Annex D. *Matlab* Code of Flexural Reinforced Concrete Element Analysis

D.1. *Matlab* script prerequisites

This section describes the prerequisite table construct in *Matlab* in order to run the appended *Matlab* codes from sections D.2 and D.3 A *Matlab* table construct named *flex_rawdata*, which then has to be saved as a *phd_flex_workspace.mat* file. The structure of the flex_rawdata must be expressed in the following manner, rows define specimens and columns are designated as follows:

Table D.1. *flex_rawdata Matlab* table structure

Column number	Column name
1	Collumn number
2	Data Source
3	Specimen No. from source
4	Section height h, m
5	Effective height d, m
6	Section width b, m
7	Compressive reinforcement cover a_{s2}, m
8	Tensile reinforcement cover c, m
9	Number of tensile bars in layer 1
10	Number of tensile bars in layer 2
11	Diameter of tensile bars in layer 1, $Ø_{s1\_1}$, mm
12	Diameter of tensile bars in layer 2, $Ø_{s1\_2}$, mm
13	Weighted average tensile bar diameter, $Ø_{s1}$, mm
14	Tensile reinforcement area A_{s1}, m^2
15	Number of compressive bars
16	Diameter of compressive bars $Ø_{s2}$, mm
17	Compressive reinforcement area A_{s2}, m^2
18	Reinforcement ratio ρ, %
19	Ratio of tensile reinforcement elasticity modulus to concrete n_{s1}
20	Ratio of compressive reinforcement elasticity modulus to concrete n_{s2}
21	Concrete elasticity modulus E_c, MPa
22	Tensile reinforcement elasticity modulus E_{s1}, MPa
23	Compressive reinforcement elasticity modulus E_{s2}, MPa
24	Concrete strength f_{cm}, MPa
25	Experimental mean crack spacing s_{rm}, m

D.2. Matlab code of the derivation of the central zone l_c model

The following code, starting from just below this paragraph, is intended to be applied on the 23 specimens as discussed in Chapter 3 of this PhD thesis. In order to replicate the results, it is crucial, that the initial data is the same order as presented in this dissertation. In order to apply to other data, the table construct in *Matlab* can be filled up in any order, provided the columns represent the same data as given in section D.1.

```
% Cleanup of Matlab workspace
close force all
clear all
clc

% Loads data file
load('phd_flex_workspace')

%% Definition of initial variables and parameters
% Experimental mean crack spacing in [m]
srm_exp = table2array(flex_rawdata(:,25));

% Concrete compressive strength from data in [MPa]
fcm = table2array(flex_rawdata(:,24));

% Calculates tensile strength of concrete in [MPa]
if fcm > 50
   fct = 2.12 *log(1+(fcm/10));
else
   fct = 0.3 * (fcm - 8).^(2/3);
end

% Elasticity modulus of tensile reinforcement in [MPa]
Es1 = table2array(flex_rawdata(:,22));

% Elasticity modulus of compressive reinforcement in [MPa]
Es2 = table2array(flex_rawdata(:,23));

% Elasticity modulus of concrete in [MPa]
Ec = 22000 * (fcm/10).^0.3;

% Equivalent diameter of all tensile rebars in [mm]
diam_s1 = table2array(flex_rawdata(:,13));
```

```
% Equivalent diameter of all compressive rebars in [mm]
diam_s2 = table2array(flex_rawdata(:,16));

% Number of tensile rebars
No1 = table2array(flex_rawdata(:,9)) + table2array(flex_rawdata(:,10));

% Number of compressive rebars
No2 = table2array(flex_rawdata(:,15));

% Reinforcement area of tensile rebars in [m2]
As1 = table2array(flex_rawdata(:,14));

% Reinforcement area of compressive rebars in [m2]
As2 = table2array(flex_rawdata(:,17));

% Section height in [m]
h = table2array(flex_rawdata(:,4));

% Section width in [m]
b = table2array(flex_rawdata(:,6));

% Effective section depth in [m]
d = table2array(flex_rawdata(:,5));

% Tensile rebar concrete cover in [m]
c = table2array(flex_rawdata(:,8));

% Compressive rebar concrete cover in [m]
as2 = table2array(flex_rawdata(:,7));

% Reinforcement ratio of elements in [decimal fraction]
ro = As1 ./ (d.*b);

% Bond stress in [MPa]
Tau = 2*fct;

%% Geometrical properties and related parameters
% Transformed area in [m2]
Atr = h .* b + ((Es1./Ec)-1).*As1 + ((Es2./Ec)-1).*As2;
```

```
% Location of neutral axis in [m]
y0 = (((As1.*(Es1./Ec)+As2.*((Es2./Ec)-1))./b).^2 ...
+ 2*(As1.*(Es1./Ec).*d + As2.*((Es2./Ec)-1).*as2)…
./b).^0.5- (As1.*(Es1./Ec)+As2.*((Es2./Ec)-1))./b;


% Transformed cracked area in [m2]
Atr_cr = y0.*b + As1.*(Es1./Ec) + As2.*((Es2./Ec)-1);

% Transformed First moment of area in [m3]
Str = 0.5*(b.*h.^2) + As1.*((Es1./Ec)-1).*d + As2.*((Es2./Ec)-1).*as2;

% Transformed First moment of area of cracked section in [m3]
Str_cr = (y0.^2).*b*0.5 + As1.*(Es1./Ec).*d + As2.*((Es2./Ec)-1).*as2;

% Elastic neutral axis in [m]
yel = Str ./ Atr;

% Transformed Second moment of area in [m4]
Itr = ((b.*h.^3)/12) + (h.*b.*(yel-0.5*h).^2) +…
   (As1.*(Es1./Ec).*(d-yel).^2)...
   + (As2.*((Es2./Ec)-1).*(yel-as2).^2);

% Transformed Second moment of area of cracked section in
% [m4]
Itr_cr = ((b.*y0.^3)/12) + (y0.*b.*(y0-0.5*y0).^2) +...
   As1.*(Es1./Ec).*(d-y0).^2) + (As2.*((Es2./Ec)-1).*(y0-
   as2).^2);

% Flexural stiffness in [MNm2]
EI = Itr.*Ec;

% Flexural stiffness of cracked section in [MNm2]
EI_cr = Itr_cr.*Ec;

%% Loading condition calculations
% Assumed maximum reinforcement strain in crack
esi = repmat(0.0015,size(h,1),1);

% Cracking moment in [kNm]
Mcr = fct*1e3.*(b.*h.^2)/6;
```

```
% Bending moment for defined esi in [kNm]
M = esi .* EI_cr * 1e3 ./ (d-y0);

% Check to ensure stabilized cracking stage is valid
M(M < 2.5*Mcr) = 2.5*Mcr(M < 2.5*Mcr);


% Recalculates reinforcement maximum strain in crack
esi = (M/1e3) .* (d-y0)./ EI_cr;

%% Average strain estimation by mean deformation based
% approach (Eurocode 2)
% Elastic reinforcement strain
eel = (M/1e3).*(d-0.5*h)./EI;

% Stiffening coefficient by Eurocode 2
xi = 1-(Mcr./M).^2;

% Mean strain estimation
esm = (1-xi).*eel + xi.*esi;

%% Estimation of the central zone length from experimental
% crack spacing
% Slope coefficient A [m^-1]
A = 4*Tau ./ (Es1.*diam_s1/1000);

% Length of debonding zone [m]
% ld = esi.*diam_s1./3;
ld = zeros(size(h,1),1);

% Central zone length from the strain compliance conditiona % m]
Lc = 2*((esm.*srm_exp - esi.*srm_exp + A.*ld.*ld + 0.25*A.*…
srm_exp.*srm_exp - A.*srm_exp.*ld)./A).^0.5;

% Effective zone length [m]
leff = (srm_exp - 2*ld - lc)/2;

% Minimum reinforcement strain at the location of the central % zone
es0 = esi - leff .* A;
```

D.3. *Matlab* Code of the Strain Compliance Approach for Flexural Reinforced Concrete Element Analysis

The code presented herein is intended for the analysis of flexural reinforced concrete structures, such as beams or slabs. Since the central zone length lc model is already defined at this stage, the model is inserted in the code. In order to investigate alternatives, the code would have to be modified accordingly. The present code is meant to replicate the results provided in this PhD thesis.

```
% Cleanup of Matlab workspace
close force all
clear all
clc

% Loads data file
load('phd_flex_workspace')

%% Definition of initial variables and parameters
% Experimental mean crack spacing in [m]
srm_exp = table2array(flex_rawdata(:,25));

% Concrete compressive strength from data in [MPa]
fcm = table2array(flex_rawdata(:,24));

% Calculates tensile strength of concrete in [MPa]
if fcm > 50
    fct = 2.12 *log(1+(fcm/10));
else
    fct = 0.3 * (fcm - 8).^(2/3);
end

% Elasticity modulus of tensile reinforcement in [MPa]
Es1 = table2array(flex_rawdata(:,22));

% Elasticity modulus of compressive reinforcement in [MPa]
Es2 = table2array(flex_rawdata(:,23));

% Elasticity modulus of concrete in [MPa]
Ec = 22000 * (fcm/10).^0.3;
```

```
% Equivalent diameter of all tensile rebars in [mm]
diam_s1 = table2array(flex_rawdata(:,13));

% Equivalent diameter of all concrete rebars in [mm]
diam_s2 = table2array(flex_rawdata(:,16));

% Number of tensile rebars
No1 = table2array(flex_rawdata(:,9)) + table2array(flex_rawdata(:,10));

% Number of compressive rebars
No2 = table2array(flex_rawdata(:,15));

% Reinforcement area of tensile rebars in [m2]
As1 = table2array(flex_rawdata(:,14));

% Reinforcement area of compressive rebars in [m2]
As2 = table2array(flex_rawdata(:,17));

% Section height in [m]
h = table2array(flex_rawdata(:,4));

% Section width in [m]
b = table2array(flex_rawdata(:,6));

% Effective section depth in [m]
d = table2array(flex_rawdata(:,5));

% Tensile rebar concrete cover in [m]
c = table2array(flex_rawdata(:,8));

% Compressive rebar concrete cover in [m]
as2 = table2array(flex_rawdata(:,7));

% Reinforcement ratio of elements in [decimal fraction]
ro = As1 ./ (d.*b);

% Bond stress in [MPa]
Tau = 2*fct;

%% Geometrical properties and related parameters
% Transformed area in [m2]
```

```
Atr = h .* b + ((Es1./Ec)-1).*As1 + ((Es2./Ec)-1).*As2;

% Location of neutral axis in [m]
y0 = (((As1.*(Es1./Ec)+As2.*((Es2./Ec)-1))./b).^2 ...
   + 2*(As1.*(Es1./Ec).*d + As2.*((Es2./Ec)-1).*as2)./b).^0.5 ...
   - (As1.*(Es1./Ec)+As2.*((Es2./Ec)-1))./b;

% Transformed cracked area in [m2]
Atr_cr = y0.*b + As1.*(Es1./Ec) + As2.*((Es2./Ec)-1);

% Transformed First moment of area in [m3]
Str = 0.5*(b.*h.^2) + As1.*((Es1./Ec)-1).*d + As2.*((Es2./Ec)-1).*as2;

% Transformed First moment of area of cracked section in [m3]
Str_cr = (y0.^2).*b*0.5 + As1.*(Es1./Ec).*d + As2.*((Es2./Ec)-1).*as2;

% Elastic neutral axis in [m]
yel = Str ./ Atr;

% Transformed Second moment of area in [m4]
Itr = ((b.*h.^3)/12) + (h.*b.*(yel-0.5*h).^2) + (As1.*(Es1./Ec).*(d-yel).^2)...
   + (As2.*((Es2./Ec)-1).*(yel-as2).^2);

% Transformed Second moment of area of cracked section in [m4]
Itr_cr = ((b.*y0.^3)/12) + (y0.*b.*(y0-0.5*y0).^2) +...
   (As1.*(Es1./Ec).*(d-y0).^2) + (As2.*((Es2./Ec)-1).*(y0-as2).^2);

% Flexural stiffness in [MNm2]
EI = Itr.*Ec;

% Flexural stiffness of cracked section in [MNm2]
EI_cr = Itr_cr.*Ec;

%% Loading condition calculations
% Assumed maximum reinforcement strain in crack
esi = repmat(0.0015,size(h,1),1);

% Cracking moment in [kNm]
Mcr = fct*1e3.*(b.*h.^2)/6;

% Bending moment for defined esi in [kNm]
```

```
M = esi .* EI_cr * 1e3 ./ (d-y0);

% Check to ensure stabilized cracking stage is valid
M(M < 2.5*Mcr) = 2.5*Mcr(M < 2.5*Mcr);

% Recalculates reinforcement maximum strain in crack
esi = (M/1e3) .* (d-y0)./ EI_cr;

%% Average strain estimation by mean deformation based approach (Eurocode
2)
% Elastic reinforcement strain
eel = (M/1e3).*(d-0.5*h)./EI;

% Stiffening coefficient by Eurocode 2
xi = 1-(Mcr./M).^2;

% Mean strain estimation
esm = (1-xi).*eel + xi.*esi;

%% Estimation of the mean crack spacing with the central zone model
% With debonding zone accounted for
% Slope coefficient A [m^-1]
A = 4*Tau ./ (Es1.*diam_s1/1000);

% Length of debonding zone [m]
ld = esi.*diam_s1./3;
% ld = zeros(size(esi,1),1);

% Central zone length from the strain compliance condition [m]
lc = 0.44*(d-y0);
% lc = 0.52*(d-y0);

% Quadratic equation coefficient B
B = lc.*A - 2*(esi-esm);

% Quadratic equation coefficient C
C = -(esi-esm).*(lc + 2*ld);

% Effective zone length [m]
leff = (-B + (B.^2 - 4*A.*C).^0.5)./(2*A);
```

```
% Mean crack spacing [m]
srm_calc = 2*ld + 2*leff + lc;

%% Estimation of the mean crack spacing with the central zone model
% Without considering debonding zone
% Slope coefficient A [m^-1]
A = 4*Tau ./ (Es1.*diam_s1/1000);

% Length of debonding zone [m]
ld = esi.*diam_s1./3;
% ld = zeros(size(esi,1),1);

% Central zone length from the strain compliance condition [m]
lc = 0.44*(d-y0);
% lc = 0.52*(d-y0);

% Quadratic equation coefficient B
B = lc.*A - 2*(esi-esm);

% Quadratic equation coefficient C
C = -(esi-esm).*(lc + 2*ld);

% Effective zone length [m]
leff = (-B + (B.^2 - 4*A.*C).^0.5)./(2*A);

% Mean crack spacing [m]
srm_calc_noLD = 2*ld + 2*leff + lc;
```