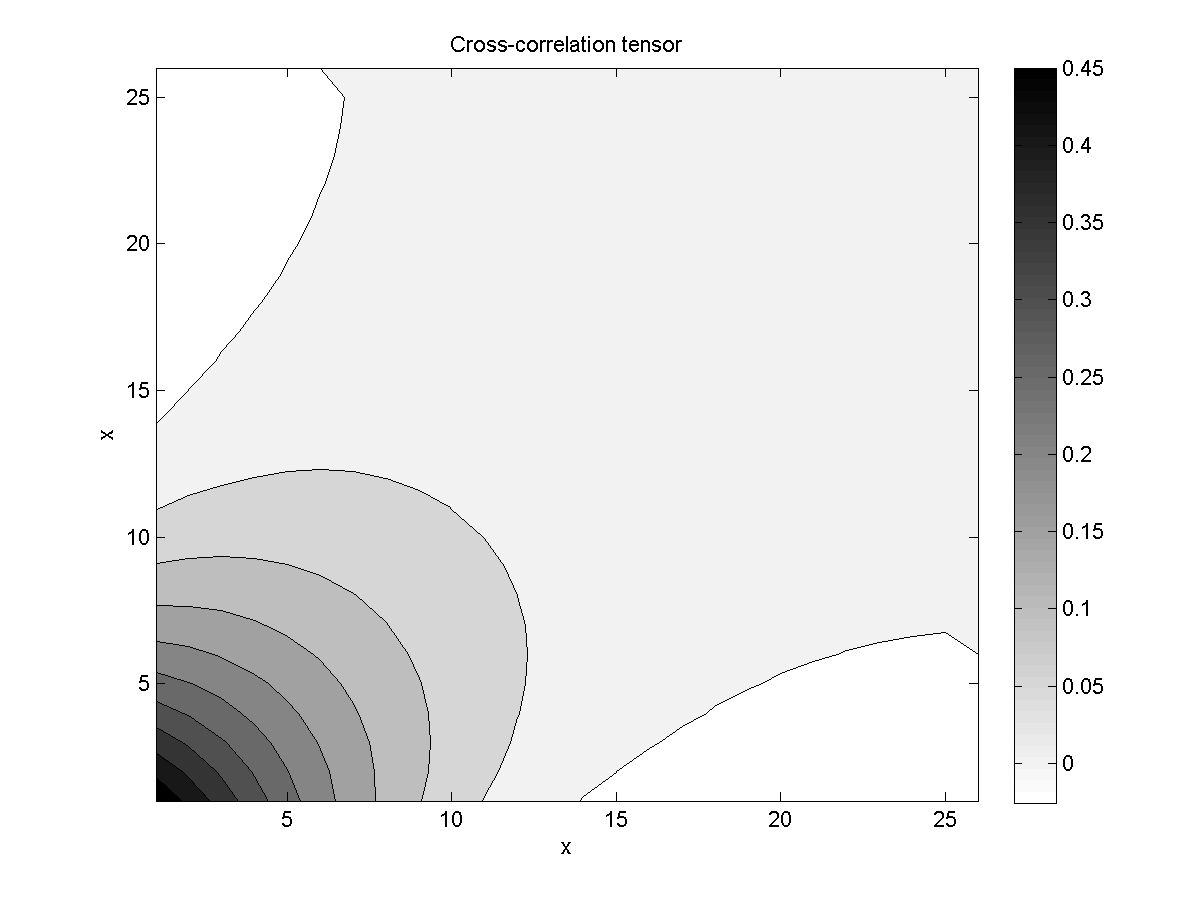
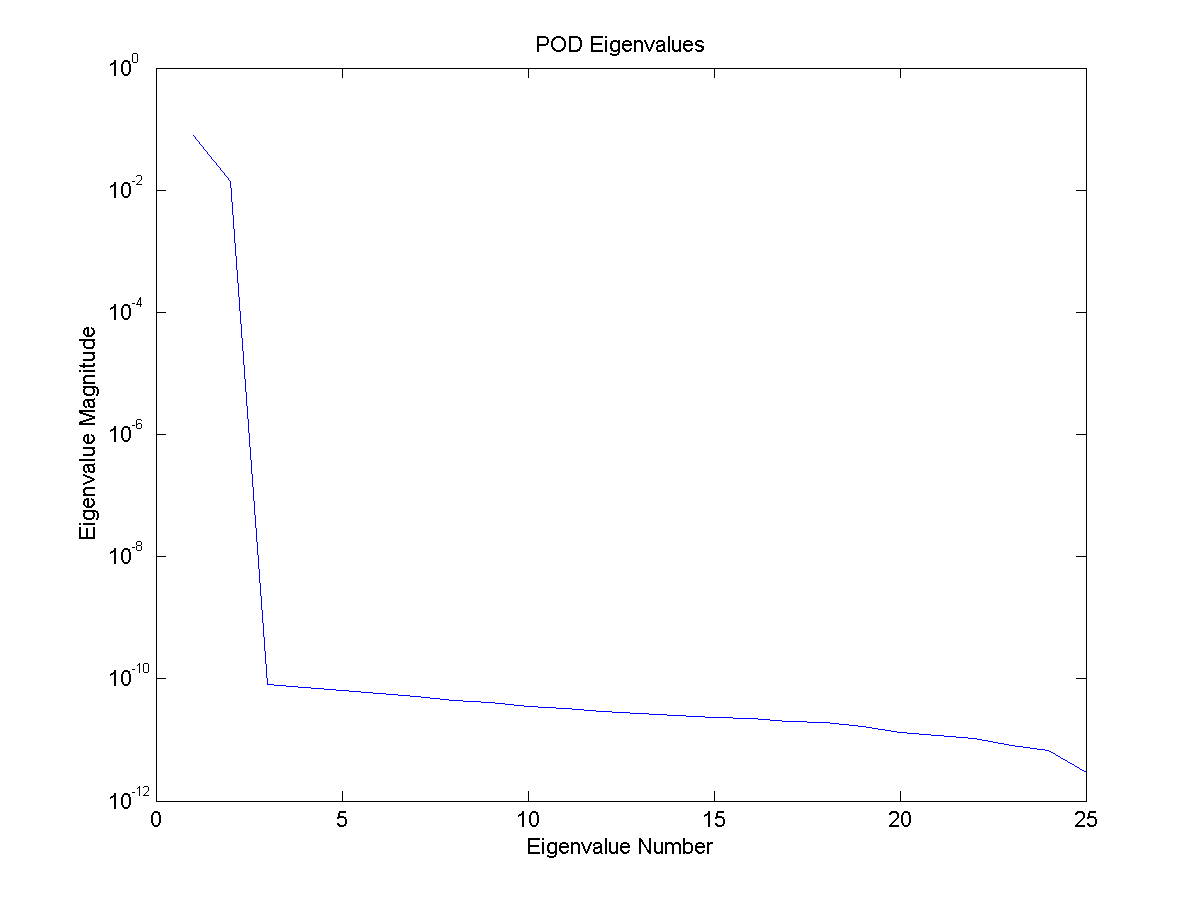
# Temperature Cross-correlation tensor

The below figure shows the contour plot of the two point cross-correlation tensor for the temperature distribution. The plot is symmetric about the diagonal, as one would expect for a one dimensional cross-correlation, with the correlation decreasing as x increases.

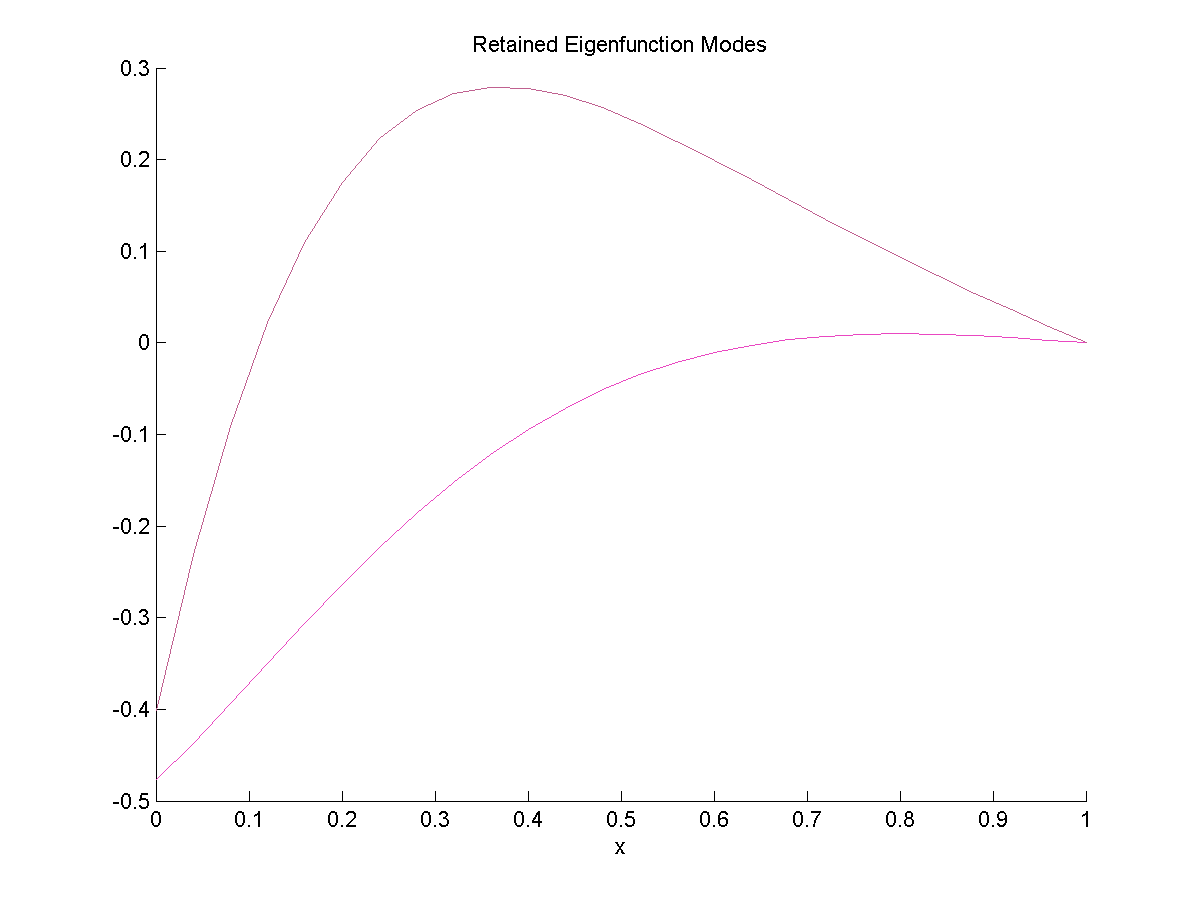


# Eigenvalues and Eigenfunctions

Below, the eigenvalues are plotted in descending order. Clearly, only the first two correspond to any significant energy in the problem; the rest of the eigenvalues appear to be due more to numerical truncation errors rather than any inherent oscillations in the temperature distribution.

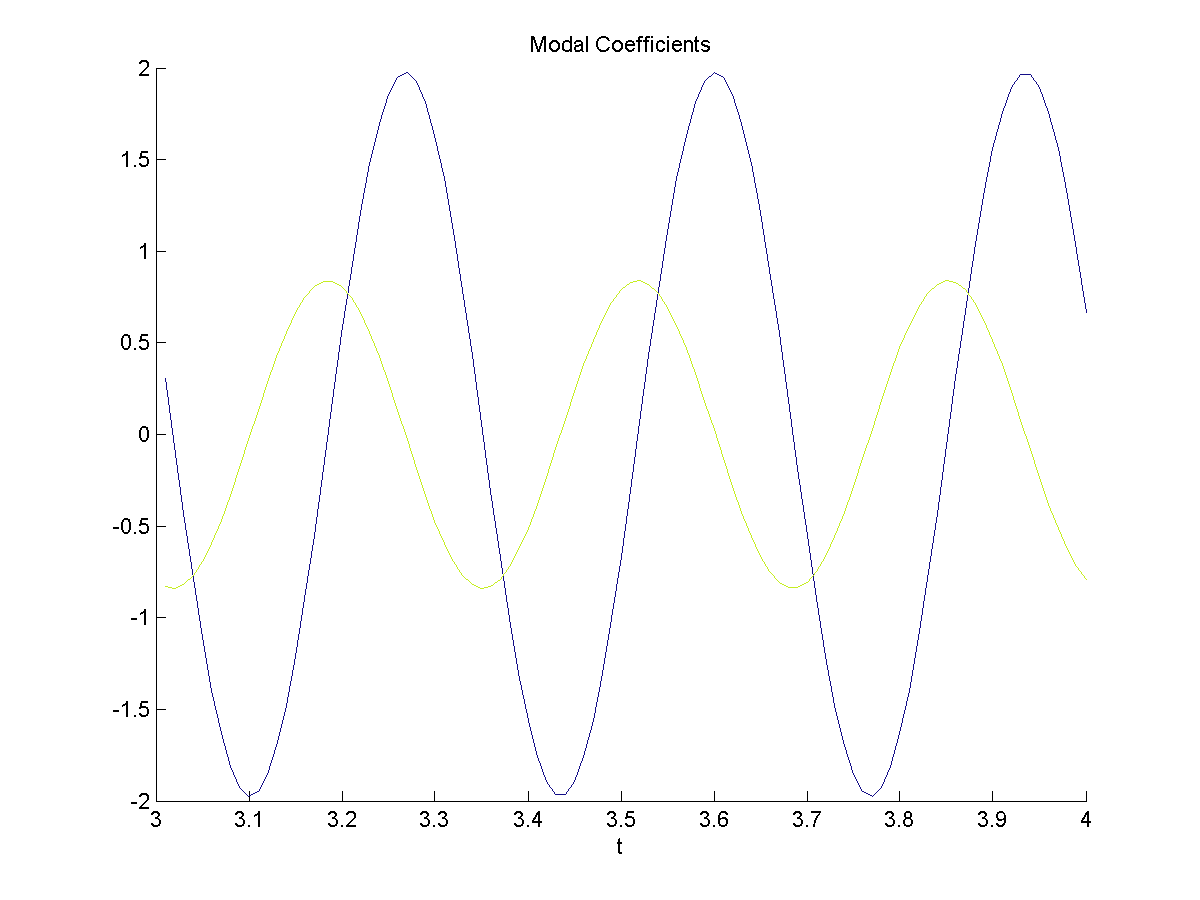


As one would expect from the preceding plot, only the first two POD modes are required to ensure that the reconstructed temperature distribution captures at least 99% of the energy. The retained eigenfunctions are plotted in the figure below. A simple calculation shows that the eigenfunctions are orthogonal (see ‘eigenfunction\_check’ in supplied m-file).

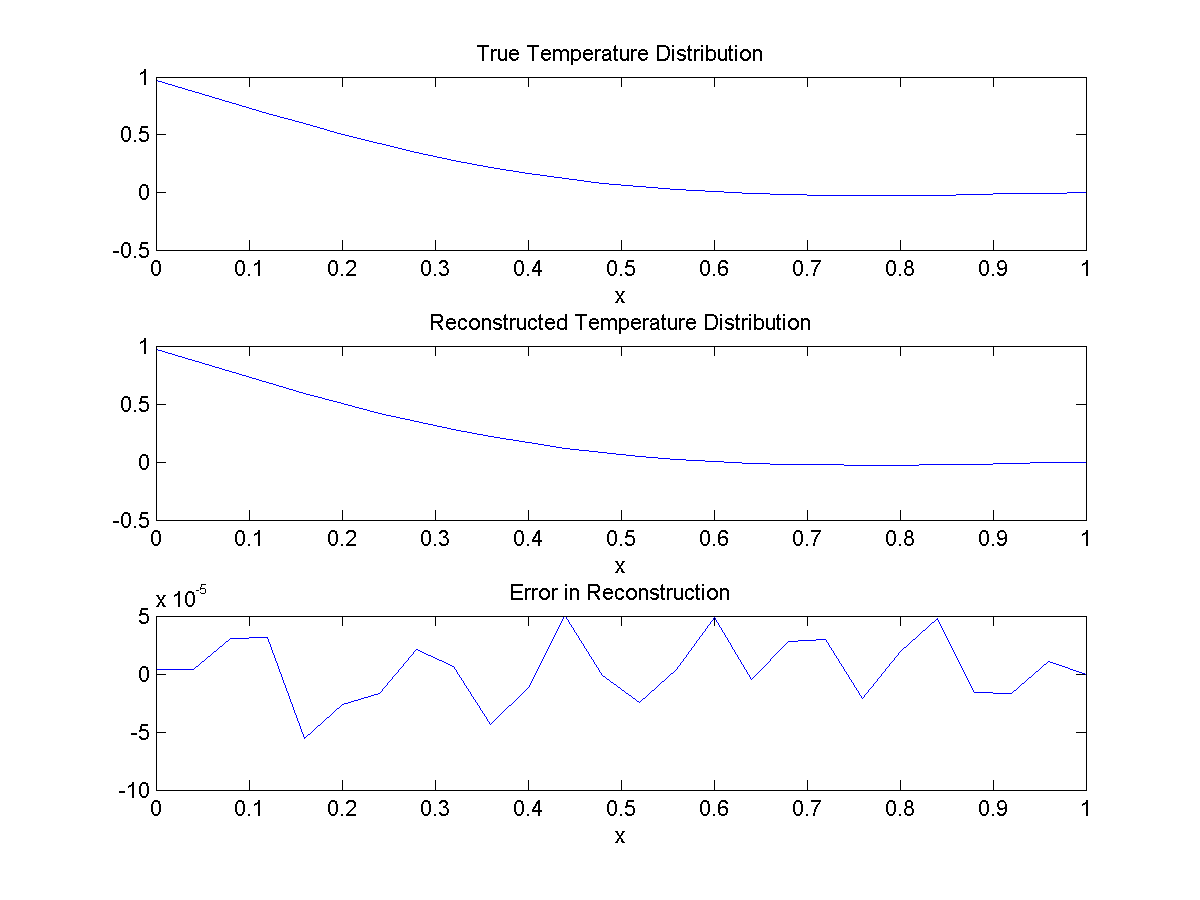


# POD Modal Coefficients

The retained POD modal coefficients are plotted in the figure below. A simple calculation verifies that the expected value of the coefficients squared is equal to the eigenvalues (see ‘lambda\_test’ in supplied m-file).

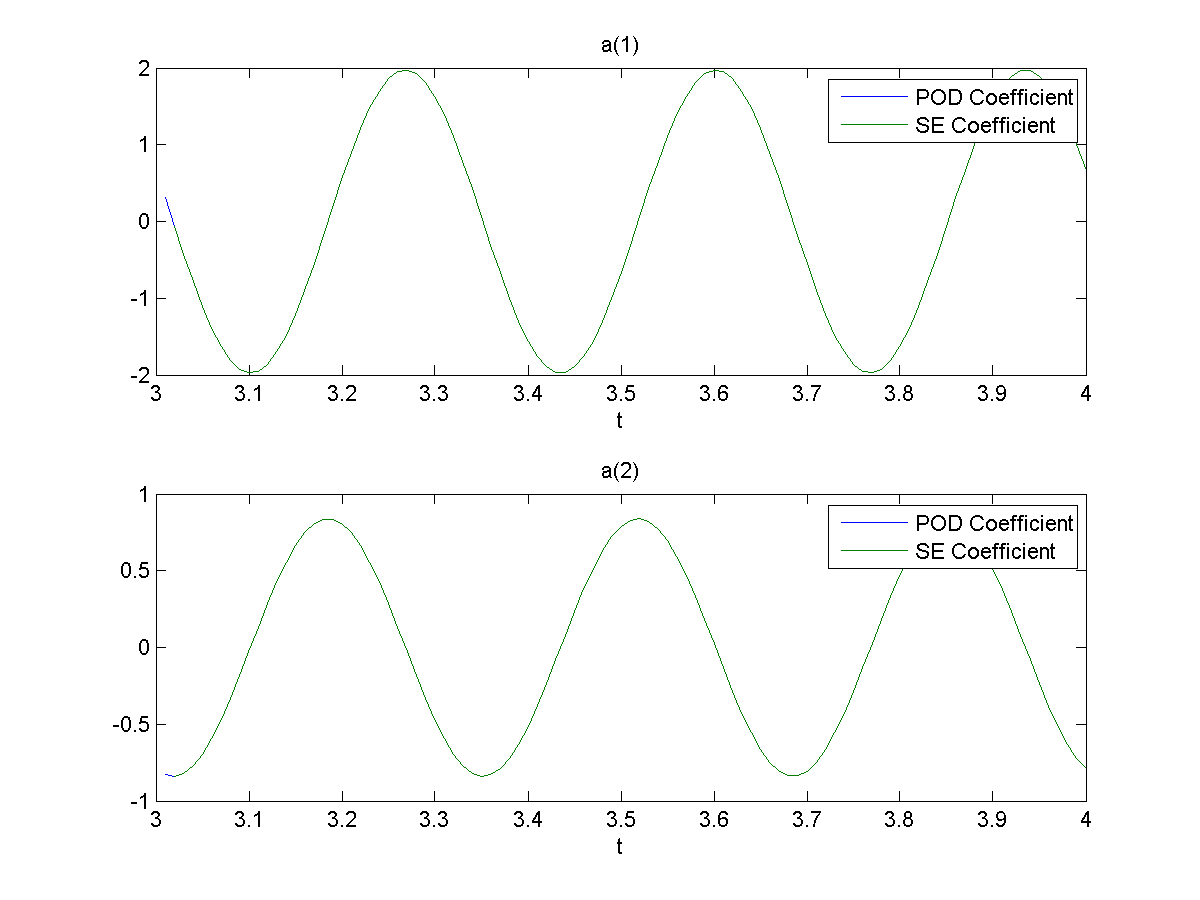


Using the retained POD modes, the temperature distribution along the bar for time 3.43 was reconstructed in the plot below. From this plot it is clear that the POD reconstruction is accurately capturing the behavior of the temperature distribution.



# Stochastic Estimation

Coefficients for linear dynamic stochastic estimation were calculated from the retained POD modes using the least-squares error framework. In the figure below, where the retained actual POD coefficients are plotted against the estimated coefficients, it is clear that the stochastic estimation method is accurately estimating the POD coefficients. The errors for each retained modal coefficient were found to be less than 1E-6.



# Appendix

function [B,phi,lambda,lambda\_test,modes,a,ac,Error,eigenfunction\_check] = HW4(data)

%Code completed by Michael Crawley for ME 813 HW#4

B = (data.u\*data.u')/length(data.t);figure;contourf(B);title('Cross-correlation tensor');xlabel('x');ylabel('x');

scale = 1/(length(data.x)-1);

[phi eigenvalue] = svd(scale\*B);

lambda = diag(eigenvalue);

EnormT = sum(scale\*sum(data.u.\*data.u,2))/length(data.t);

modes = find(cumsum(lambda) >= 0.99\*EnormT,1);

figure;hold on;title('Retained Eigenfunction Modes');xlabel('x');

for i = 1:modes

plot(data.x,phi(:,i),'Color',rand(3,1));

end

hold off;

figure;hold on;title('Modal Coefficients');xlabel('t');

a = zeros(modes,length(data.t));

lambda\_test = zeros(modes,1);

for i = 1:modes

a(i,:) = phi(:,i)'\*data.u;

lambda\_test(i,1) = mean(0.04\*a(i,:).\*a(i,:));

plot(data.t,a(i,:),'Color',rand(3,1));

end

hold off;

eigenfunction\_check = phi'\*phi;

eigenfunction\_check(eigenfunction\_check < 10\*eps) = 0;

I = find(abs(data.t - 3.43) < 10\*eps);

figure;

subplot(3,1,1);plot(data.x,data.u(:,I));title('True Temperature Distribution');xlabel('x');

subplot(3,1,2);plot(data.x,phi(:,1:modes)\*a(:,I));title('Reconstructed Temperature Distribution');xlabel('x');

subplot(3,1,3);plot(data.x,data.u(:,I)-phi(:,1:modes)\*a(:,I));title('Error in Reconstruction');xlabel('x');

X = [mean(data.u(1,2:end).^2) mean(data.u(1,2:end).\*data.u(1,1:end-1)); mean(data.u(1,2:end).\*data.u(1,1:end-1)) mean(data.u(1,1:end-1).^2)];

ac = zeros(modes,length(data.t)-1);

for i = 1:modes

Z = [mean(data.u(1,2:end).\*a(i,2:end)); mean(data.u(1,1:end-1).\*a(i,2:end))];

D = X\Z;

ac(i,:) = D(1)\*data.u(1,2:end)+D(2)\*data.u(1,1:end-1);

end

figure;

subplot(2,1,1);plot(data.t,a(1,:),data.t(2:end),ac(1,:));title('a(1)');xlabel('t');

subplot(2,1,2);plot(data.t,a(2,:),data.t(2:end),ac(2,:));title('a(2)');xlabel('t');

Error = mean((ac-a(:,2:end)).^2,2);

end