```
clear; clc;

format long;

numElem = 1e3;
lmbd = 3e-6;
Pmin = 0.98;
t = 1000;

POrgnSstm = exp(-lmbd*numElem*t);
fprintf('Initial system reliability is %0.2g.\n', POrgnSstm);

% Decrease lambda
lmbdDcrs = log(1/Pmin)/t/numElem;
lmbdRtn = lmbdDcrs/lmbd;
fprintf('To ensure the required level of reliability the labmda must
 be decreased in %g\n times.', lmbdRtn);

% Make it easy. Decrease number of elements
numElemDcrs = log(1/Pmin)/t/lmbd;
dltElmnts = numElem - numElemDcrs;
fprintf('To ensure the required level of reliability the number
 elements must be decreased by %g.\n', dltElmnts);

% Decrease using time
tDcrs = log(1/Pmin)/numElem/lmbd;
dltT = t - tDcrs;
fprintf('To ensure the required level of reliability the time must be
 decreased by %g.\n', dltT);

% Test 0.1lambda 0.1numElem 0.1t
PSstmTest = exp(-0.1^3*lmbd*numElem*t);
fprintf('If all params multiply by 0.1 then the reliability is %g.\n',
 PSstmTest);

% Test 0.5lambda 0.5numElem 0.5t
PSstmTest = exp(-0.5^3*lmbd*numElem*t);
fprintf('If all params multiply by 0.5 then the reliability is %g.\n',
 PSstmTest);

% Find k. K is coeif. change all in k times
k = (log(1/Pmin)/(numElem*lmbd*t))^(1/3);
fprintf('To ensure the required level of reliability all params must
 be multiply by %g.\n', k);

% Test k*lambda k*numElem k*t
PSstmTest = exp(-k*lmbd*floor(k*numElem)*floor(t*k));
fprintf('To ensure the required level of reliability %g, lambda must
 be %g, time must be %g, number elements must be %g.\n', PSstmTest,
 k*lmbd, floor(k*t), floor(k*numElem));

% Hot res all sheme
```

```
syms m;
eqn = Pmin == 1 - (1 - exp(-lmbd*numElem*t))^(m+1);
mHRAllSheme = double(solve(eqn, m));
fprintf('To ensure the required level of reliability the can use
 hot reserve all system with the coeffient multiplicity eqal %g.\n',
 ceil(mHRAllSheme));

% Hot res every elements
eqn = 1 - Pmin^(1/numElem) ==  (1 - exp(-lmbd*t))^(m+1);
mHREveryElem = double(solve(eqn, m));
fprintf('To ensure the required level of reliability the can use hot
 reserve all elemets with the coeffient multiplicity eqal %g.\n',
 mHREveryElem);

% Test hot res every elements in 1 times, m = 1
PSstmTest = (1 - (1 - exp(-lmbd*t))^2)^numElem;
fprintf('Reliability with HR every elemets then m = 1 is %g.\n',
 PSstmTest);

% Hot res part of elements in 1 times, m = 1
eqn = Pmin == exp(-lmbd*t*(numElem - m))*(1 - (1 - exp(-lmbd*t))^2)^m;
numElemPartHR1 = double(solve(eqn, m));
fprintf('To ensure the required level of reliability with HREE then m
 = 1 the we must reserve %g elements.\n', ceil(numElemPartHR1));

eqn = Pmin == exp(-lmbd*t*(500 - m))*(1 - (1 - exp(-lmbd*t))^2)^m;
numElemPartHR1 = double(solve(eqn, m));
fprintf('To ensure the required level of reliability with HREE then m
 = 1 and number elemetns is 500 the we must reserve %g elements.\n',
 ceil(numElemPartHR1));

eqn = Pmin == exp(-0.1*lmbd*t*(numElem - m))*(1 - (1 -
 exp(-0.1*lmbd*t))^2)^m;
numElemPartHR1 = double(solve(eqn, m));
fprintf('To ensure the required level of reliability with HREE then
 m = 1 and lambda reduce in 0.1 the we must reserve %g elements.\n',
 ceil(numElemPartHR1));

eqn = Pmin == exp(-0.1*0.9*lmbd*t*(numElem - m))*(1 - (1 -
 exp(-0.1*0.9*lmbd*t))^2)^m;
numElemPartHR1 = double(solve(eqn, m));
fprintf('To ensure the required level of reliability with HREE then m
 = 1 and lambda reduce in 0.1 and time to 0.9time the we must reserve
 %g elements.\n', ceil(numElemPartHR1));

% Hot res part of elements in 2 times, m = 2
eqn = Pmin == exp(-lmbd*t*(numElem - m))*(1 - (1 - exp(-lmbd*t))^3)^m;
numElemPartHR2 = double(solve(eqn, m));
fprintf('To ensure the required level of reliability with HREE then m
 = 2 the we must reserve %g elements.\n', ceil(numElemPartHR2));

% Hot res part of elements in 1 times, m = 1, numElemPart = 998
% PSstmTest = exp(-lmbd*t*(numElem - ceil(numElemPartHR)))*(1 - (1 -
 exp(-lmbd*t))^2)^ceil(numElemPartHR);
```

```
% fprintf('%g.\n', PSstmTest);

% Cold res part of elements in 1 times, m = 1
eqn = Pmin == exp(-lmbd*t*(numElem - m))*(exp(-lmbd*t)*(1 +
 lmbd*t))^m;
numElemPartCR1 = double(solve(eqn, m));
fprintf('To ensure the required level of reliability with CREE then m
 = 1 the we must reserve %g elements.\n', numElemPartCR1);

eqn = Pmin == exp(-lmbd*t*(500 - m))*(exp(-lmbd*t)*(1 + lmbd*t))^m;
numElemPartCR1 = double(solve(eqn, m));
fprintf('To ensure the required level of reliability with CREE then m
 = 1 and number elemetns is 500 the we must reserve %g elements.\n',
 numElemPartCR1);

eqn = Pmin == exp(-0.1*lmbd*t*(numElem - m))*(exp(-0.1*lmbd*t)*(1 +
 0.1*lmbd*t))^m;
numElemPartCR1 = double(solve(eqn, m));
fprintf('To ensure the required level of reliability with CREE then
 m = 1 and lambda reduce in 0.1 the we must reserve %g elements.\n',
 numElemPartCR1);

eqn = Pmin == exp(-0.1*0.9*lmbd*t*(numElem -
 m))*(exp(-0.1*0.9*lmbd*t)*(1 + 0.1*0.9*lmbd*t))^m;
numElemPartCR1 = double(solve(eqn, m));
fprintf('To ensure the required level of reliability with CREE then m
 = 1 and lambda reduce in 0.1 and time to 0.9time the we must reserve
 %g elements.\n', numElemPartCR1);

% Cold res part of elements in 2 times, m = 2
eqn = Pmin == exp(-lmbd*t*(numElem - m))*(exp(-lmbd*t)*(1 + lmbd*t +
 (lmbd*t)^2/2))^m;
numElemPartCR2 = double(solve(eqn, m));
fprintf('To ensure the required level of reliability with CREE then m
 = 2 the we must reserve %g elements.\n', numElemPartCR2);
```

```
Initial system reliability is 0.05.
To ensure the required level of reliability the labmda must be
 decreased in 0.00673424
 times.To ensure the required level of reliability the number elements
 must be decreased by 993.266.
To ensure the required level of reliability the time must be decreased
 by 993.266.
If all params multiply by 0.1 then the reliability is 0.997004.
If all params multiply by 0.5 then the reliability is 0.687289.
To ensure the required level of reliability all params must be
 multiply by 0.188841.
To ensure the required level of reliability 0.980176, lambda must be
 5.66523e-07, time must be 188, number elements must be 188.
To ensure the required level of reliability the can use hot reserve
 all system with the coeffient multiplicity eqal 76.
To ensure the required level of reliability the can use hot reserve
 all elemets with the coeffient multiplicity eqal 0.860328.
Reliability with HR every elemets then m = 1 is 0.991067.
```

```
To ensure the required level of reliability with HREE then m = 1 the
 we must reserve 997 elements.
To ensure the required level of reliability with HREE then m = 1 and
 number elemetns is 500 the we must reserve 495 elements.
To ensure the required level of reliability with HREE then m = 1 and
 lambda reduce in 0.1 the we must reserve 933 elements.
To ensure the required level of reliability with HREE then m = 1 and
 lambda reduce in 0.1 and time to 0.9time the we must reserve 926
 elements.
To ensure the required level of reliability with HREE then m = 2 the
 we must reserve 994 elements.
To ensure the required level of reliability with CREE then m = 1 the
 we must reserve 994.755 elements.
To ensure the required level of reliability with CREE then m = 1 and
 number elemetns is 500 the we must reserve 494.005 elements.
To ensure the required level of reliability with CREE then m = 1 and
 lambda reduce in 0.1 the we must reserve 932.798 elements.
To ensure the required level of reliability with CREE then m = 1 and
 lambda reduce in 0.1 and time to 0.9time the we must reserve 925.3
 elements.
To ensure the required level of reliability with CREE then m = 2 the
 we must reserve 993.267 elements.
```

Published with MATLAB® R2016b