Annex B. *Matlab* Code for the Analysis of Tensile Reinforced Concrete Elements

The following code, starting below this paragraph, enables the application of the strain compliance approach for tensile RC elements. The code includes two definitions of tension stiffening laws and the Eurocode 2 implementation for the mean strain approach. The procedure follows the concept proposed in Chapter 2 of the dissertation. The same code can be applied to cases when debonding is considered or neglected, as well as accounting for shrinkage and cases when it is not considered. In addition, the code obtains all the key results for all defined variable cases. The percentage sign marks comment lines that are not executed.

```
clear all
close force all
clc

% Begins counting cpu runtime
runtime = tic;

% Diameter of reference element
diam_ref = 14; % [mm]

% Reinforcement ratio of reference element
ro_ref   = 1.54; % [%]

% Crack spacing of reference element
srm_ref = 162.4; % [mm]

% Material properties of investigated element
fcm_ref = 30; % [MPa]
fs_ref  = 500; % [MPa]
Es_ref  = 200; % [MPa]
Ec_ref  = 21.5*1*(fcm_ref*0.1)^(1/3);
fct_ref = 0.3*(fcm_ref)^(2/3); % [GPa]
if fct_ref > 50
	fct_ref = 2.12 * log(1+0.1*(fcm_ref+8)); % [MPa]
end

% Assumed shrinkage strain
es_sh = 2.0e-4;
% es_sh = 0;

% Loading conditions
```

```
Fu = fs_ref * (pi()*(diam_ref^2)*0.25) *1e-3; % [kN]
F_ref = 0.6*Fu; % [kN]

% Debonding zone coefficient
cof = 1/3; % With debonding
% cof = 0; % No debonding

% List of investigated parameters for alternative elements
diam_list = [10 12 14 16 20 25]'; % [mm]
ro_list   = [0.78 1 1.13 1.25 1.54 2.0 2.5 3.0 3.14 3.5... 4]'; % [%]
srm_list  = [0 216.2 182.5 161.2 149.9 137.6 0]'; % [mm]

% Scales data to required units
diam_ref = diam_ref *1e-3;
srm_ref  = srm_ref  *1e-3;
fcm_ref  = fcm_ref  *1e6;
fs_ref   = fs_ref   *1e6;
Es_ref   = Es_ref   *1e9;
fct_ref  = fct_ref  *1e6;
Ec_ref   = Ec_ref   *1e9;
F_ref    = F_ref    *1e3;

% Area of concrete and reinforcement
As_ref = pi()*(diam_ref^2)*0.25; % [mm^2]
Ac_ref = As_ref / (ro_ref*0.01); % [mm^2]

% Cracking load
Fcr = (Ac_ref * fct_ref * (1 + ((Es_ref*As_ref)/(Ec_ref*Ac_ref)))) *1e-3; % [kN]

% Shrinkage force and strain
N_sh = es_sh.*Es_ref.*As_ref;
es_sh1 = es_sh.*Es_ref.*As_ref ./ (Ec_ref.*Ac_ref +... Es_ref.*As_ref);

% Defines and estimates results neccesary for tension stiffening models
fun_stiff_1 = @(fcm,esm) 3 - (12.5/(fcm.^0.5)) +... 1.76*exp(-esm);
fun_stiff_2 = @(fcm,esm) 0.025*fcm-...((0.85.*(esm.^0.8)-1.5)./(0.25*(esm.^0.3)+0.8));

stiff_tb(:,1) = (0:1e-6:2.5e-3);
stiff_tb(:,2) = fun_stiff_1(fcm_ref*1e-6,... stiff_tb(:,1)*1000);
stiff_tb(:,3) = fun_stiff_2(fcm_ref*1e-6,... stiff_tb(:,1)*1000);
stiff_tb(:,4) = Es_ref.*As_ref.*stiff_tb(:,1);

stiff_tb(:,5) = stiff_tb(:,2)*1e6.*Ac_ref;
stiff_tb(:,6) = stiff_tb(:,4) + stiff_tb(:,5);
```

```
stiff_tb(:,7) = stiff_tb(:,6) ./ (Es_ref.*As_ref +... Ec_ref.*Ac_ref);

stiff_tb(:,8) = stiff_tb(:,3)*1e6.*Ac_ref;
stiff_tb(:,9) = stiff_tb(:,4) + stiff_tb(:,8);
stiff_tb(:,10) = stiff_tb(:,9) ./ (Es_ref.*As_ref +... Ec_ref.*Ac_ref);

% First tension stiffening model results
stiff_tb(:,11) = stiff_tb(:,1) - es_sh1;
stiff_tb(:,12) = stiff_tb(:,6) - N_sh;
stiff_tb(:,13) = stiff_tb(:,4) - N_sh;

% Second tension stiffening model results
stiff_tb(:,14) = stiff_tb(:,1) - es_sh1;
stiff_tb(:,15) = stiff_tb(:,9) - N_sh;
stiff_tb(:,16) = stiff_tb(:,4) - N_sh;

%% Reference Element Analysis - Stiffening Model 1
% Reinforcement and concrete strains at location of crack
es_1(3) = interp1(stiff_tb(:,13),stiff_tb(:,11),F_ref);
ec_1(3) = 0;
esi = es_1(3);

% Debonding zone and effective zone length
ld_ref_1 = esi*diam_ref*1000*cof;
leff_1 = 0.5*(srm_ref - 2*ld_ref_1);

% Mean reinforcement and concrete strains
es_1(2) = interp1(stiff_tb(:,12),stiff_tb(:,11),F_ref);
ec_1(2) = (F_ref-es_1(2).*Es_ref.*As_ref)./(Ec_ref.*Ac_ref);
esm = es_1(2);

% Slope coefficient for linear shape function
A_1 = (esi.*leff_1 + esi.*ld_ref_1 - 0.5*esm.*srm_ref)... ./ (0.5*leff_1.*leff_1);

% Minimum reinforcement and concrete strain values at middle of RC block
es0_1 = esi - A_1.*0.5*(srm_ref-2*ld_ref_1);
es_1(1) = es0_1;
ec_1(1) = (F_ref-es_1(1).*Es_ref.*As_ref)./(Ec_ref.*Ac_ref);

% Estimates bond stress and ratio to fct
Tau_ref_1 = A_1.*Es_ref.*diam_ref*0.25;
Tau_ref_fact_1 = Tau_ref_1 ./ fct_ref;

% Stores results in separate array for easier comparison
```

```
ref_res(1:3,1) = es_1;
ref_res(4:6,1) = ec_1;
ref_res(7,1) = ld_ref_1;
ref_res(8,1) = leff_1;
ref_res(9,1) = A_1;
ref_res(10,1) = Tau_ref_1;
ref_res(11,1) = Tau_ref_fact_1;

% Data preparation for crack spacing analysis of alternative elements
[diam_mesh,ro_mesh] = meshgrid(diam_list,ro_list);

% Creates list of all reinforcement and concrete areas from variable list
As_list = (pi()*0.25*diam_mesh.^2) * 1e-6;
Ac_list = As_list ./ (ro_mesh*0.01);

% Creates list of cracking and ultimate loads from variable list
Fcr_list = (Ac_list .* fct_ref .* (1 +... ((Es_ref.*As_list)./(Ec_ref.*Ac_list)))) *1e-3; %
[kN]
Fu_list = [fs_ref .* (pi()*((diam_list*0.001).^2)*0.25)... *1e-3]'; % [kN]

% Creates list of all shrinkage forces and strains
N_sh_list = es_sh.*Es_ref.*As_list;
es_sh_list = es_sh.*Es_ref.*As_ref ./ (Ec_ref.*Ac_list... + Es_ref.*As_list);

% Calculates slope coefficient values for alternative elements
for j = 1:1:numel(ro_mesh(1,:))

   A_mesh(1,j) = Tau_ref_1 * 4 ./ (Es_ref*diam_mesh(ro_mesh(:,1) == ro_ref,j)*1e-3);

end

% Calculations of alternative elements
for i = 1:1:numel(ro_mesh(1,:))

   for j = 1:1:numel(ro_mesh(:,1))

      clear stiff_plotn
      stiff_plotn(:,1) = [0:1e-6:3.5*1e-3];
      stiff_plotn(:,2) = fun_stiff_1(fcm_ref*...      1e-6,stiff_plotn(:,1)*1000);
      % Force F
      stiff_plotn(:,4) = Es_ref.*As_list(j,i).*stiff_plotn(:,1);
      % Axial Concrete F
      stiff_plotn(:,5) = stiff_plotn(:,2)*1e6.*Ac_list(j,i);
      stiff_plotn(:,6) = stiff_plotn(:,4)+stiff_plotn(:,5);
```

```
        stiff_plotn(:,7) = stiff_plotn(:,6) ./... (Es_ref.*As_list(j,i) + Ec_ref.*Ac_list(j,i));

        stiff_plotn(:,11) = stiff_plotn(:,1) -... es_sh_list(j,i);
        stiff_plotn(:,12) = stiff_plotn(:,6) -... N_sh_list(j,i);
        stiff_plotn(:,13) = stiff_plotn(:,4) -... N_sh_list(j,i);

        iter_leff = (stiff_plotn(:,11) - es0_1) ./... A_mesh(i);
        iter_ldx = stiff_plotn(:,11).*diam_mesh(j,i)*cof;
        iter_con =... ((es0_1+stiff_plotn(:,11)).*0.5.*iter_leff +...
stiff_plotn(:,11).*iter_ldx)./(iter_leff + iter_ldx);

        [mean_con,F_max] =... intersections(stiff_plotn(:,11),stiff_plotn(:,12),...
iter_con,stiff_plotn(:,13),'ROBUST');

        mean_con = mean_con(end);
        F_max = F_max(end);

        iter_esi =... interp1(stiff_plotn(:,13),stiff_plotn(:,11),F_max);
        iter_esi = iter_esi(end);
        ldx = iter_esi.*diam_mesh(j,i)*cof;
        leff = (iter_esi - es0_1) / A_mesh(i);

        max_es(j,i) = iter_esi;
        mean_es(j,i) = mean_con;
        sigmas_mesh(j,i) = iter_esi*Es_ref*1e-6;
        F_mesh(j,i) = F_max;

        % Stores zone and cracking length results
        ld_mesh(j,i) = ldx;
        leff_mesh(j,i) = leff;
        srm_mesh(j,i) = (leff+ldx)*2;

        F_res(j,i) = iter_esi * Es_ref* As_list(j,i)*1e-3;
    end
end

% Stores results for stiffening model 1 based predictions
model1_srm = srm_mesh *1000;
model1_ld = ld_mesh *1000;
model1_leff = leff_mesh *1000;
model1_esm = mean_es;
model1_esmax = max_es;
model1_F = F_res;
model1_srm_comp = model1_srm([1 14 27 39 53]);
```

```
model1_srm_comp(2,:) = model1_srm_comp./srm_list(2:6)';
clear max_es mean_es sigmas_mesh F_mesh ld_mesh... leff_mesh srm_mesh F_res
A_mesh


%% Reference Element Analysis - Stiffening Model 2
% Reinforcement and concrete strains at location of crack
es_2(3) = interp1(stiff_tb(:,16),stiff_tb(:,14),F_ref);
ec_2(3) = 0;
esi = es_2(3);

% Debonding zone and effective zone length
ld_ref_2 = esi*diam_ref*1000*cof;
leff_2 = 0.5*(srm_ref - 2*ld_ref_2);

% Mean reinforcement and concrete strains
es_2(2) = interp1(stiff_tb(:,15),stiff_tb(:,14),F_ref);
ec_2(2) = (F_ref-es_2(2).*Es_ref.*As_ref)./(Ec_ref.*Ac_ref);
esm = es_2(2);

% Slope coefficient for linear shape function
A_2 = (esi.*leff_2 + esi.*ld_ref_2 - 0.5*esm.*srm_ref)... ./ (0.5*leff_2.*leff_2);

% Minimum reinforcement and concrete strain values at middle of RC block
es0_2 = esi - A_2.*0.5*(srm_ref-2*ld_ref_2);
es_2(1) = es0_2;
ec_2(1) = (F_ref-es_2(1).*Es_ref.*As_ref)./(Ec_ref.*Ac_ref);

% Estimates bond stress and ratio to fct
Tau_ref_2 = A_2.*Es_ref.*diam_ref*0.25;
Tau_ref_fact_2 = Tau_ref_2 ./ fct_ref;

% Stores results in separate array for easier comparison
ref_res(1:3,2) = es_2;
ref_res(4:6,2) = ec_2;
ref_res(7,2) = ld_ref_2;
ref_res(8,2) = leff_2;
ref_res(9,2) = A_2;
ref_res(10,2) = Tau_ref_2;
ref_res(11,2) = Tau_ref_fact_2;

% Calculates slope coefficient values for alternative elements
for j = 1:1:numel(ro_mesh(1,:))
```

```
    A_mesh(1,j) = Tau_ref_2 * 4 ./... (Es_ref*diam_mesh(ro_mesh(:,1) == ro_ref,j)*1e-3);

end

% Calculations of alternative elements
for i = 1:1:numel(ro_mesh(1,:))

    for j = 1:1:numel(ro_mesh(:,1))

        clear stiff_plotn
        stiff_plotn(:,1) = [0:1e-6:3.5*1e-3];
        stiff_plotn(:,2) = fun_stiff_2(fcm_ref*1e-6,stiff_plotn(:,1)*1000);
        % Force F
        stiff_plotn(:,4) =... Es_ref.*As_list(j,i).*stiff_plotn(:,1);
        % Axial Concrete F
        stiff_plotn(:,5) =... stiff_plotn(:,2)*1e6.*Ac_list(j,i);
        stiff_plotn(:,6) =... stiff_plotn(:,4)+stiff_plotn(:,5);
        stiff_plotn(:,7) = stiff_plotn(:,6) ./... (Es_ref.*As_list(j,i) + Ec_ref.*Ac_list(j,i));

        stiff_plotn(:,11) = stiff_plotn(:,1) -... es_sh_list(j,i);
        stiff_plotn(:,12) = stiff_plotn(:,6) -... N_sh_list(j,i);
        stiff_plotn(:,13) = stiff_plotn(:,4) -... N_sh_list(j,i);

        iter_leff = (stiff_plotn(:,11) - es0_2) ./... A_mesh(i);
        iter_ldx = stiff_plotn(:,11).*diam_mesh(j,i)*cof;

        iter_con=((es0_2+stiff_plotn(:,11)).*0.5.*iter_leff+stiff_plotn(:,11).*iter_ldx)./…(it
        er_leff + iter_ldx);

        [mean_con,F_max]=intersections(stiff_plotn(:,11),stiff_plotn(:,12),...
        iter_con,stiff_plotn(:,13),'ROBUST');

        mean_con = mean_con(end);
        F_max = F_max(end);

        iter_esi =... interp1(stiff_plotn(:,13),stiff_plotn(:,11),F_max);
        iter_esi = iter_esi(end);
        ldx = iter_esi.*diam_mesh(j,i)*cof;
        leff = (iter_esi - es0_2) / A_mesh(i);

        max_es(j,i) = iter_esi;
        mean_es(j,i) = mean_con;
        sigmas_mesh(j,i) = iter_esi*Es_ref*1e-6;
        F_mesh(j,i) = F_max;
```

```
        % Stores zone and cracking length results
        ld_mesh(j,i) = ldx;
        leff_mesh(j,i) = leff;
        srm_mesh(j,i) = (leff+ldx)*2;

        F_res(j,i) = iter_esi * Es_ref* As_list(j,i)*1e-3;
    end
end

% Stores results for stiffening model 2 based predictions
model2_srm = srm_mesh *1000;
model2_ld = ld_mesh *1000;
model2_leff = leff_mesh *1000;
model2_esm = mean_es;
model2_esmax = max_es;
model2_F = F_res;
model2_srm_comp = model2_srm([1 14 27 39 53]);
model2_srm_comp(2,:) = model2_srm_comp./srm_list(2:6)';

clear max_es mean_es sigmas_mesh F_mesh ld_mesh leff_mesh srm_mesh F_res


%% Reference Element Analysis - Eurocode 2
% Reinforcement and concrete strains at location of crack
es_3(3) = F_ref./(Es_ref.*As_ref);
ec_3(3) = 0;
esi = es_3(3);

% Debonding zone and effective zone length
ld_ref_3 = esi*diam_ref*1000*cof;
leff_3 = 0.5*(srm_ref - 2*ld_ref_3);

% Elastic strain for EC2
es_el_ref_3 = F_ref./(Ec_ref.*Ac_ref+Es_ref.*As_ref);  % [] Elastic strains

% Interpolation coefficient (stiffening coefficient)
xi = 1-(Fcr*1000./F_ref)^2;

% Mean reinforcement and concrete strains
es_3(2) = (1-xi)*es_el_ref_3 + xi*es_3(3);
ec_3(2) = (F_ref-es_3(2).*Es_ref.*As_ref)./(Ec_ref.*Ac_ref);
esm = es_3(2);
```

```
% Slope coefficient for linear shape function
A_3 = (esi.*leff_3 + esi.*ld_ref_3 - 0.5*esm.*srm_ref)... ./ (0.5*leff_3.*leff_3);

% Minimum reinforcement and concrete strain values at middle of RC block
es0_3 = esi - A_3.*0.5*(srm_ref-2*ld_ref_3);
es_3(1) = es0_3;
ec_3(1) = (F_ref-es_3(1).*Es_ref.*As_ref)./(Ec_ref.*Ac_ref);

% Estimates bond stress and ratio to fct
Tau_ref_3 = A_3.*Es_ref.*diam_ref*0.25;
Tau_ref_fact_3 = Tau_ref_3 ./ fct_ref;

% Stores results in separate array for easier comparison
ref_res(1:3,3) = es_3;
ref_res(4:6,3) = ec_3;
ref_res(7,3) = ld_ref_3;
ref_res(8,3) = leff_3;
ref_res(9,3) = A_3;
ref_res(10,3) = Tau_ref_3;
ref_res(11,3) = Tau_ref_fact_3;

% Calculates slope coefficient values for alternative elements
for j = 1:1:numel(ro_mesh(1,:))

   A_mesh(1,j) = Tau_ref_3 * 4 ./... (Es_ref*diam_mesh(ro_mesh(:,1) == ro_ref,j)*1e-3);

end

for i = 1:1:numel(ro_mesh(1,:))

   for j = 1:1:numel(ro_mesh(:,1))

      con = 1;
      ct = 0;

      while abs(con) > 1e-9

         ct = ct + 1;

%          Fcr1 = (Ac_list(j,i) * fct_ref * (1 + ((Es_ref*As_list(j,i))/(Ec_ref*Ac_list(j,i)))));
% [kN]
%           Fu1 = fs_ref * As_list(j,i);

         if ct == 1
```

```
            es1max(ct,1) =... (Fu_list(i)*1000/(Es_ref*As_list(j,i)));
        elseif ct == 2
            es1max(ct,1) =... (Fcr_list(j,i)*1000/(Es_ref*As_list(j,i)));
        else
            es1max(ct,1) = es1max(ct-1,1) -...
            con_list(ct-1,1)*((es1max(ct-1,1)-es1max(...
            ct-2,1))/(con_list(ct-1,1)-con_list(ct-2,1)));
            if es1max(ct,1) < 0
                es1max(ct,1) = es1max(ct-1,1)*0.5;
            end
        end

        eps_el1 = …
        (es1max(ct,1)*Es_ref*As_list(j,i))/(Ec_ref*Ac_list(j,i)+...Es_ref*As_list(j,i))
        % [] Elastic strains
        xi1 = 1-(Fcr_list(j,i)*1000/... (es1max(ct,1)*Es_ref*As_list(j,i)))^2;
        es1mean = (1-xi1)*eps_el1 + xi1*es1max(ct,1);

        leff = (es1max(ct,1) - es0_3) / A_mesh(i);

        ldx = es1max(ct,1)*diam_mesh(j,i)*cof;

        mean_con = ((es0_3+es1max(ct,1))*0.5*leff +... es1max(ct,1)*ldx)/(leff + ldx);

        con_list(ct,1) = es1mean - mean_con;
        con = con_list(ct,1);

    end

    max_es(j,i) = es1max(ct,1);
    mean_es(j,i) = es1mean;
    sigmas_mesh(j,i) = es1max(ct,1)*Es_ref*1e-6;
    F_mesh(j,i) = es1max(ct,1)*Es_ref*As_list(j,i)*1e-3;

    ld_mesh(j,i) = ldx;
    leff_mesh(j,i) = leff;
    srm_mesh(j,i) = (leff+ldx)*2;

    F_res(j,i) = es1max(ct,1) * Es_ref*... As_list(j,i)*1e-3;
  end
end

% Stores results for Eurocode 2 based predictions
model3_srm = srm_mesh *1000;
```

```
model3_ld = ld_mesh *1000;
model3_leff = leff_mesh *1000;
model3_esm = mean_es;
model3_esmax = max_es;
model3_F = F_res;
model3_srm_comp = model3_srm([1 14 27 39 53]);
model3_srm_comp(2,:) = model3_srm_comp./srm_list(2:6)';

clear max_es mean_es sigmas_mesh F_mesh ld_mesh... leff_mesh srm_mesh F_res
```