Electronic Supplementary Material (ESI) for Analytical Methods.

Supplementary Materials:

A surface-based calibration approach to enable dynamic and more accurate quantification of colorimetric assay systems

Lin Tong*[a] and Joshua D. Hutcheson[a]

[a]Department of Biomedical Engineering, Florida International University, 10555 W Flagler St, EC 2612, Miami, FL, 33174

1. **Buffer solution:** 0.1M sodium acetate buffer pH=5.6 was prepared using the AAT Bioquest protocol18. 10mL 0.05M sodium acetate buffer was prepared by mixing 5mL 0.1M sodium acetate buffer with 5mL deionized water with a final pH of 5.55. The 0.05M sodium acetate buffer solution was stored at 4°C overnight. 0.01M Phosphate buffered saline (PBS) was diluted from 0.1M 10X PBS stock using deionized water.
2. **Reaction cocktail solution:** To avoid measurement errors, we prepared the cocktail solution as following steps.
   Step 1: During the measurement, we took 5.4mg glucose oxidase from the container, so we dissolved this in 1mL 0.05M cold sodium acetate buffer to obtain 5.4mg/mL glucose oxidase solution. Then covered the vail by foil and stored at 4°C.
   Step 2: Similarly, 1.1mg/mL horseradish peroxidase (HRP) solution was prepared by dissolving 1.1mg HRP in 1mL 0.05M cold sodium acetate buffer. Then covered the vail by foil and stored at 4°C.
   Step 3: Through calculation, 222µL 5.4mg/mL glucose oxidase solution and 110µL 1.1mg/mL HRP was added to 668µL 0.05M cold sodium acetate buffer. This is equivalent to add 1.2mg glucose oxidase and 0.12 mg HRP into 1mL 0.05M cold sodium acetate buffer.
   Step 4: 105.2mg trehalose, 171mg sucrose was then added into above solution and mixed well.
   Step 5: 92mg sodium iodide was added into the mixture and mixed well. The final cocktail was covered with foil and will be used immediately.
3. **Notes for preparing invertase standard solution:** Pipette tips need to be changed between each dilution step. Each dilution step is preferably less than 10 times.
4. **Camera requirement**
   A standard 8-megapixel camera was used to gather data for the current study, and we acquired the images of the pads at 6-megapixels (3264×1836 pixels). The effective image area of 24 paper pads in 8 rows and 3 columns was only 840×340 pixels. Each paper pad diameter was approximately 81 pixels. This should be easily achieved by almost any CCD devices currently available, including smart phones, tablets, or web cameras. We conducted similar paper pad experiments with two smart phones (rear cameras on Huawei Honor 6 pro C8817D and Apple iPhone SE MHGE3LL/A) and one tablet (Microsoft surface pro 2017 rear camera), and all devices yielded similar results. We recommend using the same or higher resolution per paper pad to have enough pixels for image processing.
5. **Image processing**
   After applying the 24-circular mask, we divided the saturation channel image into 24 parts. For each piece, we have performed a colormap for better visualization. We also did a statistical analysis and observed the histogram of saturation value distribution of all pixels in a paper pad at each capture time.

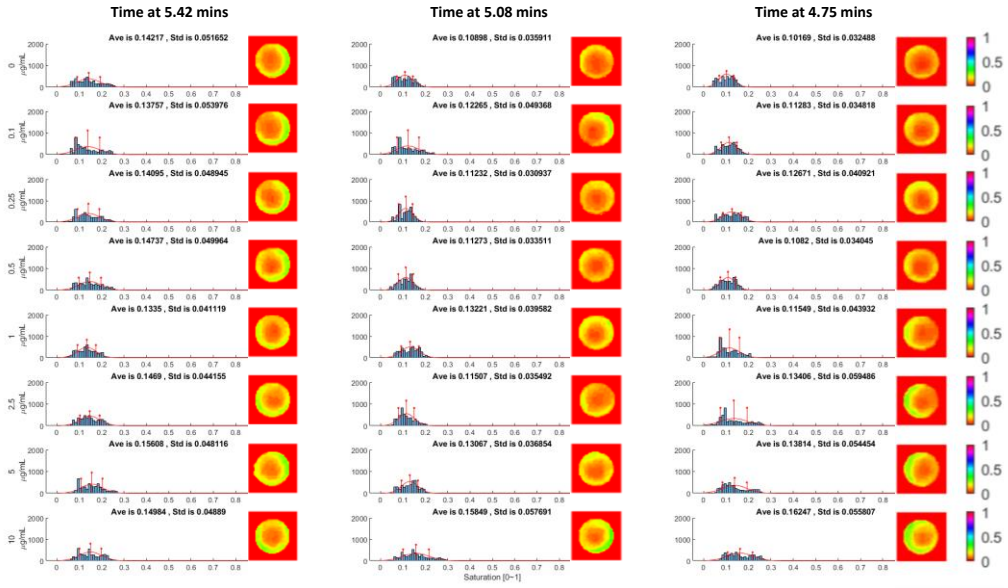

Fig. S1. Histogram of saturation values of all pixels in all 24 paper pads. The red line is the normal distribution with average value and standard deviation. The X-axis is the saturation value. The assay time of each column is 5.42, 5.08 and 4.75 minutes. For each column, the reagent was added 20s after the previous column. The concentration of each row is 0, 0.1,0.25,0.5,1,2.5,5,10 µg/mL respectively. The paper pad image is the saturation channel with applied colormap.

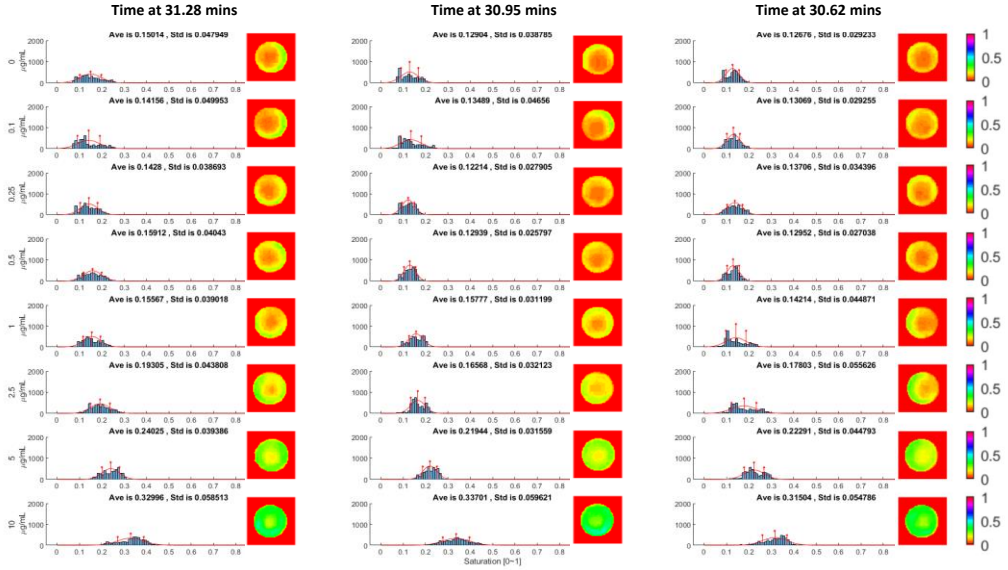

Fig. S2. Histogram of saturation values of all pixels in all 24 paper pads. The red line is the normal distribution with average value and standard deviation. The X-axis is the saturation value. The assay time of each column is 31.28, 30.95 and 30.62 minutes. For each column, the reagent was added 20s after the previous column. The concentration of each row is 0, 0.1,0.25,0.5,1,2.5,5,10 µg/mL respectively. The paper pad image is the saturation channel with applied colormap.

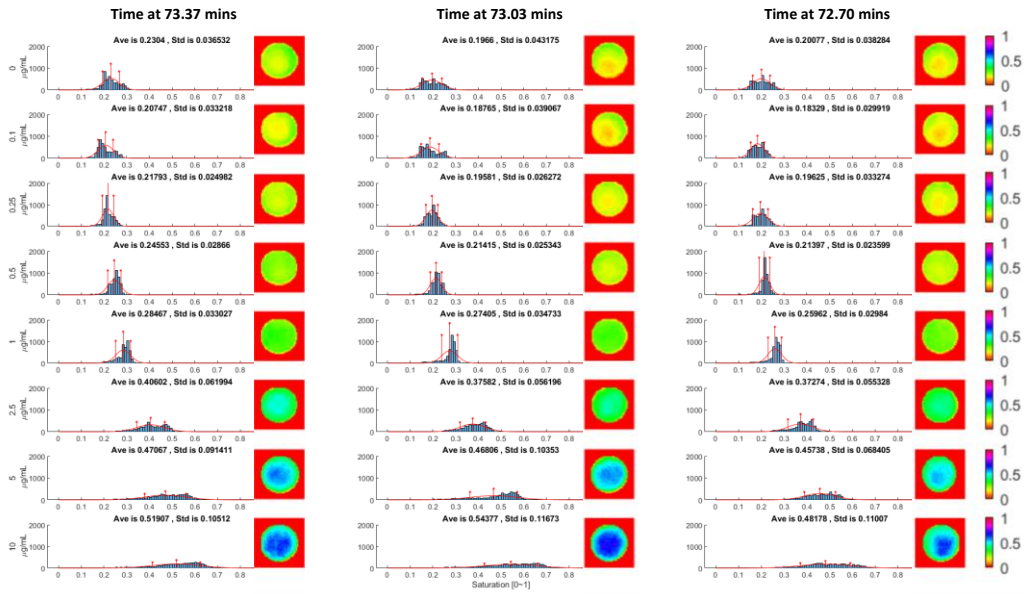

Fig. S3. Histogram of saturation values of all pixels in all 24 paper pads. The red line is the normal distribution with average value and standard deviation. The X-axis is the saturation value. The assay time of each column is 73.37, 73.03 and 72.70 minutes. For each column, the reagent was added 20s after the previous column. The concentration of each row is 0, 0.1,0.25,0.5,1,2.5,5,10 µg/mL respectively. The paper pad image is the saturation channel with applied colormap.

6. In order to observe the trend of the reaction, we also plotted the average saturation value of all pixels in the paper pads (raw data) vs time, and the average of saturation value of all pixels in the paper pads (raw data) vs concentration.

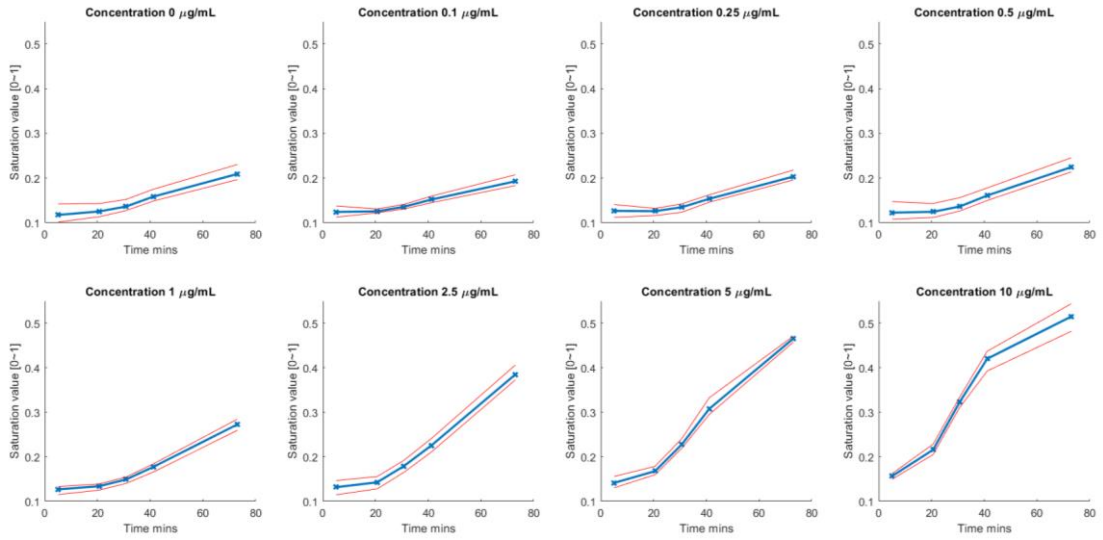

Figure S4. Saturation raw data vs time. The blue line was the average value for each time. Red range indicates the maximum and minimum values.

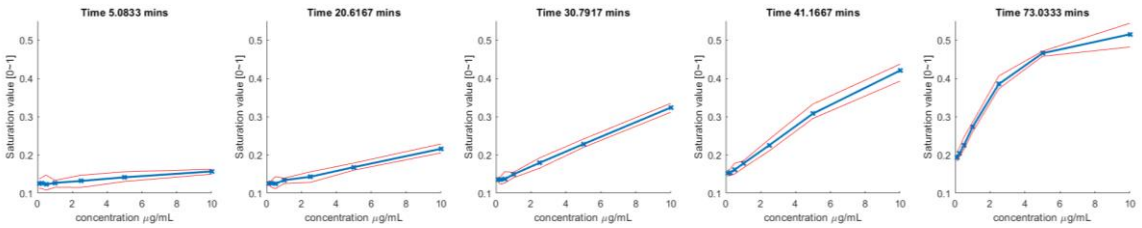

Figure S5. Saturation raw data vs concentration. The blue line was average value for each concentration. Red range indicated the maximum and minimum values.

7. From Fig. S6, we could see our designed cost function converged. The training was terminated at 640 steps of iteration with a cost value of 4.7337×10^{-5}.

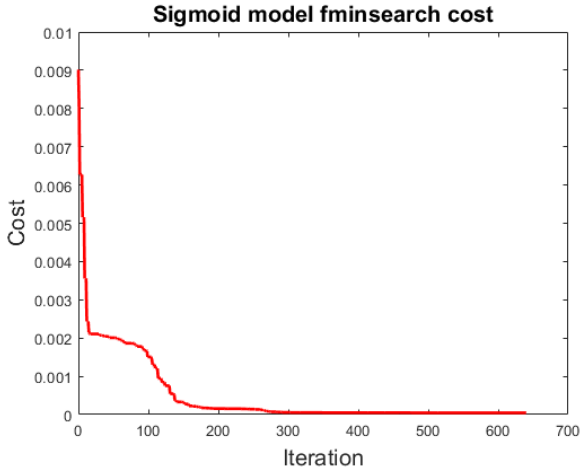

Fig. S6. Sigmoid Model Training Cost vs Iteration

8. The RGB and HSV color space.

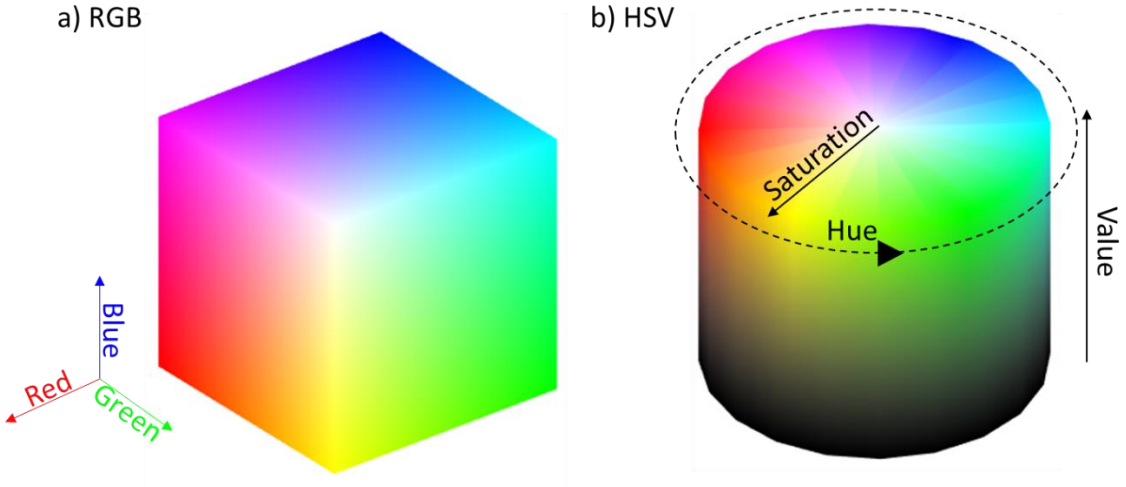

Figure S7. a) RGB color space and b) HSV color space.

Attachement:

Matlab Code:

1. Color space conversion and plot

```
% compare RBG, HSV and grey
clear; close all;
RGB = imread('30min2.jpg');
GRAY=rgb2gray(RGB);
HSV = rgb2hsv(RGB);
S=HSV(:,:,2);
figure
subplot(1,8,1)
imshow(RGB)
title('Raw Image')
subplot(1,8,2)
imshow(GRAY)
title('GRAY Image')
subplot(1,8,3)
imshow(RGB(:,:,1))
title('Red Channel')
subplot(1,8,4)
imshow(RGB(:,:,2))
title('Green Channel')
subplot(1,8,5)
imshow(RGB(:,:,3))
title('Blue Channel')
subplot(1,8,6)
imshow(HSV(:,:,1))
title('Hue Channel')
subplot(1,8,7)
imshow(HSV(:,:,2))
title('Saturation Channel')
subplot(1,8,8)
imshow(HSV(:,:,3))
title('Value Channel')
```

2. **Correlation Coefficients Calculation:**

```
% calculate the average value for each paper pads at about 30mins
H=HSV(:,:,1);
S=HSV(:,:,2);
V=HSV(:,:,3);
R=RGB(:,:,1);
G=RGB(:,:,2);
B=RGB(:,:,3);
sz=size(S);
l=round(sz(1)/8:sz(1)/8:sz(1));
w=round(sz(2)/3:sz(2)/3:sz(2));
w1=[1,w];%(4 number)
l1=[1,l];%(9 number)
conc = [0.1 0.25 0.5 1 2.5 5 10];
for i=1:length(l)
   for j=1:length(w)
      Idx=3*(i-1)+j;
      temp=S(l1(i):l(i),w1(j):w(j));
      temp2=temp(temp>0&temp<1);%delete artifical white point S=1;
      region(Idx).Save=nanmean(temp2);
         clear temp temp2
```

```
        temp=double(B(l1(i):l(i),w1(j):w(j)))/255;
        temp2=temp(temp>0&temp<1);%delete artifical white point S=1;
        region(Idx).Bave=nanmean(temp2);
          clear temp temp2
        temp=double(R(l1(i):l(i),w1(j):w(j)))/255;
        temp2=temp(temp>0&temp<1);%delete artifical white point S=1;
        region(Idx).Rave=nanmean(temp2);
          clear temp temp2
        temp=double(G(l1(i):l(i),w1(j):w(j)))/255;
        temp2=temp(temp>0&temp<1);%delete artifical white point S=1;
        region(Idx).Gave=nanmean(temp2);
          clear temp temp2
        temp=H(l1(i):l(i),w1(j):w(j));
        temp2=temp(temp>0&temp<1);%delete artifical white point S=1;
        region(Idx).Have=nanmean(temp2);
              clear temp temp2
        temp=V(l1(i):l(i),w1(j):w(j));
        temp2=temp(temp>0&temp<1);%delete artifical white point S=1;
        region(Idx).Vave=nanmean(temp2);
         clear temp temp2
        temp=double(GRAY(l1(i):l(i),w1(j):w(j)))/255;
        temp2=temp(temp>0&temp<1);%delete artifical white point S=1;
        region(Idx).GRAYave=nanmean(temp2);
    end
end

% calculate raw data's correlation coefficients
AveCorrMatrix=[conc cell2mat({region.GRAYave})' cell2mat({region.Rave})'...
    cell2mat({region.Gave})' cell2mat({region.Bave})'...
    cell2mat({region.Have})' cell2mat({region.Save})' cell2mat({region.Vave})'];
[R,p]=corrcoef(AveCorrMatrix(4:end,:));
R_sqr=R(1,:).^2

% calculate removed background data's correlation coefficients
% BG has the same calculation method as region, but from image captured at about 5mins
AveBGMatrix=[cell2mat({BG.GRAYave})' cell2mat({BG.Rave})'...
    cell2mat({BG.Gave})' cell2mat({BG.Bave})'...
    cell2mat({BG.Have})' cell2mat({BG.Save})' cell2mat({BG.Vave})'];
temp=AveCorrMatrix-AveBGMatrix;
temp=[conc temp];
[R,p]=corrcoef(temp(4:end,:));
R_sqr=R(1,:).^2
```

3. Linear Regression Training

```
% Linear Regression Training
orT =[5.0833   20.6167   30.7917   41.1667   73.0333];
rTime=[orT+20/60;orT;orT-20/60];
zt=repmat(rTime,5,1);
orC =[0.1000    0.5000    1.0000    2.5000   10.0000];
temp=repmat(orC,3,1);
zc=repmat(temp(:),5,1);
Normalization_factor=max(X);
XN=X./Normalization_factor;
lambda=0.01 ;
degree = 3
XN28 = mapFeature(XN(:,1), XN(:,2), degree);
[m, n] = size(XN28);
initial_theta = zeros(n,1);
```

```
temp = eye(size(XN28,2));
temp(1,1)=0;
theta=pinv(XN28'*XN28+lambda*temp)*XN28'*y;

% mapFeature
function out = mapFeature(X1, X2, degree)
out = ones(size(X1(:,1)));
for i = 1:degree
   for j = 0:i
      out(:, end+1) = (X1.^(i-j)).*(X2.^j);
   end
end
end

% calculate error
prediction=XN28*theta
error=prediction-y;
E=sum(sum(real(error).^2));

% mesh calculation and plot
mTime=0:0.2:74;
mConc=0:0.05:10;
LcalMesh.time=mTime;
LcalMesh.concentration=mConc;
u=mTime/Normalization_factor(1);
v=mConc/Normalization_factor(2);
z=zeros(length(mTime),length(mConc));
for i=1:length(mTime)
   for j=1:length(mConc)
      z(i,j)=mapFeature(u(i),v(j),degree)*theta;
   end
end
z=z';
LcalMesh.mesh=z;
figure
mesh(mTime,mConc,z)
hold on
[T,C]=meshgrid(orT,orC)
% orData is the average of 3 duplication of removed BG data
plot3(T(:),C(:),orData(:),'ko','markerfacecolor','y','markersize', 5)
xlabel('Time in mins','fontsize',16)
ylabel('Concentration in \mug/mL','fontsize',16)
zlabel('Saturation Value','fontsize',16)
title(['CalMesh Linear Regression model. Error: ' num2str(E) ],'fontsize',16)
grid on

% costFunction
function [J, grad] = costFunction(theta, X, y, lambda)

if exist('lambda','var')==0
   lambda=0;
end
m= length(y);
predictions=X*theta;
sqrError=(predictions-y).^2;
J=1/(2*m)*sum(sqrError)+lambda/(2*m)*theta(2:end)'*theta(2:end);

temp=theta;
```

```
temp(1)=0;
grad=1/m*X'*(X*theta-y)+lambda/m*temp;

end
```

4. Sigmoid Training

```
% Sigmoid Training
options = optimset('MaxFunEvals',300*length(zt),'MaxIter',300*length(zt));
[k,fval,exitflag,output] = fminsearch(@(K) myFun1(K,zc,zt,data), k0,options);

% calculate error
z=sqrt(k(5)*zc+zt+(zc.^k(6)).*zt);
x6=k(1)./(k(2)+exp(-(k(3)*(z-k(4)))))+k(7);
error=x6-data;
E=sum(sum(real(error).^2));

% myFun1 (cost function)
function F=myFun1(k,zc,zt,ydata)
z=sqrt(k(5)*zc+zt+(zc.^k(6)).*zt);
x6=k(1)./(k(2)+exp(-(k(3)*(z-k(4)))))+k(7);
error=x6-ydata;
F=sum(sum(error.^2));
End

% mesh calculation and plot
zt=0:0.2:74;
zc=0:0.05:10;
calMesh.time=zt;
calMesh.concentration=zc;
[zt,zc]=meshgrid(zt, zc);
z=sqrt(k(5)*zc+zt+(zc.^k(6)).*zt);
x6=k(1)./(k(2)+exp(-(k(3)*(z-k(4)))))+k(7);
calMesh.mesh=real(x6);

figure;
mesh(zt0, zc, abs(x6))
hold on
[T,C]=meshgrid(orT,orC)
% orData is the average of 3 duplication of removed BG data
plot3(T(:),C(:),orData(:),'ko','markerfacecolor','y','markersize', 5)
xlabel('Time in mins','fontsize',16)
ylabel('Concentration in \mug/mL','fontsize',16)
zlabel('Saturation Value','fontsize',16)
title('Sigmoid model fminsearch cost','fontsize',15)
```

5. Algorithm for using calibration mesh

```
% calMesh=LcalMesh; % Do this for Linear Regression Model
R=3; % each sample has 3 duplication inputs;
Time=rTime(:,2:end);
Input=D025RBG(:,2:end); % This is for 0.25 ug/mL test data set
L=size(Time,2);
ARD=[];%abs(current concentration- previous concentration)/current concentration
C=[];
c=[];
% initiation
i=1;
ARD(i)=nan;
```

```
c(1:R,i)=numCalMesh(Time(1:R,i),Input(1:R,i),calMesh);
C=[C;mean(c(:,end))];

i=i+1;

while (true)
   c(1:R,i)=numCalMesh(Time(1:R,i),Input(1:R,i),calMesh);
   C=[C;mean(c(:,end))];

   ARD(i)=abs(C(i)-C(i-1))/C(i);
   if ARD(i)<0.05 %stop condition
      disp( ['Current time is ' ,num2str(mean(Time(:,i))), ' min.'])
      disp( ['Concentration is ', num2str(C(i)), ' ug/mL.'])
      C
      break;
   end
   if i==L %stop condition
      disp( ['Time up or manual stopped. Current time is ', num2str(mean(Time(:,i))), ' min.'])
      disp( ['Concentration is ', num2str(C(i)), ' ug/mL.'])
      C
      break;
   end
   i=i+1;
end

% numCalMesh (mapping function for calibration mesh)
function out=numCalMesh(t,d,calMesh)
% updated; can used for array input
[value IT] = min(abs(calMesh.time-t),[],2);
[value IC] = min(abs(calMesh.mesh(:,IT)-d'));
out=calMesh.concentration(IC)';
```