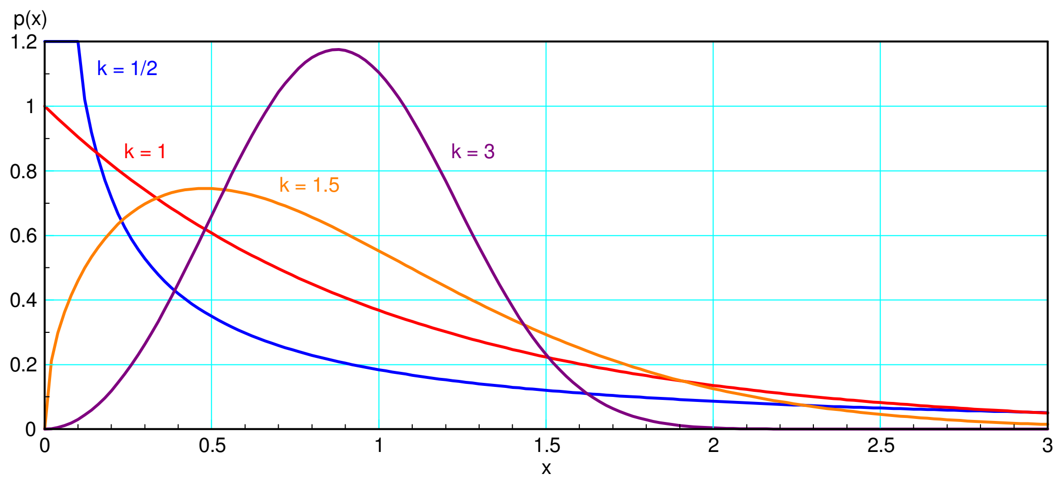
Weibull Distribution

The Weibull distribution has no theoretical underpinnings. It is just a function which is able to approximate a large number of probability density functions fairly well - using only two degrees of freedom: λ and k. For example, the pdf of a Weibull distribution looks like the following:



pdf of a Weibull distribution with λ=1

The pdf and cdf of a Weibull distribution are:

pdf:

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

cdf:

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

With just two parameters, it can provide a close approximation for other probability distributions. It can also approximate other function where you don't know the actual distribution.

To determine λ and k, we'll use the function *fminsearch* in Matlab (*leastsq* in SciLab).

MATLAB: fminsearch()

A very useful function is fminsearch. This finds the minimum of a function.

For example, find the square root of two. Create a function which is a minimum at the square root of two:

```
Cost.m
function [y] = cost(z)
e = z*z - 2;
y = e*e;
```

This function computes the error between your guess, z, and the correct answer. To make the result a minimum when the error is zero, the function returns the square of the error.

Now from MATLAB type

```
fminsearch('cost',4)
```

This routine will iterate until if finds the minimum: 1.414.

```
-->cost2(3)
```

Example 1: Matching an exponential distribution:

The pdf for an exponential distribution is

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

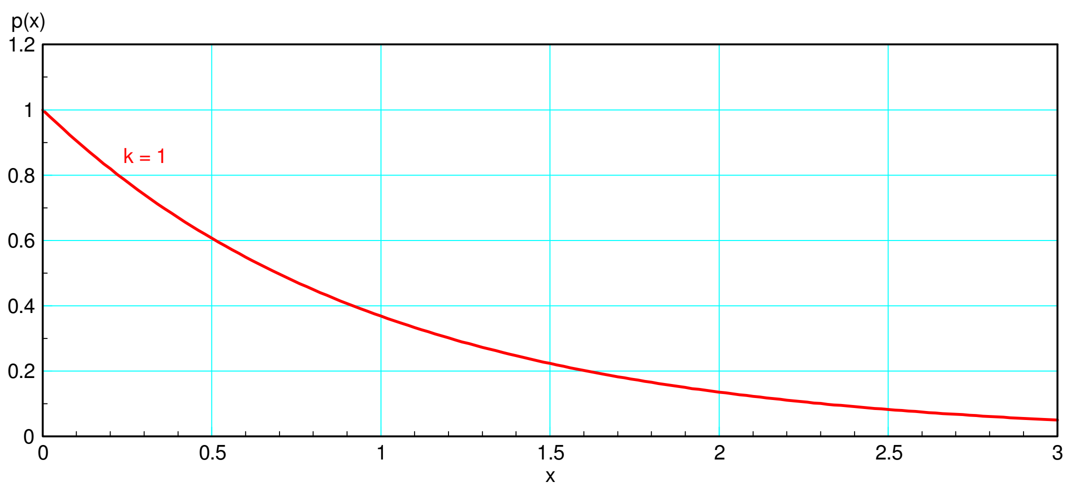
The Weibull distribution can match this exactly

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

Matching terms:

$$k = 1$$

$$\lambda = \frac{1}{a}$$



k=1 results in a Weibull approximation for an exponential distribution with a=1.

Matching an Gamma distribution:

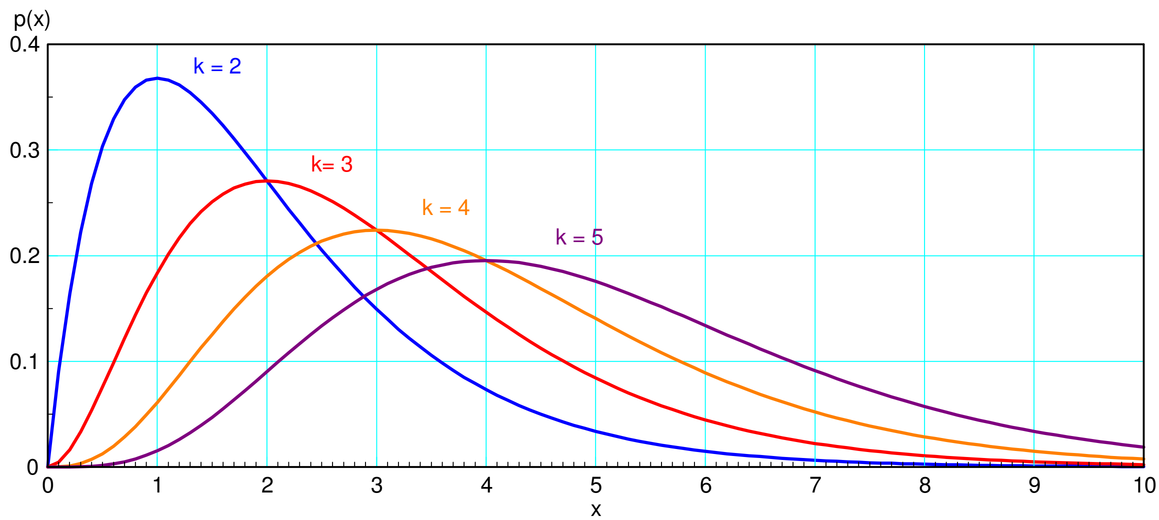
The geometric distribution is the time until the next customer arrives at a restaurant. A Gamma distribution is the time until the kth customer arrives at a restaurant. The pdf for a Gamma distribution (Wikipedia) is:

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

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

For example, if the average time between customers arriving is 1 minute, the pdf for a Gamma distribution is:



pdf for a Gamma distribution (time until k customers arrive) for k = 2, 3, 4, 5. Average arrival time = 1.

If you pick k = 5, then you can match this to a Weibull distribution as

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

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

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

First, create a cost function which computes

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

```
function [J] = cost(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:

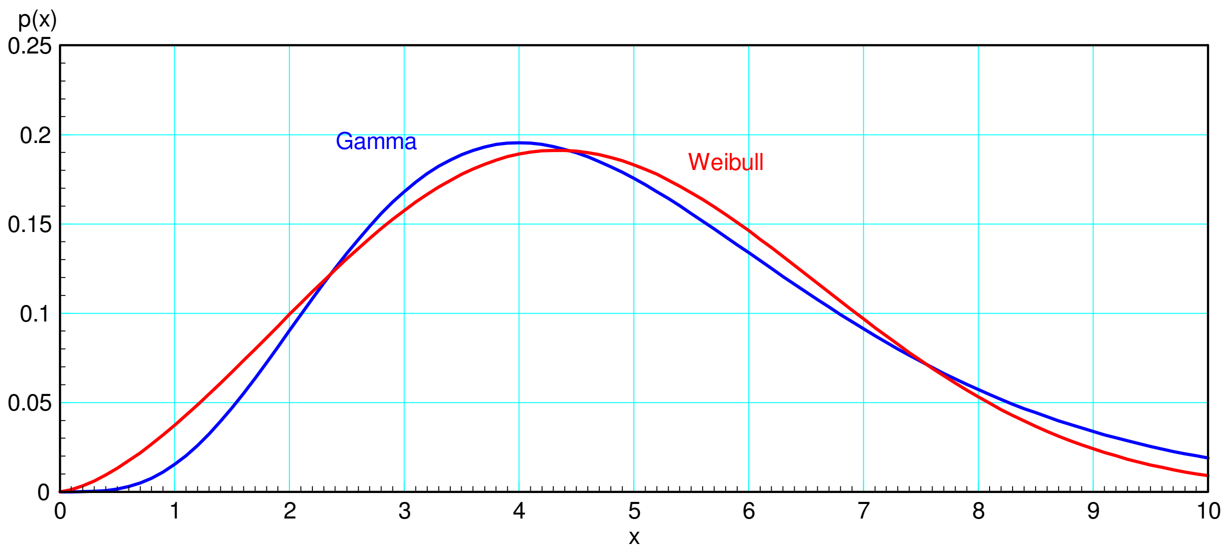
```
[Z,e] = fminsearch('cost',[1,1])

Z =     5.3043    2.5146
e =     0.0110
```

This tells you that a Weibull distribution with

$$\lambda = 5.3042$$

$$k = 2.5146$$

is the closest you can get. The two pdf's look like the following:



Gamma distribution (blue) and a Weibull approximation to it (red)

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 $\lambda = np = 5$.

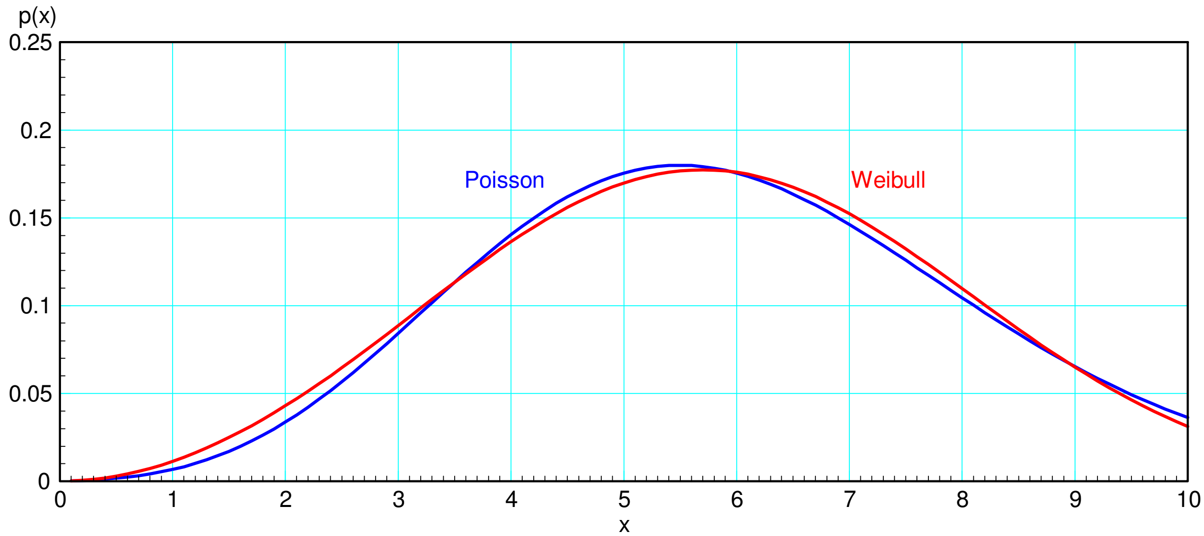
Repeating the previous procedure, define

```
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])

            k           lambda
Z  =    2.9585655    6.5479469

e  =    0.0000109
```

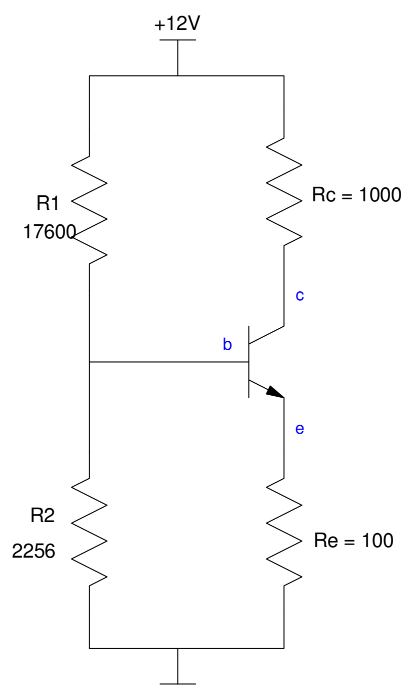
The resulting pdf's are almost the same:



Poisson distribution (blue) and Weibull approximation (red)

Weibull Approximation for Circuit Voltage

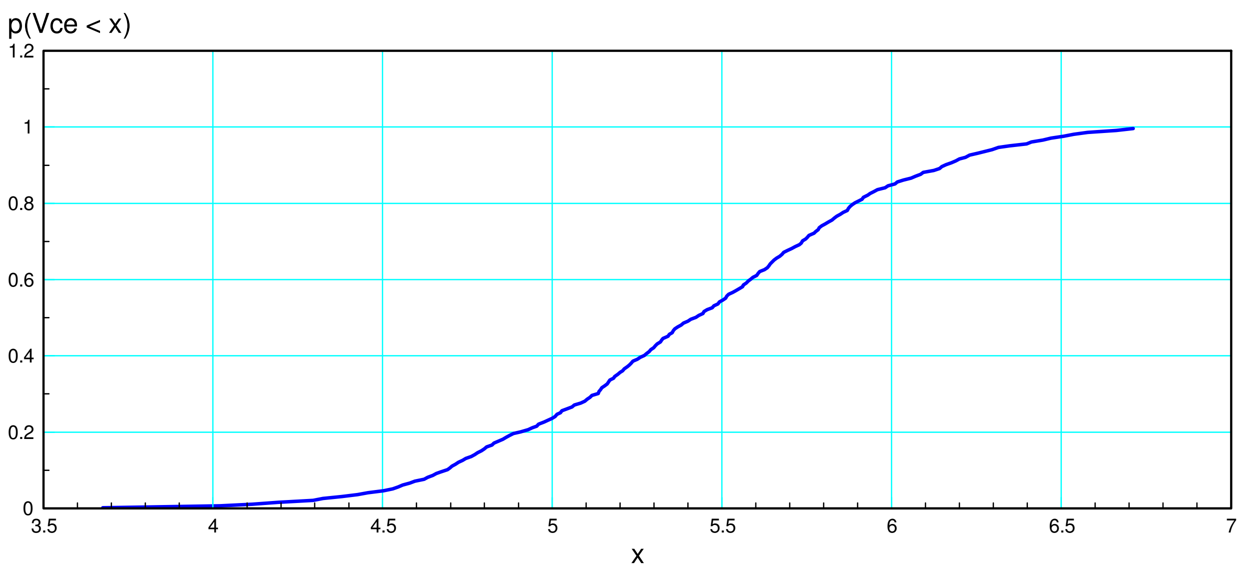
Since the Weibull distribution is so versatile, it can be used when you don't really know what the distribution really is. For example, consider the following circuit where the components have 5% tolerance:



Circuit with 5% tolerance components (analyzed in ECE 321)

Using a Monte Carlo simulation, you can determine the CDF for the voltage Vce

```
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);

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



cdf for the voltage, Vce

Determine a Weibull distribution to approximate this data.

Recall the cdf of a Weibull distribution is

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

Write a function to compare the Weibull CDF to the actual CDF

```
function y = cost(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 = DATA(:,2);

   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
```

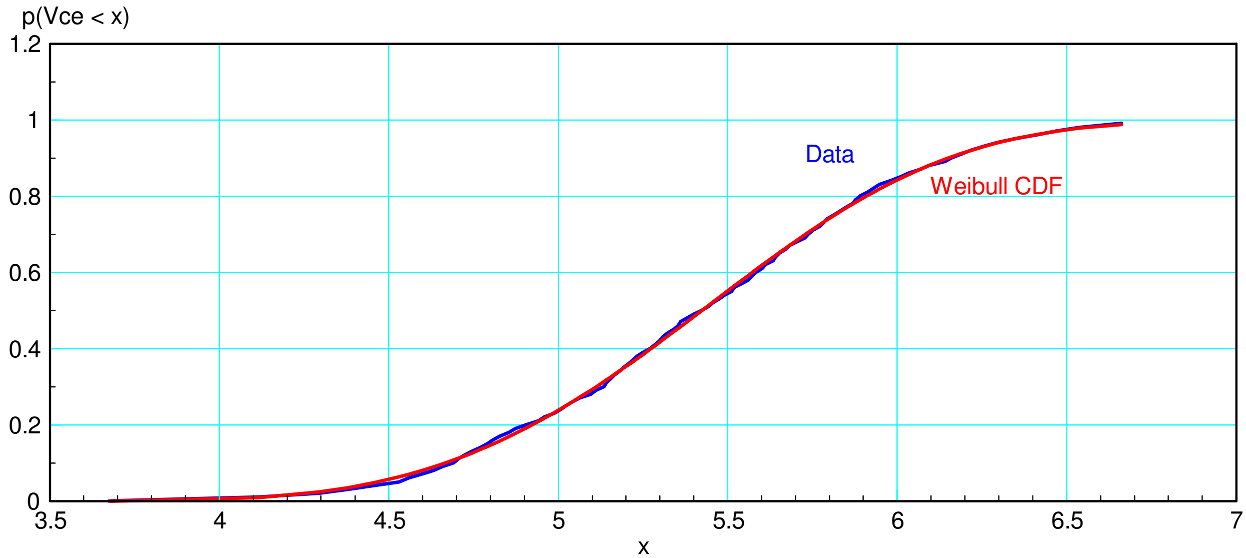
which tells you that the pdf for Vce is approximately

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

$$k = 3.7155$$

$$\lambda = 2.1034$$

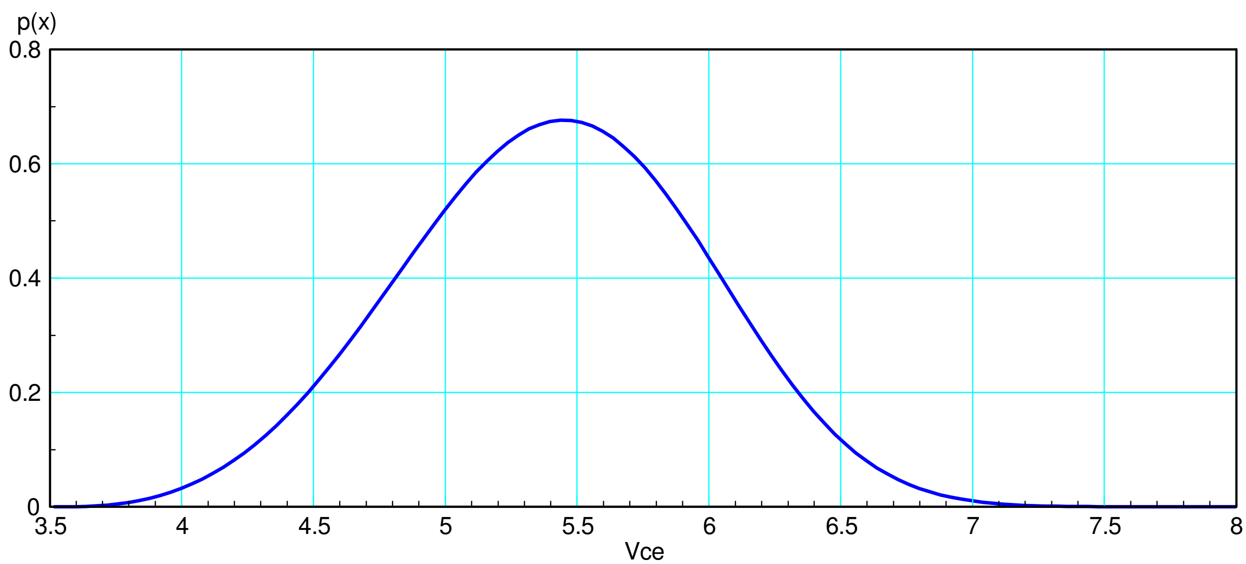
$$x_0 = 3.5173V$$



cdf for Vce and its Weibull approximation

which then tells you the pdf is

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



pdf for Vce (Weibull approximation)