```
    % Makes right hand side of KKT equation
    bb = [-q + rho*(z - u); b];
    % Solves KKT equation
    xx = KK\bb;
    %  update x, z, u (ADMM update steps)
    x = xx(1:n);
    z = poslin(x + u);
    u = u + x - z;
    %  to test stopping criterion
    r = x - z;                    % primal residual
    nr = sqrt(r'*r);              % norm of primal residual
    s = rho*(z - z0);             % dual residual
    ns = sqrt(s'*s);              % norm of dual residual
end
end
```

The second program `SBVMhard2` implements hard margin SVM (version 2).

```
function [lamb,mu,w] = SVMhard2(rho,u,v)
%
%   Runs hard margin SVM version 2
%
%   p green vectors u_1, ..., u_p in n x p array u
%   q red   vectors v_1, ..., v_q in n x q array v
%
%   First builds the matrices for the dual program
%
p = size(u,2); q = size(v,2); n = size(u,1);
[A,c,X,Pa,qa] = buildhardSVM2(u,v);
%
%  Runs quadratic solver
%
tolr = 10^(-10); tols = 10^(-10); iternum = 80000;
[lam,U,nr,ns,kk] = qsolve1(Pa, qa, A, c, rho, tolr, tols, iternum);
fprintf('nr =  %d ',nr)
fprintf('   ns =  %d \n',ns)
fprintf('kk =  %d \n',kk)
if kk > iternum
   fprintf('** qsolve did not converge. Problem not solvable ** \n')
end
w = -X*lam;
nw = sqrt(w'*w);   % norm of w
fprintf('nw =  %.15f \n',nw)
```