```
function ex8full(part) switch part case 1 ex8full1() case 2 ex8full2() end end function [A1,A2] = get_matrices() A1 = [-2 5 -3; 5 -6 6; -3 6 -6]; A2 = [-1 3 -1; 2 -3 5; -2 1 -3]; end function [emax, emin] = method_power (A, shift, maxiter) bmax = rand(length(A),1); bmin = rand(length(A),1); Ashift = (A-shift*eye(length(A)))^-1; emax = zeros(maxiter,1); emin = zeros(maxiter,1); for j=1:maxiter bmax = A*bmax/norm(A*bmax); bmin = Ashift*bmin/norm(Ashift*bmin); emax(j) = max((A*bmax)./bmax); emin(j) = min((A*bmin)./bmin); end end function ex8full1() clear all; close all; maxiter=100; [A1,A2] = get_matrices(); [e2max, e2min] = method_power (A2,i,maxiter); [e1max, e1min] = method_power (A1,0,maxiter); display('Eigenvalues of A1:'); display('Max: '); e1max(length(e1max)) display('Min: '); e1min(length(e1min)) display('MatLab result: '); eig(A1) display('Eigenvalues of A2:'); display('Max: '); e2max(length(e2max)) display('Min: '); e2min(length(e2min)) display('MatLab result: '); eig(A2) e1 = eig(A1); e2 = eig(A2); [tmp, pos1max] = max(abs(e1)); [tmp, pos1min] = min(abs(e1)); [tmp, pos2max] = max(abs(e2)); [tmp, pos2min] = min(abs(e2)); d1max = abs(e1max-e1(pos1max(1))); d1min = abs(e1min-e1(pos1min(1))); d2max = abs(e2max-e2(pos2max(1))); d2min = abs(e2min-e2(pos2min(1))); figure(1); semilogy(1:maxiter,d1max,'-rx'); hold on; semilogy(1:maxiter,d1min,'-ro'); semilogy(1:maxiter,d2max,'-bx'); semilogy(1:maxiter,d2min,'-bo'); legend('Maximal - A1', 'Minimal - A1', 'Maximal - A2', 'Minimal - A2'); title('Power method - finding the largest and smallest eigenvalue'); ylabel('Deviation from MatLab result'); xlabel('Iteration'); bmax=rand(length(A1),1); for iter=1:100 bmax=A1*bmax/norm(A1*bmax); end borth = rand(length(A1),1); borth(1)=-sum(borth(2:length(bmax)).*bmax(2:length(bmax)))/bmax(1); for iter=1:100 borth=A1*borth/norm(A1*borth); end display('After iterations, orthogonal vector became:') borth display('Original vector is:') bmax end function ex8full2() clear all; close all; [A1,A2]=get_matrices(); A=A1; evals=eig(A); maxiter=10; n1=norm(A,'fro'); for i=1:maxiter B=triu(abs(A),1); [p,q]=find(B==max(B(:))); if (A(p,q)~=0) tau = (A(q,q)-A(p,p))/(2.0*A(p,q)); if (tau>=0) t = 1/(tau+sqrt(1+tau^2)); else t = -1 / (-tau + sqrt(1+tau^2)); end c = 1/(sqrt(1+t^2)); s = t*c; else c = 1; s = 0; end G=eye(length(A)); G(p,p)=c; G(q,q)=c; G(p,q)=s; G(q,p)=-s; A = G' * A * G; n2=norm(A,'fro'); display(sprintf('Norm deviation at iteration %d: %f',i,(n2-n1)^2)); end sort(diag(A)) evals end
```