```
% opts.format='pdf'; opts.outputDir='.'; publish('tcadamage.m',opts);
clear all
close all


% load fig 2 data from
%   Freeman, Michael L., et al. "The effect of pH on cell lethality
 induced by hyperthermic treatment." Cancer 45.9 (1980): 2291-2300.
fig2data = readtable('freemanFig2.csv')

base10min = fig2data(strcmp(fig2data.group,'10minpH7.5') ,:);
acid10min = fig2data(strcmp(fig2data.group,'10minpH6.65'),:);
base15min = fig2data(strcmp(fig2data.group,'15minpH7.5') ,:);
acid15min = fig2data(strcmp(fig2data.group,'15minpH6.65'),:);
base20min = fig2data(strcmp(fig2data.group,'20minpH7.5') ,:);
acid20min = fig2data(strcmp(fig2data.group,'20minpH6.65'),:);


% verify digitization
figure(1)
semilogy(base10min.minute,base10min.survival)
hold
semilogy(acid10min.minute,acid10min.survival)
semilogy(base15min.minute,base15min.survival)
semilogy(acid15min.minute,acid15min.survival)
semilogy(base20min.minute,base20min.survival)
semilogy(acid20min.minute,acid20min.survival)


% setup curve fit
Ea0 = optimvar('Ea0','LowerBound',0);
Ea1 = optimvar('Ea1','LowerBound',0);
frequencyfactor = 3.1e98;
deltaTheat10min = [10;10;10];
deltaTheat15min = [15;15;15];
deltaTheat20min = [20;20;20];
pHheat = 7.5;
pHincbase = 7.5;
pHincacid = 6.65;
GasConst  = 8.314 ; % J/K / mol
Theat      = 45.5 + 273; % K
Tinc       = 37.0 + 273; % K

disp('build objective function')
mycostfcn10min = sum((log( base10min.survival.^(-1)) - ...
   deltaTheat10min * frequencyfactor * exp(-(Ea0+Ea1* pHheat)/
(GasConst * Theat     )) - ...
   base10min.minute * frequencyfactor * exp(-(Ea0+Ea1* pHincbase)/
(GasConst * Tinc))).^2) + ...
                 sum((log( acid10min.survival.^(-1)) - ...
   deltaTheat10min * frequencyfactor * exp(-(Ea0+Ea1* pHheat)/
(GasConst * Theat     )) - ...
```

```
   acid10min.minute * frequencyfactor * exp(-(Ea0+Ea1* pHincacid)/
(GasConst * Tinc))).^2) ;
mycostfcn15min = sum((log( base15min.survival.^(-1)) - ...
   deltaTheat15min * frequencyfactor * exp(-(Ea0+Ea1* pHheat)/
(GasConst * Theat     )) - ...
   base15min.minute * frequencyfactor * exp(-(Ea0+Ea1* pHincbase)/
(GasConst * Tinc))).^2) + ...
                 sum((log( acid15min.survival.^(-1)) - ...
   deltaTheat15min * frequencyfactor * exp(-(Ea0+Ea1* pHheat)/
(GasConst * Theat     )) - ...
   acid15min.minute * frequencyfactor * exp(-(Ea0+Ea1* pHincacid)/
(GasConst * Tinc))).^2) ;
mycostfcn20min = sum((log( base20min.survival.^(-1)) - ...
   deltaTheat20min * frequencyfactor * exp(-(Ea0+Ea1* pHheat)/
(GasConst * Theat     )) - ...
   base20min.minute * frequencyfactor * exp(-(Ea0+Ea1* pHincbase)/
(GasConst * Tinc))).^2) + ...
                 sum((log( acid20min.survival.^(-1)) - ...
   deltaTheat20min * frequencyfactor * exp(-(Ea0+Ea1* pHheat)/
(GasConst * Theat     )) - ...
   acid20min.minute * frequencyfactor * exp(-(Ea0+Ea1* pHincacid)/
(GasConst * Tinc))).^2) ;
mycostfcn = mycostfcn10min + mycostfcn15min + mycostfcn20min;
show(mycostfcn )


disp('create optim prob')
convprob = optimproblem('Objective',mycostfcn );
%show(convprob)
problem =
 prob2struct(convprob,'ObjectiveFunctionName','generatedObjective');
```

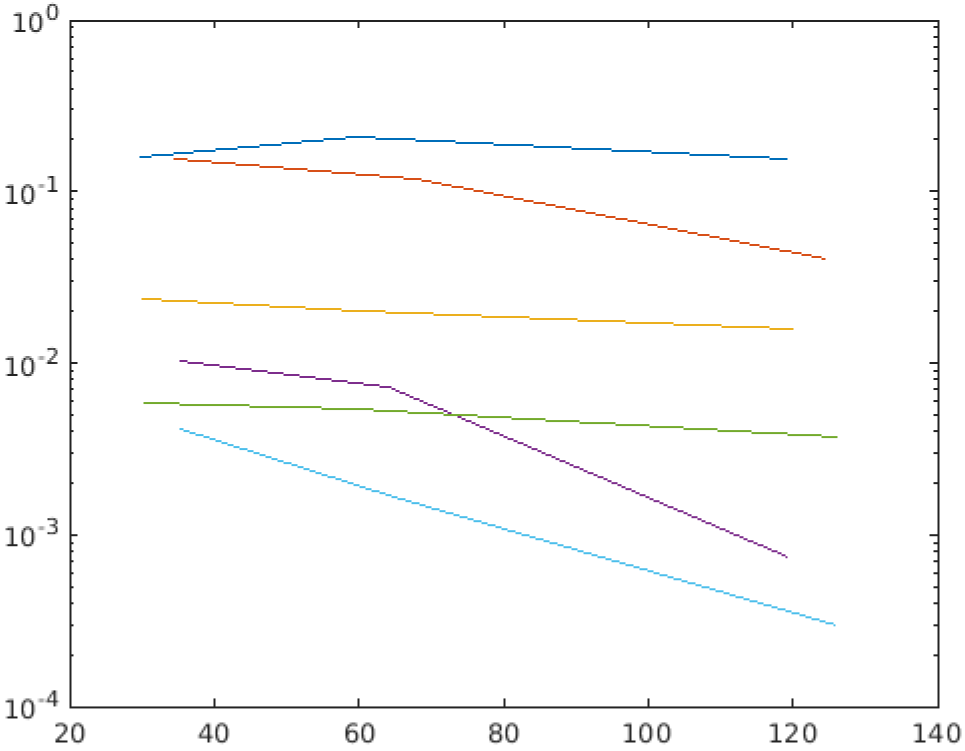
```
fig2data =

  18x4 table

         minute                 survival           tuple         group

    ________________    ____________________    _____
 _______________

    29.7727272727273       0.159985871960606       0
 {'10minpH7.5' }
    34.5454545454545       0.155051577983263       0
 {'10minpH6.65'}
    30.1136363636364      0.0236673514472524       0
 {'15minpH7.5' }
    35.2272727272727      0.0101578716522971       0
 {'15minpH6.65'}
    30.4545454545455     0.00577969288415332       0
 {'20minpH7.5' }
    35.2272727272727     0.00415956216307185       0
 {'20minpH6.65'}
```

```
    59.7727272727273       0.208798748450475      1
 {'10minpH7.5' }
    67.9545454545455        0.11880369057099      1
 {'10minpH6.65'}
    60.1136363636364      0.0202358964772516      1
 {'15minpH7.5' }
    64.2045454545455     0.00719685673001153      1
 {'15minpH6.65'}
    59.7727272727273     0.00534430402858677      1
 {'20minpH7.5' }
    64.8863636363636     0.00165077192763291      1
 {'20minpH6.65'}
    119.431818181818       0.155051577983263      2
 {'10minpH7.5' }
    124.545454545455      0.0403127269426998      2
 {'10minpH6.65'}
    120.113636363636      0.0157499402864011      2
 {'15minpH7.5' }
    119.431818181818    0.000731047469634276      2
 {'15minpH6.65'}
              126.25     0.00372759372031494      2
 {'20minpH7.5' }
    125.909090909091    0.000294705170255181      2
 {'20minpH6.65'}

Current plot held
build objective function

  (((sum(((extraParams{2} - (extraParams{1} .* exp((-(Ea0
+ (Ea1 .* 7.5)) ./ 2648.009)))) - arg1).^2) + arg3) + arg7) + arg11)

  where:

    arg1 = (extraParams{3} .* exp((-(Ea0 + (Ea1 .* 7.5)) ./
 2577.34)));
    arg2 = (extraParams{6} .* exp((-(Ea0 + (Ea1 .* 6.65)) ./
 2577.34)));
    arg3 = sum(((extraParams{5} - (extraParams{4} .* exp((-(Ea0
+ (Ea1 .* 7.5)) ./ 2648.009)))) - arg2).^2);
    arg4 = (extraParams{9} .* exp((-(Ea0 + (Ea1 .* 7.5)) ./
 2577.34)));
    arg5 = (extraParams{12} .* exp((-(Ea0 + (Ea1 .* 6.65)) ./
 2577.34)));
    arg6 = sum(((extraParams{11} - (extraParams{10} .* exp((-(Ea0
+ (Ea1 .* 7.5)) ./ 2648.009)))) - arg5).^2);
    arg7 = (sum(((extraParams{8} - (extraParams{7} .* exp((-(Ea0
+ (Ea1 .* 7.5)) ./ 2648.009)))) - arg4).^2) + arg6);
    arg8 = (extraParams{15} .* exp((-(Ea0 + (Ea1 .* 7.5)) ./
 2577.34)));
    arg9 = (extraParams{18} .* exp((-(Ea0 + (Ea1 .* 6.65)) ./
 2577.34)));
    arg10 = sum(((extraParams{17} - (extraParams{16} .* exp((-(Ea0
+ (Ea1 .* 7.5)) ./ 2648.009)))) - arg9).^2);
    arg11 = (sum(((extraParams{14} - (extraParams{13} .* exp((-(Ea0
+ (Ea1 .* 7.5)) ./ 2648.009)))) - arg8).^2) + arg10);

    extraParams{1}:

   1.0e+99 *

   3.100000000000000
   3.100000000000000
   3.100000000000000

  extraParams{2}:

   1.832669767893219
   1.566384417002751
   1.863997456233274

  extraParams{3}:

  1.0e+100 *

   0.922954545454545
   1.852954545454545
   3.702386363636363

  extraParams{4}:

   1.0e+99 *

   3.100000000000000
   3.100000000000000
   3.100000000000000

  extraParams{5}:

   1.863997456233274
   2.130282807123742
   3.211088054855641

  extraParams{6}:

  1.0e+100 *

   1.070909090909091
   2.106590909090909
   3.860909090909090

  extraParams{7}:

   1.0e+99 *

   4.650000000000000
   4.650000000000000
   4.650000000000000
```

```
extraParams{8}:

 3.743658756636577
 3.900297198336852
 4.150918705057292

extraParams{9}:

1.0e+100 *

 0.933522727272727
 1.863522727272727
 3.723522727272727

extraParams{10}:

 1.0e+99 *

 4.650000000000000
 4.650000000000000
 4.650000000000000

extraParams{11}:

 4.589506341818064
 4.934110913558668
 7.221032162382686

extraParams{12}:

1.0e+100 *

 1.092045454545454
 1.990340909090909
 3.702386363636363

extraParams{13}:

 1.0e+99 *

 6.200000000000000
 6.200000000000000
 6.200000000000000

extraParams{14}:

 5.153404731939053
 5.231723952789192
 5.591992368699825

extraParams{15}:

1.0e+100 *
```

```
 0.944090909090909
 1.852954545454545
 3.913749999999999

extraParams{16}:

 1.0e+99 *

 6.200000000000000
 6.200000000000000
 6.200000000000000

extraParams{17}:

 5.482345459509632
 6.406512265541255
 8.129535124244283

extraParams{18}:

1.0e+100 *

 1.092045454545454
 2.011477272727273
 3.903181818181818
```

create optim prob



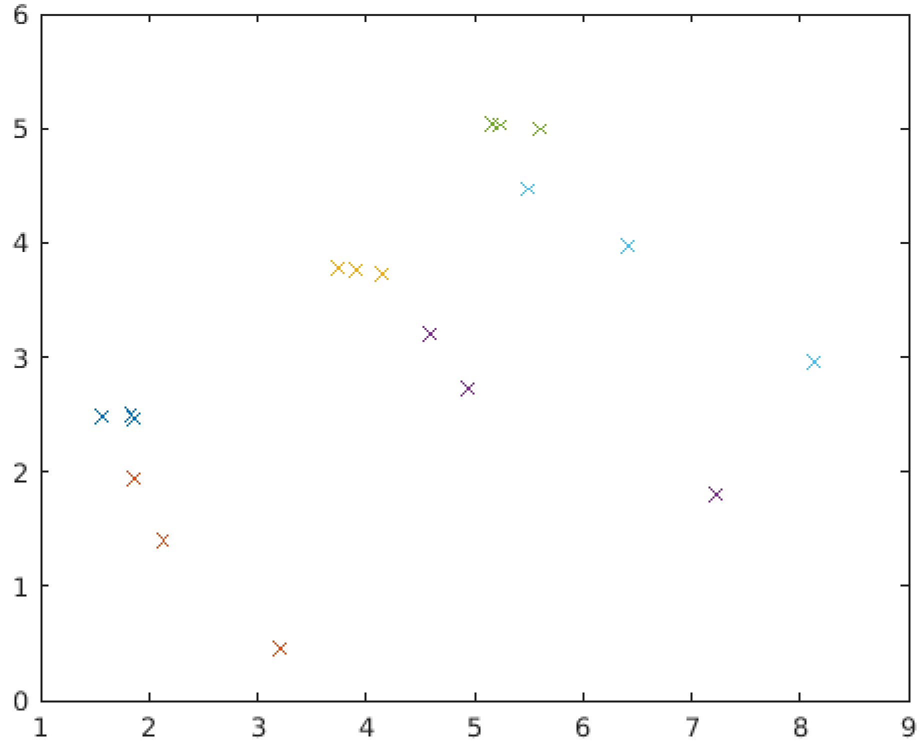
Solve the new problem. The solution is essentially the same as before.

```
myoptions = optimoptions(@lsqnonlin,'Display','iter-detailed');
x0.Ea0 = 6.28e5; % J/mol
x0.Ea1 = 0;
[popt,fval,exitflag,output] =
 solve(convprob,x0,'Options',myoptions, 'solver','lsqnonlin' )


%evaluate fit
survivalpredictionbase10min = deltaTheat10min * frequencyfactor
 * exp(-(popt.Ea0+popt.Ea1* pHheat)/(GasConst * Theat     )) -
 base10min.minute * frequencyfactor * exp(-(popt.Ea0+popt.Ea1*
 pHincbase)/(GasConst * Tinc))
survivalpredictionacid10min = deltaTheat10min * frequencyfactor
 * exp(-(popt.Ea0+popt.Ea1* pHheat)/(GasConst * Theat     )) -
 acid10min.minute * frequencyfactor * exp(-(popt.Ea0+popt.Ea1*
 pHincacid)/(GasConst * Tinc))
survivalpredictionbase15min = deltaTheat15min * frequencyfactor
 * exp(-(popt.Ea0+popt.Ea1* pHheat)/(GasConst * Theat     )) -
 base15min.minute * frequencyfactor * exp(-(popt.Ea0+popt.Ea1*
 pHincbase)/(GasConst * Tinc))
survivalpredictionacid15min = deltaTheat15min * frequencyfactor
 * exp(-(popt.Ea0+popt.Ea1* pHheat)/(GasConst * Theat     )) -
 acid15min.minute * frequencyfactor * exp(-(popt.Ea0+popt.Ea1*
 pHincacid)/(GasConst * Tinc))
```

```
survivalpredictionbase20min = deltaTheat20min * frequencyfactor
 * exp(-(popt.Ea0+popt.Ea1* pHheat)/(GasConst * Theat     )) -
 base20min.minute * frequencyfactor * exp(-(popt.Ea0+popt.Ea1*
 pHincbase)/(GasConst * Tinc))
survivalpredictionacid20min = deltaTheat20min * frequencyfactor
 * exp(-(popt.Ea0+popt.Ea1* pHheat)/(GasConst * Theat     )) -
 acid20min.minute * frequencyfactor * exp(-(popt.Ea0+popt.Ea1*
 pHincacid)/(GasConst * Tinc))


figure(2)
plot(log( base10min.survival.^(-1)),
 survivalpredictionbase10min , 'x',log( acid10min.survival.^(-1)),
 survivalpredictionacid10min , 'x', log( base15min.survival.^(-1)),
 survivalpredictionbase15min , 'x',log( acid15min.survival.^(-1)),
 survivalpredictionacid15min , 'x', log( base20min.survival.^(-1)),
 survivalpredictionbase20min , 'x',log( acid20min.survival.^(-1)),
 survivalpredictionacid20min , 'x' )
```

```
Solving problem using lsqnonlin.

                                            Norm of      First-order
 Iteration  Func-count      f(x)            step          optimality
     0           3        393.928                               9.49
     1           6        392.413              10                187
     2           9         33.334              20           1.25e+04
     3          12        21.5733         0.74113             0.0225
     4          15        21.4099              40           0.000226
     5          18        21.4074              80                9.3
     6          21        21.3693             160               3.94
     7          24        21.2872             320               7.13
     8          27        21.0962             640               11.2
     9          30        20.5806            1280               2.84
    10          33        18.7545            2560            0.00297
    11          36         18.024         5119.83               62.8
    12          39        14.5708         90.1327              0.104
    13          42        11.3558         2674.97            0.00474
    14          45        11.3465         347.228           2.58e-05
    15          48        11.3465         3.31512           6.43e-11

Optimization completed: The first-order optimality measure,
 6.433260e-11,
is less than options.OptimalityTolerance = 1.000000e-06.


popt =

  struct with fields:

    Ea0: 5.238511617911937e+05
    Ea1: 1.070877476847831e+04
```

```
fval =

  11.346472834416707


exitflag =

    OptimalSolution


output = 

  struct with fields:

    firstorderopt: 6.433260001104113e-11
       iterations: 15
        funcCount: 48
     cgiterations: 0
        algorithm: 'trust-region-reflective'
         stepsize: 3.315116633819008
          message: '...'
           solver: 'lsqnonlin'


survivalpredictionbase10min =

   2.515921886037445
   2.501354792749117
   2.472386141323463


survivalpredictionacid10min =

   1.956986596938427
   1.402456151108548
   0.463149477559978


survivalpredictionbase15min =

   3.780945662165661
   3.766378568877332
   3.737244382300675


survivalpredictionacid15min =

   3.210858960343567
   2.729888675695203
   1.813215897894791


survivalpredictionbase20min =
   5.045969438293875
   5.031733415307554
   4.999454060861827


survivalpredictionacid20min =

   4.476048271622785
   3.983761039100342
   2.970894204370258
```



Published with MATLAB® R2020a