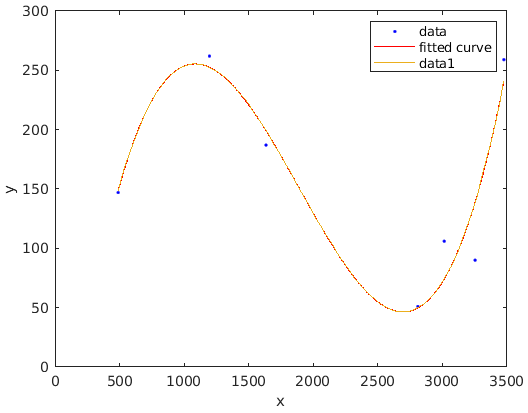
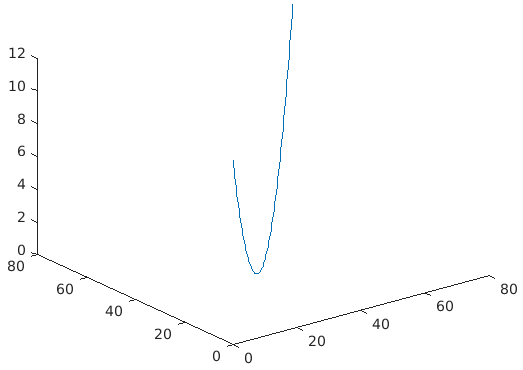
```
%load('microarray sources.txt')
%load('TF Z-score.txt')
%file1 = 'MATLAB Drive/TF Z-score.txt';
%TFZ_score = importtxt(file1, 1);
%file2 = 'MATLAB Drive/microarray sources.txt';
%Coordinates = importtxt(file2, 2);



name = ['Hax1'];
steps = 50;
tol = 1;
T = 101;
a1(1,:) = [489, 1196, 1634, 2810, 3014, 3253, 3476];
a(1,:) = [636, 1458, 1821, 2861, 3120, 3343, 3735];
a2(1,:) = a - a1;
Res = zeros(1, steps);
dTau = zeros(1, steps);
Cost = zeros(1, steps);
[dtau, ahat, QHat,lambdahat, H_theta_hat, SHat, H_hat, T_hat, H, dTau, Res, Cost, xhat,
```




```
Warning: Matrix is close to singular or badly scaled. Results may be inaccurate. RCOND =
 6.907473e-24.
Warning: Matrix is close to singular or badly scaled. Results may be inaccurate. RCOND =
 2.464300e-23.
Warning: Matrix is close to singular or badly scaled. Results may be inaccurate. RCOND =
 4.254030e-21.
Warning: Matrix is close to singular or badly scaled. Results may be inaccurate. RCOND =
 2.505583e-21.
```

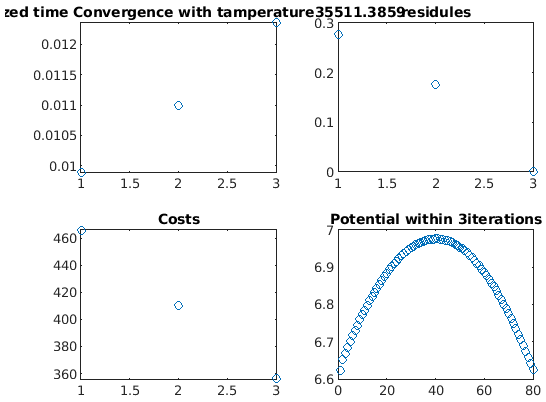


```
% while tol >= 0.05 & steps > 0 & T >= 10
%     dx = (a1(length(a1))-a1(1))/T;
%     [f1, gof, output] = fit(a1', a2', 'poly3', 'Normalize','on','Robust','Bisquare');
%     x = a1(1):dx:a1(length(a1));
%     y = f1(a1(1):dx:a1(length(a1)));
%     if steps == 1 | steps == 50
%         figure,
%         plot(f1, a1, a2);
%         hold on
%
%         plot(x, y);
%     end
%     m = 1;
%     n = 4;
%
%     b = zeros(1, T-1);
%     a = b;
%     sigma = a;
%     Qhat = zeros(T-1);
%     alpha = ones(size(x))./(1+exp(-(x-mean(x))/max(x)));%sigmoid(x)
%     for t = 1:T-1
%         b(t) = x(t+1) - x(t);
%         bphi(t) = alpha(t+1) -alpha(t);
%         sigma(t) = std(x(1:t+1));
%     end
%     a = sigma'*sigma;
%     Qhat = inv(a)*(abs(b(1:length(a)))./abs(y(1:length(a)))*y(1:length(a))-b);
%     lambda = abs(b(1:length(a)))./abs(y(1:length(a)));
%     %dphi = abs((alpha(2:T)'-alpha(1:T-1)')/dx./(mean(a(1:T-1,:),2)-mean([zeros(1,T-1
%     %bphi = abs((alpha(2:T)'-alpha(1:T-1)')./(mean(a(1:T-1,:),2)-mean([zeros(1,T-1);a
%     dphi = (alpha(2:T)'-alpha(1:T-1)')/dx;
%     bphi = alpha(2:T)'-alpha(1:T-1)';
%     steps = steps -1;
%     tol = abs(mean(alpha)-mean(mean(lambda(unique(round(alpha*100)), unique(round(bph
%     Res(50-steps) = tol;
%     Cost(50-steps) = T*log(T);
%     dTau(50-steps) = 1/T;
%     if tol < 0.05 | steps == 0 | T <= 10
%         if steps < 49
%             if (Res(50-steps) == Res(50-steps-1)) | ((Res(50-steps) == Res(50-steps-2
% %               alpha = alpha + bphi;
% %               xhat = alpha*100;
% %               yhat = f1(xhat);
% %               alphahat = 1./(1+exp(-(xhat-mean(xhat ))/max(xhat)));%sigmoid(x)
% %               for t = 1:T-1
% %                   bhat(t) = xhat(t+1) - xhat(t);
% %                   bphihat(t) = alphahat(t+1) -alphahat(t);
% %                   sigmahat(t) = std(xhat(1:t+1));
% %               end
% %               dtau = 1/T;
% %               ahat = sigmahat'*sigmahat;
% %               QHat = inv(ahat)*(abs(bhat(1:length(ahat)))./abs(yy(1:length(ahat))
% %               lambdahat = abs(bhat(1:length(ahat)))./abs(yy(1:length(ahat)));
% %
% %             %    H_theta = (alpha - alphahat)/dtau;
% %               H_theta_hat = lambdahat * bphihat;
% %               SHat = sum(meshgrid(bphihat , QHat))
```

```
% %                 H_hat = (meshgrid(bphi, Qhat) - SHat)*lambdahat;
% %                 T_hat = 1/(2*lambdahat(1));
% %                 for j = 2: length(lambdahat)-1
% %                     T_hat = T_hat + 1/lambdahat(j);
% %                 end
% %                 T_hat = T_hat + 1/(2*lambdahat(length(lambdahat)));
%                 break
%             end
%         end
%     else
%         T = T - 10;
%     end
% end
% alpha = alpha + [bphi(1),bphi',bphi(length(bphi))];
% xhat = alpha*100;
% yhat = f1(xhat);
% alphahat = ones(size(xhat))./(1+exp(-(xhat-mean(xhat))/max(xhat)));%sigmoid(x)
% for t = 1:T-1
%     bhat(t) = xhat(t+1) - xhat(t);
%     bphihat(t) = alphahat(t+1) -alphahat(t);
%     sigmahat(t) = std(xhat(1:t+1));
% end
% dtau = 1/T;
% ahat = sigmahat'*sigmahat;
% QHat = inv(ahat)*(abs(bhat(1:length(ahat)))./abs(yhat(1:length(ahat)))*yhat(1:length(
% lambdahat = abs(bhat(1:length(ahat)))./abs(yhat(1:length(ahat)));
%
% %    H_theta = (alpha - alphahat)/dtau;
% H_theta_hat = lambdahat * bphihat';
% SHat = sum(meshgrid(bphihat , QHat));
% H_hat = (meshgrid(bphi, Qhat) - SHat)*lambdahat;
% T_hat = 1/(2*lambdahat(1));
% for j = 2: length(lambdahat)-1
%     T_hat = T_hat + 1/lambdahat(j);
% end
% T_hat = T_hat + 1/(2*lambdahat(length(lambdahat)));
%
% H = reshape(H_hat, length(lambdahat),length(lambdahat),length(lambdahat));
% C = mean(mean(H,1),2);
% c = abs(abs(C/max(abs(C)))-1)*255
% figure,
% plot3(1: T-1, 1: T-1, c(:))
%
% H(:,:,1:size(H,3))

figure,
subplot(2,2,1)
plot(dTau(dTau~=0),'o')
title(['optimized time Convergence with tamperature', num2str(T_hat)])
subplot(2,2,2)
plot(Res(Res~=0),'o')
title('residules');
subplot(2,2,3)
plot(Cost(Cost~=0),'o')
title('Costs');
subplot(2,2,4)
%plot(xhat, yhat,'*')
%title(['best path at optimized tamperature',num2str(T_hat)]);
```

```
plot(SHat,'o')
title(['Potential within ', num2str(sum(dTau~=0)), 'iterations'])
```




```
%
% name = ['Arc'];
% steps = 50;
% tol = 1;
% T = 101;
% b1(1,:) = [1527, 4359, 5649, 8619, 11405 ,11695, 12069, 12323, 12601, 12803, 14563, 1
% b(1,:) = [1735, 4692, 5755, 8809, 11525, 11807, 12143, 12400, 12516, 12922, 14692, 15
% b2(1,:) = b - b1;
% m = 1;
% n = 3;
%
% Res = zeros(1, steps);
% dTau = zeros(1, steps);
% Cost = zeros(1, steps);
%
% [bdtau, bahat, bQHat,blambdahat, bH_theta_hat, bSHat, bH_hat, bT_hat, bH, dTAU, Cost,
%
%
% for
%     %numeric
%     H = a1/(1+(y/a)^n)*(exp(y/a)-1)+ a2/(1+(x/a)^n)*(exp(x/b)-1);
%
% end
```