```
   % num1
   % num2
   b =  (sy1/num1 + sy2/num2 - w'*(sx1/num1 + sx2/num2))/2;
   fprintf('b =  %.15f \n',b)
   epsilon =  (w'*(sx1/num1 - sx2/num2)  + sy2/num2 - sy1/num1)/2;
   fprintf('epsilon =  %.15f \n',epsilon)
   if epsilon < 10^(-10)
      fprintf('** Warning; epsilon is too small or negative ** \n')
   end
   nw = sqrt(w'*w);   % norm of w
   fprintf('nw =  %.15f \n',nw)
   %
   tolxi = 10^(-10);
   % tols < lambda_i < C/m - tolh or C/m - tolh <= lambda_i and xi_i < tolxi
   [lamsv,psf,xi] = findnuregsvl2(lamb,w,b,X,y,epsilon,C/m,tols,tolh,tolxi);
    % tols < mu_i < C/m - tolh or C/m - tolh <= mu_i and xi_i' < tolxi
   [musv,qsf,xip] = findnuregsvm2(mu,w,b,X,y,epsilon,C/m,tols,tolh,tolxi);
   fprintf('psf =  %d ',psf)
   fprintf('   qsf =  %d \n',qsf)
else
   fprintf('** Not enough support vectors ** \n')
end
end
```

To run `donuregb` use the function `runuregb` listed below.

```
function [lamb,mu,alpha,beta,lambnz,munz,lamK,muK,w] = runuregb (rho,nu,X,y,C)
%
%   Runs soft margin nu-regression
%   with the constraint
%   \sum_{i = 1}^m + \sum_{j = 1}^m mu_j = C nu
%  (Without the variable gamma)
%
%  Input: an m x n matrix of data points represented as
%  as the rows of X, and y a vector in R^n
%
%   First builds the matrices for the dual program
%    C is a scale factor
%

m = size(X,1); n = size(X,2);
[lamb,mu,alpha,beta,lambnz,munz,lamK,muK,numsvl1,numsvm1,w,epsilon,b]
      = donuregb(rho,nu,X,y,C);
```