CAL-RECOD
A Reinforced Concrete Design Toolbox

Luis Fernando Verduzco Martínez
Jaime Moisés Horta Rangel

Contact:
Faculty of Engineering, Autonomous University of Queretaro
Universidad 231, Cerro de las campanas S/N, Santiago de Querétaro, Querétaro, Mexico
C.P. 76010
Phone: +52 442 192 1200

Preface

CAL-RECOD ™is an interactive computer program for teaching subjects of structural reinforced concrete design and analysis using computational optimization methods-algorithms and mechanics of materials theory. The name CAL-RECOD is an abbreviation of *Computer Aided Learning of Reinforced Concrete Design.* The software can be used for different types of structural problems and cases, either for beams, columns, footings or structural frames composed by such elements.

The software has been developing since January 2022 by the Faculty of Engineering of the Autonomous University of Querétaro, under the philosophy of improve the way in which the design of reinforced concrete structures is taught by higher education institutions.

The idea of development of this software package was inspired by the CALFEM software package developed by the Division of Structural Mechanics of the Lund University [Dahlblom et al., 1986],[CALFEM, 2004]. Such software has been used internationally at this point by researchers, academics and higher education students. Similar as CALFEM, it is expected that CAL-RECOD may reach international recognition and usage, and continues development not only by UAQ Engineering Faculty members but internationally by anyone with the intention of contribution.

This release represents the first version of CAL-RECOD. The software consists of MatLab functions (.m-files) both numerical and graphical ones, although there are as well ANSYS SpaceClaim script functions (.scscript files) in python language for the visualization of the designs (as a optional complement) as described in this manual. We expect that this environment increases the ease of teaching the art of design and analysis of reinforced concrete structures.

Sincerely, the authors
Santiago de Querétaro, México, 2022

Contents

1 Introduction 7
1.1 The ISR analogy 7
1.2 Rebar optimization 7
1.3 Design of structural frames 10
1.4 Resume 10

2 Optimization methods and algorithms 11
2.1 The Steepest Gradient Descent method 11
2.2 The Particle Swarm Optimization method 12

3 Structural Mechanics 13
3.1 Function: casoConcreto 13
3.2 Function: InertiaBeamCrackedSection 14
3.3 Function: AmpMomSlenderColumns 15
3.4 Function: CrackingColumnsSym 17
3.5 Function: CrackingColumnsAsym 20
3.6 Function: RealPressuresFoot 23
3.7 Function: shearFootings 25
3.8 Function: MomentDistributionFootings 27
3.9 Function: designDimFootings 29

4 ISR Optimization functions 30
4.1 ISR analysis for beams 30
4.1.1 Function: eleMecanicos2tBeams 30
4.1.2 Function: bisectionMr2tBeams 31
4.1.3 Function: Efrec2tbeams 33
4.1.4 Function: SGD1tBeamsISR 34
4.2 ISR analysis for columns 36
4.2.1 Function: eficienciaRecISRCols 36
4.2.2 Function: bisectionMr4t 38
4.2.3 Function: eleMecanicos4t 39
4.2.4 Function: widthEfficiencyCols 41
4.2.5 Function: isrColumns 43
4.3 ISR analysis for isolated footings 45
4.3.1 Function: EvaluateISR1tFoot 45
4.3.2 Function: SGD1tFootISR 46

5 Element rebar optimization functions 47
5.1 Rebar analysis for beams 47
5.1.1 Function: ISR1tRebarBeamsOptimization 47
5.1.2 Function: RebarOptimalDesignBeams 49
5.1.3 Function: EfcriticalRebarbeams 51
5.1.4 Function: bisectionMrRebarBeams 52
5.1.5 Function: eleMecanicosRebarBeams 53
5.1.6 Function: RebarDisposition1tBeams 54
5.1.7 Function: EvaluateCostbeams 55
5.1.8 Function: beamsISR 56
5.1.9 Function: ExportResultsBeam 58
5.2 Rebar analysis for columns 59
5.2.1 Function: ExportResultsColumn 59

5.2.2 Function: isrColumnsSymAsym . . . 60
5.2.3 Symmetric reinforcement . . . 62
5.2.4 Function: optimalrebarColsSym . . . 62
5.2.5 Function: RebarDisposition . . . 64
5.2.6 Function: diagramasDisposicion . . . 65
5.2.7 Function: eficienciaRec . . . 67
5.2.8 Function: bisectionMrSymRebarCols . . . 69
5.2.9 Function: eleMecanicosRebarCols . . . 70
5.2.10 Asymmetric reinforcement . . . 71
5.2.11 Function: PSOAsymmetricRebar . . . 71
5.2.12 Function: dispositionRebarAsymmetric . . . 73
5.2.13 Function: EvaluateAsymmetric . . . 74
5.2.14 Function: eficienciaRecAsymm . . . 76
5.2.15 Function: bisectionMrVarAsymm . . . 78
5.2.16 Function: eleMecanicosVarAsymm . . . 80
5.3 Rebar analysis for isolated footings . . . 82
5.3.1 Function: RebarOptionsFootings . . . 82
5.3.2 Function: dispositionRebarSquareFootings . . . 83
5.3.3 Function: dispositionRebarRectangularFootings . . . 84
5.3.4 Function: EfcriticalFootings . . . 86
5.3.5 Function: EvaluateCostRebarFoot . . . 87
5.3.6 Function: EvaluateCostISRFoot . . . 88
5.3.7 Function: isrFootings . . . 89
5.3.8 Function: ExportResultsIsolFootings . . . 91

6 Design-Analysis of 2D frames 92
6.0.1 Function: DesignRCPlaneFrameBCIF . . . 92
6.0.2 Function: PlaneFrameStaticLinearAnalysis . . . 96

7 Graphic functions 98
7.1 Reinforcement of beam cross-sections . . . 98
7.1.1 Function: beamReinforcedSection . . . 98
7.1.2 Function: MidLeftRightBeamReinforcedSections . . . 99
7.2 Interaction diagrams and rebar layout of column cross-sections . . . 100
7.2.1 Function: diagramISR . . . 100
7.2.2 Function: diagramsFinalRebarCols . . . 101
7.3 Reinforcement of isolated rectangular footing cross-sections . . . 102
7.3.1 Function: ReinforcedSectionsFooting . . . 102

8 Structural Frame System Functions 103
8.0.1 Function: WeightStruc . . . 103
8.0.2 Function: CostStruc . . . 105
8.0.3 Function: AdjustDimElemFrames . . . 106

9 Visual CALREDOC 108
9.1 General functions . . . 108
9.1.1 Function: archivomatriz . . . 108
9.1.2 Function: designdatareading . . . 109
9.1.3 Function: dimelem . . . 110
9.2 Beams . . . 111
9.2.1 Function: RectangularPrismaticSolidBeams . . . 111
9.2.2 Function: RebarRectangularSolidBeams . . . 112

9.3 Columns . . . 113
9.3.1 Function: RectangularPrismaticSolidColumns . . . 113
9.3.2 Function: RebarRectangularSolidColumns . . . 114
9.4 Isolated Footings . . . 115
9.4.1 Function: IsolatedFootings . . . 115
9.4.2 Function: RebarRectangularIsolatedFootings . . . 116

10 User's Manual Examples **117**
10.1 Ex1: Design of rectangular beams . . . 117
10.2 Ex2: Design of rectangular columns . . . 118
10.2.1 Asymmetrical reinforcement . . . 118
10.2.2 Symmetric reinforcement . . . 119
10.3 Ex3: Design of rectangular isolated footings . . . 120
10.4 Ex4: Analysis-Design of RC frames and Visual CALRECOD . . . 122
10.5 Ex5: Optimal reinforcement area in structural elements . . . 129
10.6 Ex6: Structural efficiency of a design proposal for a column cross-section . . . 129

1 Introduction

For each problem case and type of structure there are specific functions to assess a design efficiency or design optimally either cross-section dimensions and/or its reinforcement. This way, users can compare their own designs with the optimal one, given the specific problem case. Each function is adapted to a certain set of design specification criteria from the **ACI 318** code and/or the **NTC-17** Mexican code as a default basis, although each function is flexible for modifications of their design input parameters so that they may adapt to any requirement.

1.1 The ISR analogy

The optimization design processes are based on the ISR analogy [Verduzco, 2021] using the Steepest Gradient Descent (SGD) Method (for a rapid determination of a required reinforcement area) and the Particle Swarm Optimization (PSO) algorithm (for better optimal asymmetric reinforcement areas) given an initial cross-section dimension under certain load conditions **Fig. 1**. This SGD optimization approach applies always for beam elements and footing elements, but not always for column elements, as for the latter a better optimal asymmetric reinforcement may be sought, for which the PSO would then apply. **Fig. 2**

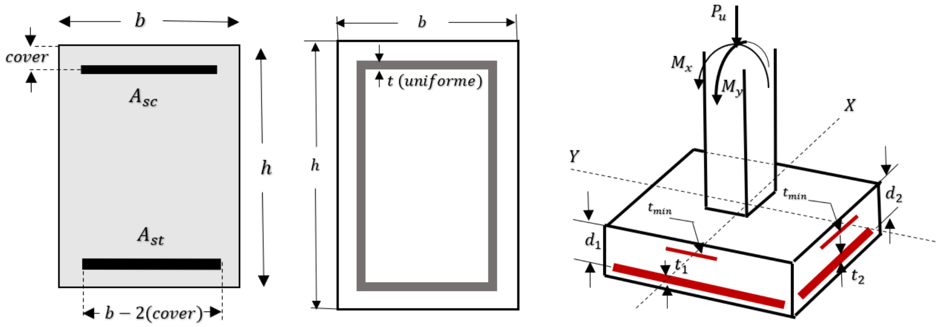


Figure 1: *ISR analogy for each type of structural element: (Left) Beam cross section, (Middle) Column cross-section, (Right) Isolated footing elements.*

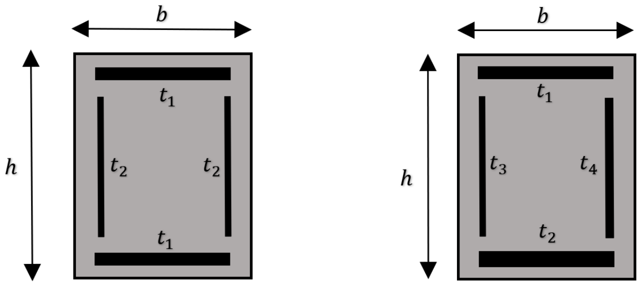


Figure 2: *ISR analogy for asymmetrical reinforced columns.*

1.2 Rebar optimization

The optimal design process of reinforcing bars depend on the structural element type case. For beams, a simple-search algorithm is used considering alternative reinforcement options regarding distribution of rebars in a particular cross-section, either of one individual rebar or in vertical two-pack rebars **Fig. 3**, given that there are limited number of potential solutions for this type of element based on its design mechanism. When a whole beam element is to be designed based on their mechanical forces distribution, then three different cross-sections along the length of the element are to be designed (left end, middle and right end), either considering cuts, free-clash and overlap reinforcement criteria **Fig. 4** or each cross-section designed independently from one another.

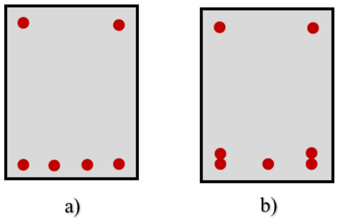


Figure 3: *Reinforcement bar options for a beam cross-section.*

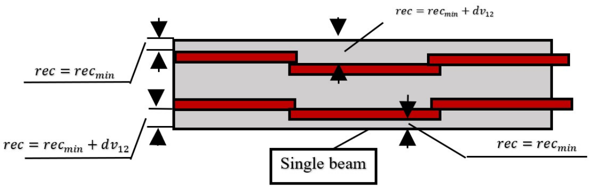


Figure 4: *Free-clash overlap reinforcement criteria for the design of beam elements along their length.*

Similar to beam elements, reinforcement in footings are usually just designed under pure flexure loads, therefore a Simple-Search algorithm is also used for these type of elements, considering an alternative reinforcement option of two-rebar packs disposed horizontally **Fig. 5** when minimum separation restrictions do not suffice for the option of individual rebars.

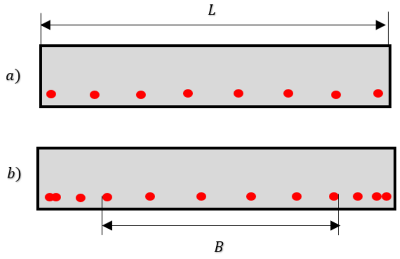


Figure 5: *Reinforcement bar options for a footing cross-section.*

For rectangular isolated footings the longitudinal bars are distributed according to the ACI-318 and NTC-17 design codes as shown in **Fig. 6**, uniformly in the smaller cross-section and non-uniformly in the longer one:

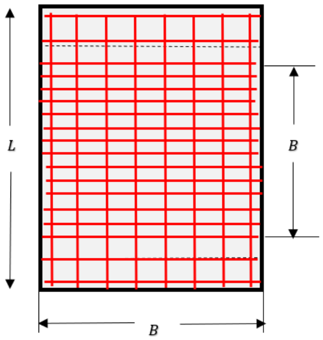


Figure 6: *Longitudinal rebar distribution over a rectangular isolated footing element.*

As for columns, several algorithms and design approaches are integrated. There are functions to design reinforcement both in an asymmetrical or symmetrical fashion. When rebar in an symmetrical arrangement is to be designed then a Simple-Search approach is best suitable. On the other hand, when an asymmetrical rebar arrangement is required, then the PSO algorithm takes place to reach the global optima faster than with a Simple-Search approach, given that the number of possibilities takes off for this latest reinforcement option.

For asymmetrical reinforcement there are two options, either using only one type of rebar for the whole arrangement or two **Fig. 7**

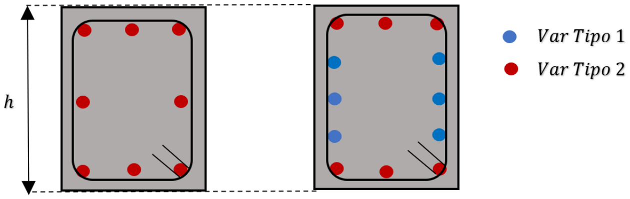


Figure 7: *(Left) Optimal symmetrical reinforcement design when only one type of rebar is considered, (Right) Optimal symmetrical reinforcement design when two different types of rebar are to be placed.*

On the other hand, for asymmetrical arrangements as many as three different types of possible topologies may take place, going from only one type of rebar to a maximum of four **Fig. 8**. As mentioned in the previous subsection, the optimal rebar designs might be obtained from an optimal ISR with the SGD method or with the PSO algorithm.

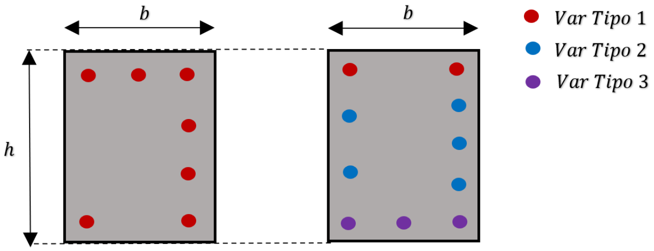


Figure 8: *Possibilities of reinforcement for an asymmetrical rebar configuration: (Left) with only one type of rebar and (Right) with as many as four different types of rebars.*

The default rebar database is integrated with the following data **Table 1** according to the most common commercial rebars in North-America, although it can be modified by any user.

Table 1: *Default rebar database.*

Type (#)	$Diam(in)$	$Diam(cm)$	Area(cm^2)
#4	0.500	1.270	1.266
#5	0.625	1.587	1.979
#6	0.750	1.905	2.850
#8	1.000	2.540	5.067
#9	1.125	2.857	6.413
#10	1.250	3.175	7.917
#12	1.500	3.810	11.400

1.3 Design of structural frames

There are also functions for the optimal design of structural frame systems for certain given initial element cross-section dimensions. At this point there are far more functions for the analysis of 2D frames rather than 3D. The functions hereby presented are able to analyse structural frames taking on account the detailed rebar on each element in a synchronized manner, out of a previous design **Fig. 9**, such that the mere self-weight of the reinforcement is considered in the analysis and even the initial dimensions for each element can be modified in case they are not fit for an optimal reinforcement design.

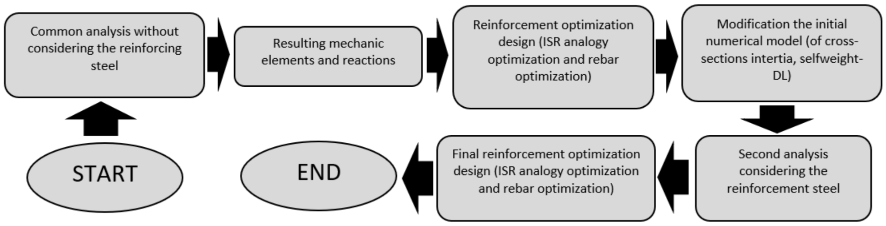

Common analysis without considering the reinforcing steel
Resulting mechanic elements and reactions
Reinforcement optimization design (ISR analogy optimization and rebar optimization)
Modification the initial numerical model (of cross-sections intertia, selfweight-DL)
START
END
Final reinforcement optimization design (ISR analogy optimization and rebar optimization)
Second analysis considering the reinforcement steel

Figure 9: *Coupled design-analysis process for the structural frames considering optimal steel rebar reinforcement.*

The way in which such designs are carried on are through sub-functions that execute design revisions tasks based on the *Strong Column - Weak Beam* criteria and beam-column node shear design. On the other hand, footings dimensions may be modified in case the initial ones do not comply with contact-pressure loads and shear criteria.

1.4 Resume

In general, all of the available functions can be organized into six different groups **Table 2**:

Table 2: *Functions available organized in groups.*

Function Group	Description
Structural mechanics	This group has the main objective of computing general calculations for the determination of mechanic properties of elements' cross-sections such as moment of inertia of cracked and non-cracked sections or resistance
ISR optimization	To determine an optimal reinforcement area for a given element cross-section dimension using either the SGD method or the PSO algorithm
Element rebar optimization	For the optimal design of configuration of rebar for any type of structural element (either through Simple-Search for beams, footings and symmetrical reinforcement in columns, or with the PSO for asymmetrical reinforcement in columns)
Design-Analysis of 2D frames	For the analysis-design of 2D frames as a coupled process. Available functions include static linear and non-linear analysis and dynamic static linear analysis. Some functions work through a CALFEM function(s)
Graphic functions	For plotting of reinforced designed cross-section, interaction diagrams for columns, optima design convergence graphics, etc
Visualization functions	Functions in python language for ANSYS SpaceClaim for the visualization of designs using the main CALRECOD MatLab functions

2 Optimization methods and algorithms

In this section a brief introduction and description of the optimization methods and algorithms used in this software are described, including: the Steepest Gradient Descent method and the Particle Swarm Optimization method.

2.1 The Steepest Gradient Descent method

Every algorithm for non-constrained gradient based optimization can be formulated as follows, starting with an iteration index $k = 0$ and from point x_k.

1. Prove convergence: if convergence conditions are satisfied, then the process may stop and x_k would be the solution, otherwise the process continues
2. To compute search-direction: compute vector ρ_k which defined the direction (positive or negative) in the $n - dimension$ search space
3. Compute step-length: To find a positive scalar α_k such that $f(x_k + \alpha_k \rho_k) < f(x_k)$
4. To update design variables: State $x_{k+1} = x_k + \alpha_k \rho_k$, $k = k + 1$ and return to step 1

The difference between all gradient-based optimization methods is the computation of the search-direction vector. In the *Steepest Gradient Descent method* of **Algorithm** 2.1 the search-direction vector is condition-based with possibilities either -1, and represents a simple solution approach for optimal convergence of concave or convex functions.

Algoritmo 2.1: Pseudo-code: Steepest Gradient Descent method

```
BEGIN
  for Nmodels=1:nm
    $t_k = inital \to t_0$
    Compute $f(initial \to t) = f(t_k)$
    While $f(t_k) > rango_{sup}$ or $f(t_k) < rango_{inf}$
      Compute $g(t_k) = \nabla f(t_k)$
      Compute search direction $p_k$
      if $f(t_k) < rango_{inf}$
        $p_k = 1$
      else if $f(t_k) > rango_{sup}$
        $p_k = -1$
      End if
      Update the current $t_{k+1} = t_k + \alpha_k(p_k)$
      $\alpha_k = -\frac{g(x_k)}{||g(x_k)||}$
      Compute $f(t_{k+1})$
      k=k+1;
    End While
    $t_{final} = t_k$
    $f_{final} = f(t_k)$
  End for
END
```

2.2 The Particle Swarm Optimization method

The PSO algorithm, inspired by the social behaviour of bird flocking. The first swarm model was developed in the 80's by Craig Raynolds [Raynolds, 1987] and then improved by Eberhart and J.Kennedy [Kennedy & Eberhart, 1995] in the 90's in its standardized form, in which potential solutions are regarded as particles with respective positions and velocities in a given time dt. Each particle is evaluated through its position to assign it a performance value based on the objective function. The position and velocities are updated for each iteration and the ones with the best performances are stored (globally and locally) until a termination condition is reached. The algorithm is presented as following in pseudo-code:

Algoritmo 2.2: Pseudo-code: Particle Swarm Optimization algorithm

```
1.- Initialize positions and velocity of each particle $p_i$
    $x_{ij} = x_{min} + r(x_{max} - x_{min}), i = 1, ..., N, j = 1, ...n$
    $v_{ij} = \frac{\alpha}{\Delta t}(-\frac{x_{max} - x_{min}}{2} + r(x_{max} - x_{min})), i = 1, ..., N, j = 1, ..., n$
2.- Evaluate each particle in the swarm with the objective function $f(x_i), i = 1, ..., N$
3.- Update best position (if a. complies) and best global position (if b. complies)
    a.) If $f(x_i) < f(x_i^{pb}$ then $x_i^{pb} \cdots x_i$
    b.) If $f(x_i) < f(x^{sb})$ then $x^{sb} \cdots x_i$
4.- Update velocities and positions:
    $v_{ij} \cdots v_{ij} + c_1 q(\frac{x_{ij}^{pb} - x_{ij}}{\Delta t}) + c_2 r(\frac{x_j^{sb} - x_{ij}}{\Delta t}), i = 1, ..., N, j = 1, ..., n$
    Constraint velocities, such that $|v_{ij}| < v_{max}$
    $x_{ij} \cdots x_{ij} + v_{ij}\Delta t, i = 1, ..., N, j = 1, ..., n$
5.- Return to step 2, unless the termination criteria complies.
```

3 Structural Mechanics

3.1 Function: casoConcreto

Purpose: to compute the contribution of resistance of the concrete compression zone of a rectangular beam cross-section, regarding axial and bending forces.

Sintax:

$$elemConc = casoConcreto(a, fdpc, b, h)$$

Description:

Output variables:

- elemConc: vector that contains the output $F_c, M_c]$ of resistant axial and bending forces

Input variables:

- a is the reduced depth of neutral axis of the cross-section in question
- $fdpc$ is the factored value of f_c' as $0.85f_c'$ according to the [ACI 318-19] code
- b, h are the cross-section dimensions

Theory:

Is considered that concrete only withstand compression forces, therefore, the zone in tension **Fig. 10** divided by the neutral axis c from the compression zone does not contribute in the total resistance of the whole cross-section. Such tension zone is assumed to cracked. Where C represents the resistant force in compression as (1).

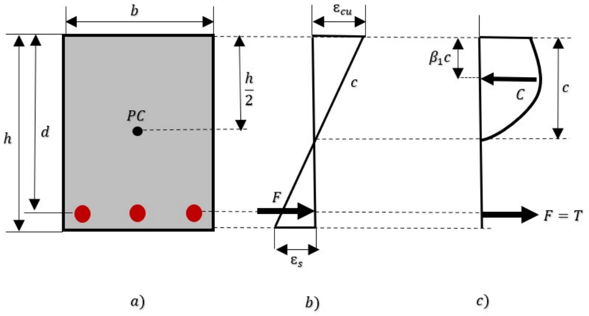


Figure 10: .

$$F_c = C = ab0.85f_c' \tag{1}$$

The bending resistance (2), on the other hand, assumes that the Plastic Centroid of the cross-section is in the same position as the Geometric Centroid **Fig. 10** (at the depth of $\frac{h}{2}$).

$$M_c = F_c(\frac{h}{2} - \frac{a}{3}) \tag{2}$$

3.2 Function: InertiaBeamCrackedSection

Purpose: To determine the modified inertia momentum of a cracked beam cross-section.

Syntax:

$Inertia\_modif = InertiaBeamCrackedSection(fc, E, areabartension, ...b, h, h\_rec)$

Description:

Output variables:

- $Inertia\_modif$ The modified inertia momentum for the beam cross-section, considering it is cracked, that is, when the tension stress is greater than the rupture modulus of the used concrete

Input variables:

- $areabartension$ rebar area in tension
- fc is the f'_c used
- b, h are the cross-section dimensions
- h_rec is the concrete cover along the height dimension

Theory:

Is considered that concrete only withstand compression forces, therefore, the zone in tension **Fig. 11** divided by the neutral axis c from the compression zone does not contribute in the inertia momentum of the whole cross-section. Such tension zone is assumed to be cracked. Where C represents the resistant force in compression and T the total resistant force in tension, so that the inertia momentum is modified as (3)

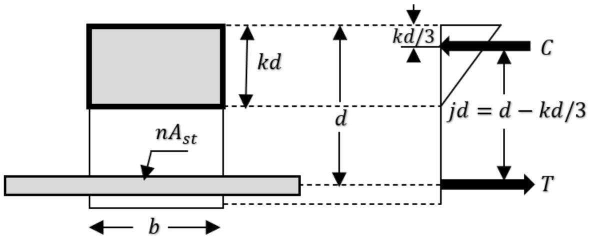


Figure 11: *Transformed section mechanism to consider beam cracked cross-sections.*

$$I_c = \frac{by^3}{12} + \frac{by^2}{4} + nA_s(d-y)^2 \tag{3}$$

3.3 Function: AmpMomSlenderColumns

Purpose: To compute the amplified moments for a column considering the slenderness effects.

Syntax:

$[Mcx] = AmpMomSlenderColumns(b, h, fc, Pu, Wu, , ...$
$m1b, m1s, m2b, m2s, delta, height, inertiax, Vu)$

Description:

Output variables:

- Mcx: amplified moment in the X-direction $(Ton \cdot m)$

Input variables:

- b, h cross-section dimensions of column (cm)
- fc compressive concrete strength $(\frac{Kg}{cm^2})$
- Pu axial load over the column's cross-section (Kg)
- Wu is the sum of axial loads in the columns of the i floor (in which the column of analysis is placed) from the n floor (Kg)
- $m2b$ is the greater moment generated by the forces that caused depreciable displacement on the structure: $(Ton \cdot m)$
- $m2s$ is the greater moment generated by the forces that caused the greater displacement on the structure: $(Ton \cdot m)$
- $m1b$ is the smaller moment generated by the forces that caused depreciable displacement on the structure: $(Ton \cdot m)$
- $m1s$ vector containing the smaller moments generated by the forces that caused the greater displacement on the structure: $(Ton \cdot m)$
- $delta$ lateral displacement at the top of the column (cm)
- $height$ effective length of the column element (cm)
- $inertiax$ inertia momentum for the axis direction in question of the cross-section (cm^4)
- Vu shear base force (Kg)

Theory:

By the application of a analytical method, this function computes the amplification factors due to slenderness effects as(4), where M_2 is the greater of both moments at the ends of the element, M_{2b} is the vector containing the moments generated by the forces that caused depreciable displacement on the structure, M_{2s} is the vector containing the moments generated by the forces that caused the greater displacement on the structure. Finally, F_{as}

is the amplification factor determined as (5)

$$M_2 = M_{2b} + F_{as} M_{2s} \tag{4}$$

$$1.5 \geq [F_{as} = \frac{1}{1-\lambda}] \geq 1.0 \tag{5}$$

When $\frac{L_c}{r} \geq \frac{35}{\sqrt{\frac{P_u}{f_c' A_g}}}$ then the amplified factor is computed as (6), where M_2 is also the greater of both moments at the ends of the column, F_{ab} is the amplification factor determined as (7), where $[C_m = 0.6 + 0.4\frac{M_1}{M_2}] \geq 0.4$, P_u is the axial design load and P_c is the buckling critical load of Euler defined as (8) for which $H' = kl$ being $0.5 \leq k \leq 1.0$ the slenderness factor and $EI = 0.4\frac{E_c I_g}{1+u}$ (E_c is the Elasticity Modulus, I_g is the inertia momentum of the cross-section and u is the ratio between the axial design dead load and the sum of the design dead and live load $u = \frac{P_D}{P_D + P_L}$ for which in most cases a value of 0.5 is acceptable).

$$M_2 = F_{ab} M_2 \tag{6}$$

$$F_{ab} = \frac{C_m}{1 - \frac{P_u}{0.75 P_c}} \tag{7}$$

$$P_c = \frac{\pi^2 EI}{H'^2} \tag{8}$$

3.4 Function: CrackingColumnsSym

Purpose: To compute the reduced inertia momentum of a column cross-section with symmetrical reinforcement considering cracking mechanisms.

Syntax:

$[InertiaXYmodif, Atransf\_xy] = CrackingColumnsSym(h, b, fdpc, rec, t\_value\_x, excentricityXY, ...$ $t\_value\_y, Pu, cxy, conditionCrack, E)$

Description:

Output variables:

- $InertiaXYmodif$ is the modified reduced inertia momentum for both axis directions of the cross-section considering cracking mechanisms: $[Ix, Iy]$
- $Atransf\_xy$ is the transformed effective area for both axis directions according to the cracking mechanism (cracked or non-cracked)

Input variables:

- b, h cross-section dimensions of column
- $fdpc$ compressive concrete strength, reduced by the factor 0.85 according to code
- Pu axial load over the column's cross-section
- rec concrete cover for both axis direction of the cross-section; format: $[cover_x, cover_y]$
- $eccentricityXY$ axial load eccentricity for both axis directions: $[e_x, e_y]$
- tx, ty ISR width for both axis directions
- cxy neutral axis depth for both axis directions of the cross-section, corresponding to the optimal reinforcement design
- $conditionCrack$ parameter that indicates which mechanism to consider neglecting the rupture modulus fr_{ot}: format is $''Cracked''/''Non-cracked''$
- E: Elasticity Modulus

Theory:

For high axial load eccentricity surpassing the limit (9) a cracked cross-section inertia is computed with equation (10) based on the **Fig. 12**, with an equivalent effective transformed area as (11). The computation of e_{lim} involves the variables: whose variables are $fr_{ot} = 0.8(2\sqrt{f_c''})$ rupture modulus, I_g gross inertia momentum for the axis in question, P_u axial load and A_t which is determined by (13).

$$e_{lim} = \frac{2(\frac{P}{A_t} + f_r)I_g}{Ph} \tag{9}$$

$$\begin{aligned} I_t = Ixx_{ag_{sym}} = \frac{bc^3}{12} + \frac{bc^2}{4} + nt_1(b-2cover)(h-cover-c)^2 + (n-1)t_1(b-2cover)(c-cover)^2 + ... \\ \frac{2(n-1)t_2(\frac{h}{2}-cover-k)^3}{12} + 2(n-1)t_2(\frac{h}{2}-cover-k)(\frac{1}{2}(\frac{h}{2}-cover-k))^2 + ... \\ \frac{2nt_2(\frac{h}{2}-cover+k)^3}{12} + 2nt_2(\frac{h}{2}-cover+k)(\frac{1}{2}(\frac{h}{2}-cover+k))^2 \end{aligned} \tag{10}$$

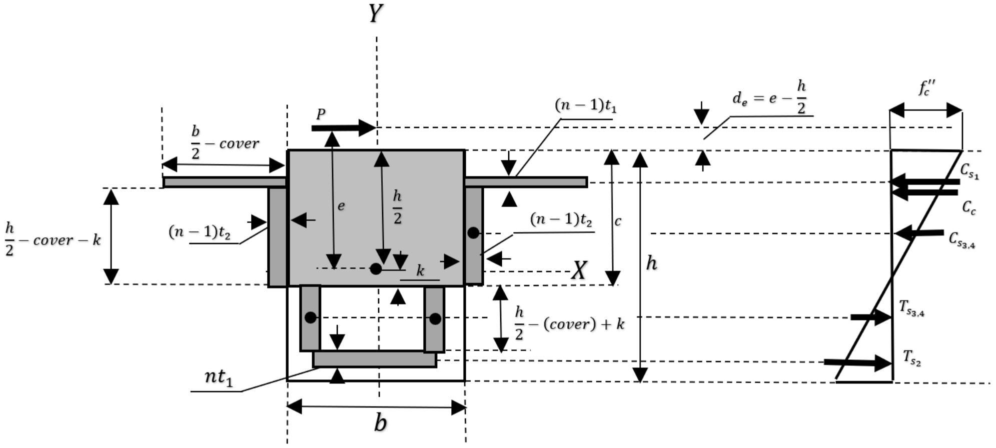


Figure 12: *Transformed cracked cross-section mechanism with symmetric reinforcement..*

$$A_t = A_{ag_{sym}} = b(h-c) + 2(b-2cover)((n-1)+n)t_1 + 2(\frac{h}{2}-cover-k)(n-1)t_2 + 2(\frac{h}{2}-cover+k)(n-1)t_2 \tag{11}$$

On the other hand, for a non-cracked cross-section for which the axial load eccentricities are lower than e_{lim} then equation (12) applies based on **Fig. 13** with a corresponding transformed cross-section area computed as (13):

$$\begin{aligned} I_t = Ixx_{no-ag_{sym}} = \frac{bh^3}{12} + 2\frac{(b-2(cover))((n-1)t_1)^3}{12} + ... \\ 2(n-1)t_1(b-2(cover))(\frac{h}{2}-cover)^2 + \frac{2(n-1)t_2(h-2(cover))^3}{12} \end{aligned} \tag{12}$$

$$A_t = A_{no-ag_{sym}} = bh + 2(b-2cover)(n-1)t_1 + 2(h-2cover)(n-1)t_2 \tag{13}$$

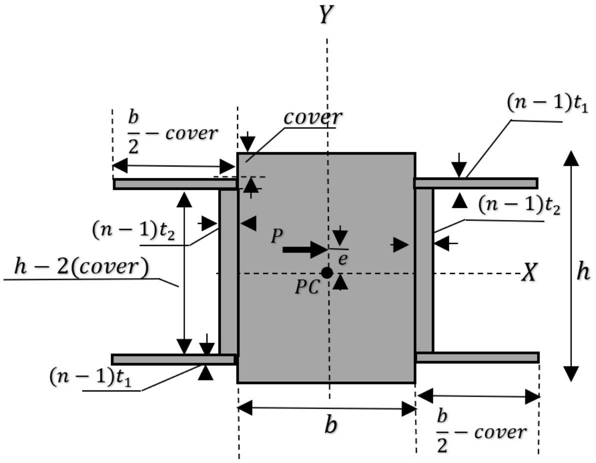


Figure 13: *Transformed non-cracked cross-section mechanism with symmetric reinforcement for small axial load eccentricities* $e \leq e_{lim}$.

3.5 Function: CrackingColumnsAsym

Purpose: To compute the reduced inertia momentum of a column cross-section with asymmetrical reinforcement considering cracking mechanisms.

Syntax:

$[InertiaXYmodif, Atransfxy] = CrackingColumnsAsym(h, b, fdpc, rec, eccentricityXY, ...$
$t1bar, t2bar, t3bar, t4bar, Pu, cxy, conditionCracking, cp)$

Description:

Output variables:

- $InertiaXYmodif$ is the modified reduced inertia momentum for both axis directions of the cross-section considering cracking mechanisms: $[Ix, Iy]$
- $Atransfxy$ is the transformed effective area for both axis directions according to the cracking mechanism (cracked or non-cracked)

Input variables:

- b, h cross-section dimensions of column
- $fdpc$ compressive concrete strength, reduced by the factor 0.85 according to code
- Pu axial load over the column's cross-section
- rec concrete cover for both axis direction of the cross-section
- $eccentricity\_xy$ axial load eccentricity for both axis directions: $[e_x, e_y]$
- $t1bar, t\_value\_2$ ISR horizontal width (X-axis)
- t_value_3, t_value_4 ISR vertical width (Y-axis)
- cxy neutral axis depth for both axis directions of the cross-section, corresponding to the optimal reinforcement design
- $condition\_cracking$ parameter that indicates which mechanism to consider neglecting the rupture modulus fr_{ot}: format is $''Cracked''/''Non-cracked''$
- cp Plastic Center depth
- E: Elasticity Modulus

Theory:

For high axial load eccentricity surpassing the limit (9) a cracked cross-section inertia is computed with equation (14) based on the **Fig. 14**, with an equivalent effective transformed area as (15). The computation of e_{lim} involves the variables: $fr_{ot} = 0.8(2\sqrt{f''_c})$ rupture modulus, I_g gross inertia momentum for the axis in question, P_u axial load and A_t which is determined by (17).

$$
\begin{aligned}
I_t = Ixx_{ag_{asym}} = \frac{bc^3}{12} + \frac{bc^3}{4} + nt_2(b-2cover)(h-cover-c)^2 + (n-1)t_1(b-2cover)(c-cover)^2 + ... \\
\frac{(n-1)(t_3+t_4)(\frac{h}{2}-cover-k)^3}{12} + (n-1)(t_3+t_4)(\frac{h}{2}-cover-k)(\frac{1}{2}(\frac{h}{2}-cover-k))^2 + ... \\
\frac{n(t_3+t_4)(\frac{h}{2}-cover+k)^3}{12} + n(t_3+t_4)(\frac{h}{2}-cover+k)(\frac{1}{2}(\frac{h}{2}-cover+k))^2
\end{aligned}
\tag{14}
$$

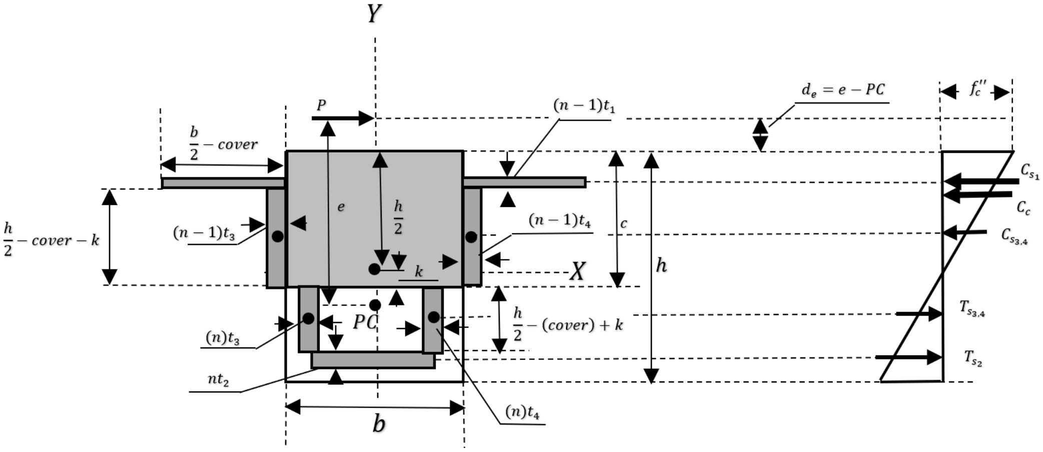


Figure 14: *Transformed cracked cross-section mechanism with asymmetric reinforcement..*

$$
\begin{aligned}
A_t = A_{ag_{asym}} = b(h-c) + (b-2cover)((n-1))t_1 + (b-2cover)(n)t_2 + ... \\
2(\frac{h}{2}-cover-k)(n-1)t_3 + 2(\frac{h}{2}-cover+k)(n)t_4
\end{aligned}
\tag{15}
$$

On the other hand, for a non-cracked cross-section for which the axial load eccentricities are lower than e_{lim} then equation (12) applies based on **Fig. 15** with a corresponding transformed cross-section area computed as (17):

$$
\begin{aligned}
I_t = Ixx_{no-ag_{asym}} = \frac{bh^3}{12} + bh(\frac{h}{2}-PC)^2 + (n-1)t_1(b-2cover)(cover-CP)^2 + ... \\
(n-1)t_2(b-2cover)(h-cover-CP)^2 + \frac{(n-1)t_3(h-2(cover))^3}{12} + ... \\
((n-1)t_3)(h-2(cover))(\frac{h}{2}-CP)^2 + \frac{(n-1)t_4(h-2cover)^3}{12} + ... \\
(n-1)t_4(h-2cover)(\frac{h}{2}-PC)^2
\end{aligned}
\tag{16}
$$

$$
\begin{aligned}
A_t = A_{no-ag_{asym}} = bh + (b-2cover)(n-1)t_1 + (b-2cover)(n-1)t_2 + ... \\
(h-2cover)(n-1)t_3 + (h-2cover)(n-1)t_4
\end{aligned}
\tag{17}
$$

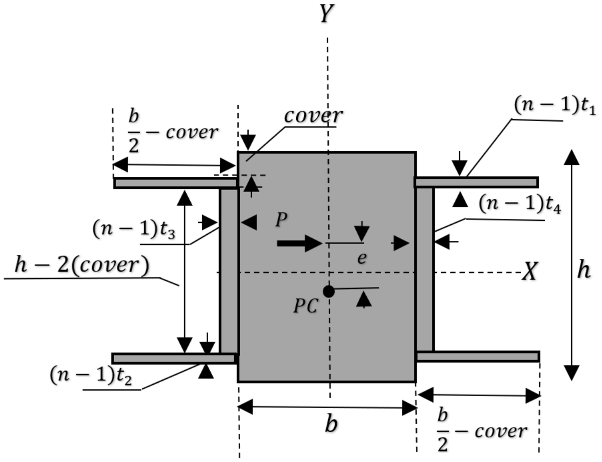


Figure 15: *Transformed non-cracked cross-section mechanism with asymmetric reinforcement for small axial load eccentricities* $e \leq e_{lim}$.

3.6 Function: RealPressuresFoot

Purpose: To compute the distribution of bending moments over the transversal cross-sections of the footing based on the actions applied through the supporting column.

Syntax:

$$[qu01, qu02, qu03, qu04, qprom] = RealPressuresFoot(load\_conditions\_cols, be, le)$$

Description:

Output variables:

- $qu01$ is the distributed pressure at the upper-right corner of the isolated footing
- $qu02$ is the distributed pressure at the upper-left corner of the isolated footing
- $qu03$ is the distributed pressure at the lower-left corner of the isolated footing
- $qu04$ is the distributed pressure at the lower-right corner of the isolated footing
- $qprom$ is the average distributed pressure considering the four distributed pressures at the corners of the element. This pressure is to be compared with the maximum withstanding contact pressure restriction

Input variables:

- $load\_conditions\_cols$ is the vector containing the critical load condition that the supporting column imposed over the isolated footing (Ton,m)
- be, le are the transversal dimensions of the isolated footing on plan view (cm)

Theory:

For isolated footings undergoing eccentric biaxial loads a linear distribution of contact pressures or stresses is considered (18) so that $q_{real} < q_{max} < q_{adm}$ (19) (see **Fig. 16**)

$$q_{max-min} = \frac{P}{A} \pm \frac{6M_x}{BL^2} \pm \frac{6M_y}{LB^2} \tag{18}$$

$$q_{real} = \frac{P}{LB} \tag{19}$$

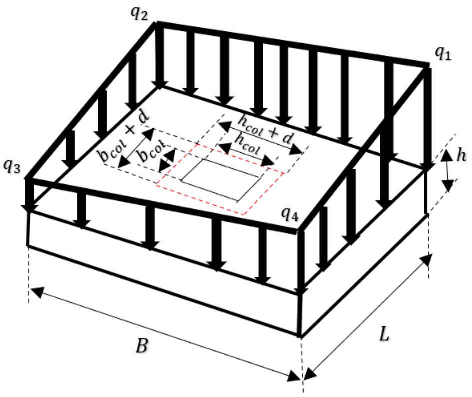


Figure 16: *Linear distribution of contact pressures over an isolated rectangular footing undergoing biaxial eccentric loads.*

3.7 Function: shearFootings

Purpose: To compute the demand of shear stresses and resistant ones of an isolated footing subject to biaxial eccentric actions, considering two mechanisms (punching and beam shearing).

Syntax:

$[d, sigma13, sigma24, sigma12, sigma34, a13, a24, a12, a34, bp, lp, qmax13, ...$
$qmax24, qmax12, qmax34] = shearFootings(be, le, qprom, dimCol, qu01, qu02, qu03, ...$
$qu04, pu, d, fc)$

Description:

Output variables:

- d effective modified height dimension based on the critical acting shear demand over the isolated footing
- $a13, a24, a12, a34$ are max effective unit-stress acting on each axis direction of the footing due to the contact pressures (see **Fig. 18**) for each of its four plan boundaries
- $qmax13, qmax24, qmax12, qmax34$ are the max contact pressures for each of the four footing boundaries at their respective ends (see **Fig. 18**)
- $sigma13, sigma24, sigma12, sigma34$ define the stress magnitude at the other end of the block unit stress (see **Fig. 18**) for each of the four footing boundaries

Input variables:

- $qprom$ is the average contact pressure considering the pressures at the four footing's corners $\frac{Kg}{cm^2}$
- $dimCol$ are the cross-section dimensions of the column that the footing supports (cm)
- $qu01, qu02, qu03, qu04$ are the contact pressure magnitudes at the four footing's corners (see **Fig. 18**)
- be, le are the transversal dimensions of the isolated footing on plan view (cm)
- pu is the axial load reaction from the column (Kg)
- d is the effective footing height (cm)
- fc is the f'_c used for the footing ($\frac{Kg}{cm^2}$)

Theory:

In order to design effective the height of the footing, the shear actions and demands have to be considered (both punching shear (20) and shear as a beam mechanisms (21) **Fig. 17**). The condition $V_u < V_{CR}$ is complied at any cost. The punching shear acts over the critical perimeter surrounding the supporting column **Fig. 17 (Right)** and when beam-shear is considered is acts as in **Fig. 17 (Left)**. The maximum of the resultant effective height values for both mechanisms d is considered for design.

$$Punching - Shear = \begin{cases} V_{net} = P_u - q_{av}(b_{col} \cdot h_{col}) \\ A_{shear} = d(2(b_{col} + d) + 2(h_{col} + d)) \\ V_u = \frac{V_{net}}{A_{shear}} \\ V_{CR} = 0.85\sqrt{f'_c} \end{cases} \tag{20}$$

$$Beam - Shear = \begin{cases} V_{net} = \frac{1}{2}B(q_{av})(L - b_{col} - d) \\ V_{CR} = \frac{1}{2}0.85\sqrt{f'_c} \\ d = \frac{q_{av_{max}}(\frac{1}{2}(L - b_{col} - d))}{BV_{CR}} \end{cases} \tag{21}$$

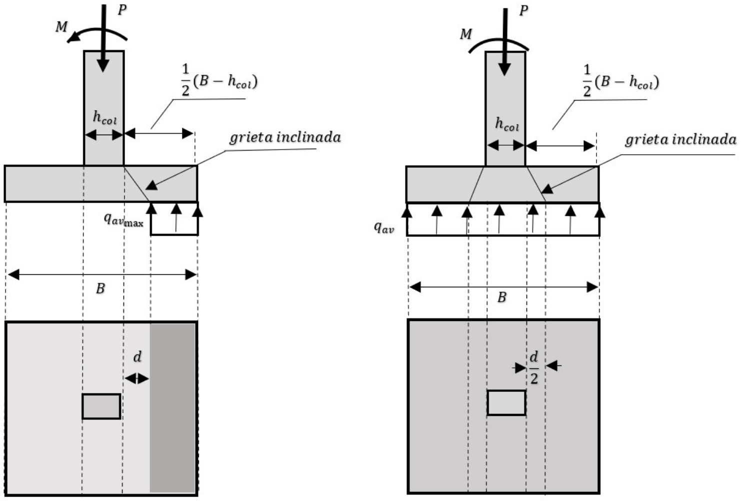


Figure 17: *(Left). Design shear mechanism as a beam, (Right). Design punching shear mechanism.*

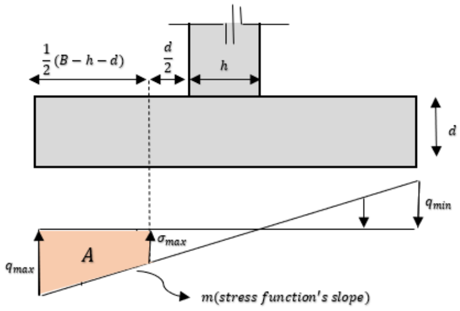


Figure 18: *Max unit-stress (contact pressure) distribution for a footing boundary.*

3.8 Function: MomentDistributionFootings

Purpose: To compute the effective bending moment acting at the transversal cross-sections of the footing, given the max unit contact pressures at each of the four footing's boundaries.

Syntax:

$$[m] = Moment\_Distribution\_Footings(sigma1, sigma2, qmax1, qmax2, a1, a2, dimp, dimfoot)$$

Description:

Output variables:

- m effective bending moment acting over a transversal footing cross-section

Input variables:

- $sigma1, sigma2$ are magnitudes of the contact pressures at the other end of both effective unit contact pressures at each side of the axis direction in question. **Fig.** 19
- $qmax1, qmax2$ are the max contact pressures at both sides of the axis direction of anlaysis. **Fig.** 19
- $a1, a2$ are the effective unit contact pressures at both sides of the axis direction of analysis. **Fig.** 19
- $dimp$ is the effective dimension over which the flexure stress acts
- $dimfoot$ is the footing dimension over the axis direction in analysis

Theory:

For a given footing transversal axis direction **Fig.** 19 the average max shear is computed considering both sides stresses of that axis direction so that a bending moment is calculated considering the effective flexure area a cantilever beam **Fig.** 20.

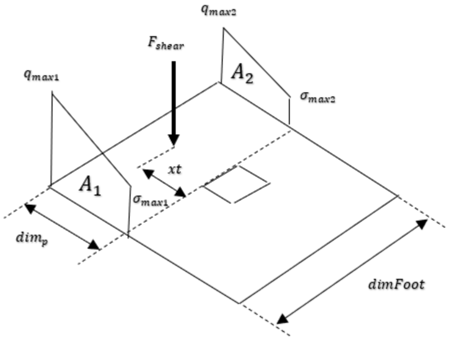


Figure 19: *Contact pressure distribution at both sides of an axis direction of analysis.*

Once the average unit contact pressure is determined, the total shear force F_{shear} is computed along the axis direction in question and its centroid xt is used to determine the bending moment as (22). When the contact pressure is considered of constant magnitude q_{max}, then the calculation of M would be determined as in (**??**) for each axis direction.

$$M = F_{shear} \cdot xt \tag{22}$$

$$Mu_{axis} = q_{max}(dimfoot\frac{dimp}{2}) \tag{23}$$

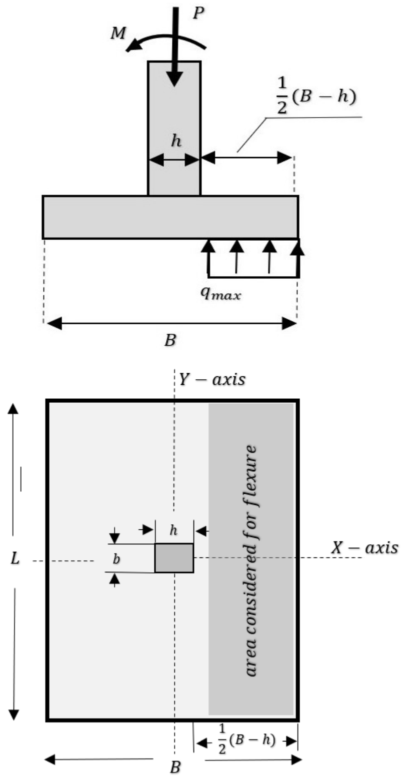


Figure 20: *Flexure mechanism of an isolated rectangular footing as a cantilever beam subject to the contact pressures.*

3.9 Function: designDimFootings

Purpose: To design the transversal dimensions of a rectangular isolated footing, based on the acting vertical reaction from the intersecting column and the admissible load of soil considering the Safety Design Factor, as well as the cross-section dimensions of the intersecting column, so that the footing dimension may be at least 40 cm wider than such column cross-section dimensions.

Syntax:

$[be, le, contact_pressure] = designDimFootings(pu, qu, dimCol, ...$
$hfooting, rec)$

Description:

Output variables:

- be, le transversal isolated footing dimensions
- *contact_pressure* is the resulting contact pressure from the soil (less or equal than qu)

Input variables:

- pu is the vertical reaction from he supporting column onto the footing
- qu is equal to $FS(q_{adm})$
- *dimCol* column cross-section dimensions $[b, h]$
- $hfooting$ is the height or width of the isolated footing
- rec is the concrete cover (cm)

4 ISR Optimization functions

4.1 ISR analysis for beams

4.1.1 Function: eleMecanicos2tBeams

Purpose: to compute the sum of resistant forces of a beam cross-section, considering the contribution of steel in tension, steel in compression and concrete in compression.

Syntax:

$$eleMec = eleMecanicos2tBeams(c, a, fdpc, h, b, b\_rec, h\_rec, E, t1, t2)$$

Description:

Output variables:

- eleMec: vector that contains the output $[\sum F_s, \sum M_s; F_c, M_c]$

Input variables:

- t_1,t_2 are the given width of ISR in compression and tension, respectively
- b_rec,h_rec are the concrete cover parameters horizontally and vertically, respectively (cm)
- $fdpc = 0.85f'_c$: according to [ACI 318-19]

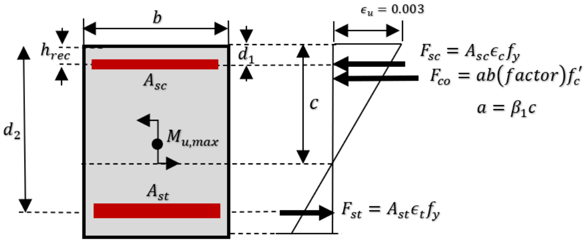
Theory:

The function considers the location of the Plastic Center (PC) the same as the Geometric Center (GC) (which is at a depth $\frac{h}{2}$), so that the resistant moment is calculated as (24), where $Fs_i = As_i E_y \epsilon_i$ for reinforcement steel.

$$M_R = (\sum Fs_i + \beta_1 ab0.85f'_c)(\frac{h}{2} - d_i) \tag{24}$$

4.1.2 Function: bisectionMr2tBeams

Purpose: to determine the neutral axis depth and resistant bending moment of reinforced beam cross-section taking on account both the steel in compression and steel in tension with the aid of the bisection method as a root for the pre-established equilibrium condition $\sum F = 0$.


$\epsilon_u = 0.003$
b
h_{rec}
d_1
$F_{sc} = A_{sc}\epsilon_c f_y$
$F_{co} = ab(factor)f_c'$
$a = \beta_1 c$
A_{sc}
c
d_2
$M_{u,max}$
A_{st}
$F_{st} = A_{st}\epsilon_t f_y$

Sintax:

$$[Root] = bisectionMr2tBeams(c1, c2, fr, E, t1, t2, h, b, b\_rec, h\_rec, fdpc, beta1, ea);$$

Description:

In order to calculate the resistant bending moment, the equilibrium condition of forces $\sum F = 0$ must be complied. For this function, it assumed that the acting axial load on the beam cross-section is too small and can be neglected. This axial load should be evaluated by the user, such that such axial load is smaller than the tenth part of the cross-section axial load resistance $P_{oc} < \frac{1}{10}(bh - A_s)f_c'$ according to the [ACI 318-19] code and other international codes.

Output variables:

- raiz: vector that contains the output $[c, \sum F_i, M_R]$

Input variables:

- t_1,t_2 are the given width of ISR in compression and tension, respectively
- b_rec,h_rec are the concrete cover parameters horizontally and vertically, respectively (cm)
- $fdpc = 0.85 f_c'$: according to [ACI 318-19]
- β_1 is determined as following (73) in units Kg, cm

$$0.65 \leq (\beta_1 = 1.05 - \frac{f_c'}{1400}) \leq 0.85 \tag{25}$$

- ea is the approximation root error

Theory:

The nominal resistant bending moment is determined as (26) for a given neutral axis depth value c with the function **eleMecanicos2tBeams** (p.30).

$$M_n = \sum F_s + (\beta_1 c)b0.85f'_c(\frac{h-a}{2}) \tag{26}$$

The neutral axis depth is restricted by a ductility strain requirement established by code **ACI 318-19** as (27):

$$c \leq \frac{d}{\frac{0.005}{0.003}+1} \tag{27}$$

For more reference of the bisection method see [Chapra & Canale, 2015]

4.1.3 Function: Efrec2tbeams

Purpose: Calculates the structural efficiency of beam cross-section according to the applied load conditions.

Syntax:

$$[maxEf, Mr, c] = Efrec2tbeams(load\_conditions, fc, factor\_fc, E, h, b, Ast, Asc, h, brec, hrec, fy, beta1)$$

Description:
Output variables

- $maxEf, Mr, c$ is the structural efficiency of the reinforced beam cross-section (0-1), the resistant beding moment and the neutral axis depth, respectively

Input variables

- $load\_conditions = [n\_load, M_u]$ size nloads x 2
- $factor\_fc$ is determined by de applicable design code. The [ACI 318-19] specifies it as 0.85
- β_1 is determined as following (73) in units Kg, cm

$$0.65 \leq (\beta_1 = 1.05 - \frac{f'_c}{1400}) \leq 0.85 \tag{28}$$

- b_rec, h_rec is the concrete cover along the width and height cross-section dimension, respectively (cm)
- h, b cross-section dimensions (cm)
- ast, asc are the reinforcement steel area in tension and compression, respectively (cm^2)
- E Elasticity Modulus of steel reinforcement $(\frac{Kg}{cm^2})$

Theory:
Once the equilibrium condition is complied, with its respective neutral axis and nominal resistant bending moment with the **bisectionMr2tbeams** function (p.31), the resistance reduction factor ϕ (52) is then applied as (53) according to the condition of *tension controlled* or *compression controlled* for the cross-section:

$$\phi = \begin{cases} 0.65 + (\epsilon_t - 0.002)\frac{250}{3} & [0.004 \leq \epsilon_t < 0.005] \\ 0.9, & [\epsilon_t > 0.005] \end{cases} \tag{29}$$

$$M_R = \phi Mn \tag{30}$$

4.1.4 Function: SGD1tBeamsISR

Purpose: to determine an optimal reinforcement area for a given beam cross-section with specified initially dimensions (b,h) through the SGD method.

Syntax:

$[c\_best, bestMr, bestEf, best\_area, tbest, h] = SGD1tBeamsISR(b, h, duct, b\_rec, h\_rec, fc, load\_conditions, factor\_fc, E)$

Description:

The optimization function uses the SGD method to determine the optimal t width of the ISR for a beam cross-section, so that the structural efficiency is within a certain range ($[0.8, 0.95]$ by default).

Output arguments:

- $c\_best, bestMr, bestEf, best\_area, tbest$: the neutral axis depth for the optimal design, the resistant bending moment for the optimal design, the optimal reinforcement area, the optimal t width of the ISR
- h: The final cross-section height in case it is modified from the given initial proposal value

Input arguments:

- $duct$: is the ductility demand parameter, with possible values of 1,2 or 3, for low ductility, medium ductility or high ductility respectively
- $load\_conditions = [n\_load, M_u]$ size nloads x 2
- $factor\_fc$ is determined by de applicable design code. The [ACI 318-19] specifies it as 0.85
- b_rec, h_rec is the concrete cover along the width and height cross-section dimension, respectively (cm)
- h, b cross-section dimensions (cm)
- E Elasticity Modulus of steel reinforcement ($\frac{Kg}{cm^2}$)

Theory:

Such efficiency is calculated as (31) where is M_{umax} is the critical bending load and M_R is the resistant bending moment of the reinforced cross-section, determined by function **Efrec2t_beams** (p.**??**).

$$Eff = \frac{M_{umax}}{M_R} \tag{31}$$

The function can modify the initial given cross-section height in case the dimensions are too small to reach an acceptable design within the design reinforcement area limits established by code for any required ductility demand: low ductility, medium ductility or high ductility (32), (33) and (34) respectively, controlled by the *duct* input argument.

Low ductility:

$$\frac{0.7bd}{b-2b_{rec}}\frac{\sqrt{0.85f_c'}}{f_y} \leq t \leq \frac{0.9(0.85f_c')}{(b-2b_{rec})f_y}\frac{bd(6000\beta_1)}{(f_y+6000)} \tag{32}$$

Medium ductility:

$$\frac{0.7bd}{b-2b_{rec}}\frac{\sqrt{0.85f_c'}}{f_y} \leq t \leq \frac{0.75(0.85f_c')}{(b-2b_{rec})f_y}\frac{bd(6000\beta_1)}{(f_y+6000)} \tag{33}$$

High ductility:

$$\frac{0.7bd}{b-2b_{rec}}\frac{\sqrt{0.85f_c'}}{f_y} \leq t \leq \frac{0.025bd}{b-2b_{rec}} \tag{34}$$

4.2 ISR analysis for columns

4.2.1 Function: eficienciaRecISRCols

Purpose: to compute the structural efficiency of an ISR for a column cross-section.

Syntax:

$[maxef, diagramaInteraccion, tablaEficiencias, cxy] = eficienciaRecISRCols(nconditionss, ...$
$diagramaInteraccion, load\_conditions, pot, poc, c\_vector)$

Description: The function computes the interaction diagram of a column cross-section idealized with an 1t-ISR and its structural efficiency based on the given load conditions.

Output variables:

- $maxef$ is the critical structural efficiency of the column cross-section given different load conditions
- $diagramaInteraccion$ is the interaction diagram for the reinforced column cross-section
- $tablaEficiencias$ is the resume table of results consisting of $nload\_conditions$ rows and eight columns as $[P_u, M_{ux}, M_{uy}, P_{Rx}, P_{Ry}, M_{Rx}, M_{Ry}, Eff]$
- cxy is a vector containing the neutral axis depth of each cross-section direction according to the most critical load condition as $[cx, cy]$

Input variables:

- poc, pot are the resistant axial load in compression and axial load in tension, respectively, of the reinforced column cross-section
- $load\_conditions$ is the vector consisting of $nload\_combinations$ as rows and four columns corresponding to $[nload, P_u, Mu_x, Mu_y]$
- $diagramaInteraccion$ is the array containing the points of the interaction diagram for both axis directions of the column cross-section as $(P_u,\ Mu_{axis})$ determined in the **function** $width\_efficiency$ (p. 41)
- c_vector is an array consisting of n interaction diagram points as rows and two columns. Each combination $[cx_i, cy_i]$ corresponds to each point of the interaction diagram which is determined with the **function** $bisectionMr4t$ (p. 38)

Theory:

The structural efficiency Ef is computed using the Bresler's formula with the *inverse load method* (68) for axial load values in which $\frac{P_u}{P_{oc}} \geq 0.1$ is complied, so that $Ef = [\frac{P_n}{P_{oc}} \leq 1.0]$.

$$\frac{1}{P_n} = \frac{1}{P_{Rx}} + \frac{1}{P_{Ry}} - \frac{1}{P_{ot}} \tag{35}$$

On the other hand, for small values of the axial load such that $\frac{P_u}{P_{oc}} \leq 0.1$ is complied, the *Contour load method* applies through the *bidirectional interaction equation* (69):

$$Ef = \frac{M_{nx}}{M_{Rx}} + \frac{M_{ny}}{M_{Ry}} \leq 1.0 \tag{36}$$

Analytic geometry is applied for the determination of the resistance corresponding to each load condition P_R, M_{Rx}, M_{Ry} (see **Fig. 27**) as (70) and (71):

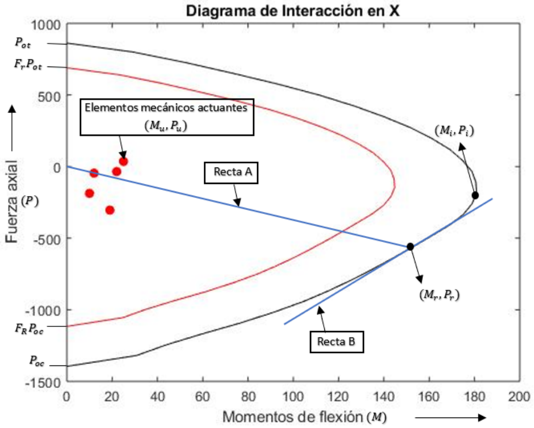


Figure 21: *Interaction diagram in the Cartesian plane as reference for comprehension of the application of analytical geometry for the computation of the structural efficiency of reinforced column cross-sections.*

$$M_r = \frac{P_{i+1} + (\frac{P_i - P_{i+1}}{M_{i+1} - M_i})}{\frac{P_u}{M_u} - \frac{P_{i+1} - P_i}{M_{i+1} - M_i}} \tag{37}$$

$$P_r = \frac{P_u}{M_u} M_r \tag{38}$$

4.2.2 Function: bisectionMr4t

Purpose: To determine the neutral axis depth and bending moment resistance from the interaction diagram of a reinforced column cross-section given a resistant axial load.

Syntax:

$[raiz] = bisectionMr4t(cUno, cDos, fr, E, t1, t2, t3, t4, h, b, rec, fdpc, beta1, ea)$

Description:

Output variables:

- $raiz$ vector containing the neutral axis depth, axial resistant force and bending resistance of a reinforced column cross-section as $[c, F_R, M_R]$

Input variables:

- $cUno, cDos$ Are the initial neutral axis depth values for the implementation of the *bisection method*, recommended to be close to the values $cUno = 0.0001$ and $cDos = 2h$
- fr is the axial load resistance from which the bending resistance will be determined from the interaction diagram
- $t1, t2, t3, t4$ are the ISR depths of a 4t-ISR (strictly for columns)
- rec is a vector containing the concrete cover for both cross-section axis as $[cover_x, cover_y]$
- $fdpc$ is the reduced value of f'_c with the factor 0.85 as prescribed in the **ACI 318-19** code
- β_1 is determined as following (73) in units Kg, cm

$$0.65 \leq (\beta_1 = 1.05 - \frac{f'_c}{1400}) \leq 0.85 \tag{39}$$

- ea is the approximation root error (close to 0)

Theory:

The root **bisection method** is employed. Given that such method is categorized as *closed*, the initial root values (neutral axis depth values in this case) have to be in both extremes of its more likely value, that is along the cross section height or width. For more info about this method see [Chapra & Canale, 2015].

4.2.3 Function: eleMecanicos4t

Purpose: To determine the axial load and bending resistance of a reinforced column cross section with an ISR.

Syntax:

$eleMec = eleMecanicos4t(c, a, fdpc, h, b, rec, E, t1, t2, t3, t4)$

Description:

Output variables:

- $eleMec$ array containing the sum of resistance forces (axial and being) as $[\sum F_s, \sum M_s; \sum F_{conc}, \sum M_{conc}]$

Input variables:

- c neutral axis value
- b, h cross-section dimensions of column (width and height)
- $t1, t2, t3, t4$ are the ISR depths of a 4t-ISR (strictly for columns)
- rec is a vector containing the concrete cover for both cross-section axis as $[cover_x, cover_y]$
- E Elasticity modulus in units $\frac{Kg}{cm^2}$
- $fdpc$ is the reduced value of f'_c with the factor 0.85 as prescribed in the **ACI 318-19** code
- β_1 is determined as following (73) in units Kg, cm

$$0.65 \leq (\beta_1 = 1.05 - \frac{f'_c}{1400}) \leq 0.85 \tag{40}$$

Theory:

The function considers the location of the Plastic Center (PC) the same as the Geometric Center (GC) (which is at a depth $\frac{h}{2}$) **Fig. 22**, so that the resistant moment is calculated as (41), where $Fs_i = As_i E_y \epsilon_i$ for reinforcement steel.

$$M_R = (\sum Fs_i + \beta_1 ab0.85f'_c)(\frac{h}{2} - d_i) \tag{41}$$

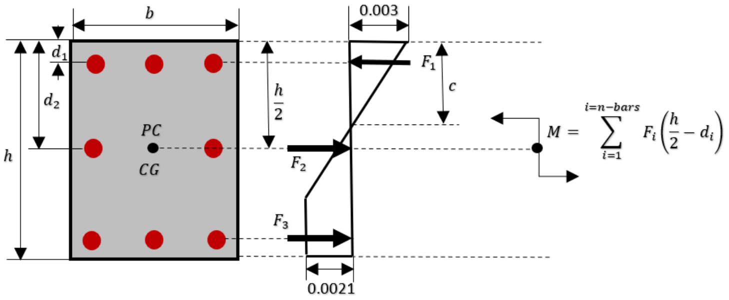

b
d_1
d_2
h
PC
CG
$\frac{h}{2}$
0.003
F_1
c
F_2
F_3
0.0021
$M = \sum_{i=1}^{i=n-bars} F_i\left(\frac{h}{2} - d_i\right)$

Figure 22: *Flexure-compression mechanism of column cross-section.*

4.2.4 Function: widthEfficiencyCols

Purpose: To compute the interaction diagram of an ISR reinforced column cross-section and its structural efficiency given some load conditions.

Syntax:

$[Eft, diagramaInteraccion, tablaEficiencias, cxy] = widthEfficiencyCols(t, ...$
$dimensionesColumna, rec, fy, npuntos, conditions, fdpc, E, beta)$

Description:

Output variables:

- Eft Structural efficiency
- $diagramaInteraccion$ interaction diagram coordinates for both cross-section axis directions
- $tablaEficiencias$ is the resume table of results consisting of $nload\_conditions$ rows and eight columns as $[P_u, M_{ux}, M_{uy}, P_{Rx}, P_{Ry}, M_{Rx}, M_{Ry}, Eff]$
- cxy is a vector containing the neutral axis depth of each cross-section direction according to the most critical load condition as $[cx, cy]$

Input variables:

- c neutral axis value
- $dimensionesColumna$ cross-section dimensions of column (width and height) as $[b, h]$
- t is the ISR width of a 1t-ISR for column cross-sections
- rec is a vector containing the concrete cover for both cross-section axis as $[cover_x, cover_y]$
- E Elasticity modulus in units $\frac{Kg}{cm^2}$
- $fdpc$ is the reduced value of f'_c with the factor 0.85 as prescribed in the **ACI 318-19** code
- β_1 is determined as following (73) in units Kg, cm

$$0.65 \leq (\beta_1 = 1.05 - \frac{f'_c}{1400}) \leq 0.85 \tag{42}$$

- $npuntos$ number of points to be analysed from the interaction diagram

Theory:

The max compression resistance of the column cross-section is determined as (66) where A_c is the concrete net cross-section area and A_s is the total reinforcement area. On the other hand, the max tension resistance is determined as (67).

$$P_{oc} = 0.85 f'_c (A_c - A_s) + f_y (A_s) \tag{43}$$

$$P_{ot} = f_y (A_s) \tag{44}$$

For the computation of the interaction diagram, the resistance factors of $F_R = 0.65$ and $F_R = 0.75$ are applied respectively for a compression controlled and tension controlled cross-section case, according to the **ACI 318-19** and **NTC-17** codes. **Fig. 23**.

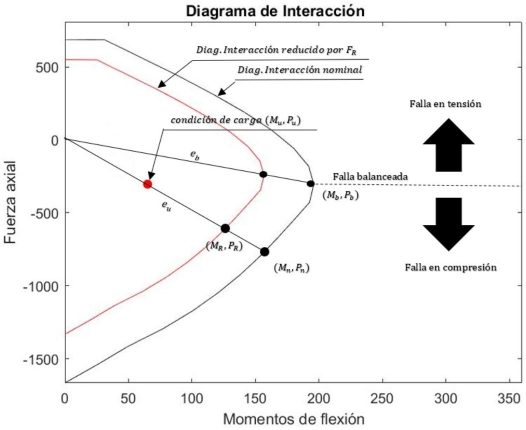


Figure 23: *Interaction diagram of reference.*

4.2.5 Function: isrColumns

Purpose: To determine an optimal ISR for a column cross-section given certain load conditions through the SGD method.

Syntax:

$[b, h, cost_elem\_col, Ac\_sec\_elem, Ef\_sec\_col, Mr\_col, t\_value\_x, t\_value\_y, cxy] = ...isrColumns(pu\_cols, ...$
$height, b, h, rec, fy, fc, load\_conditions, ductility, optimaConvPlot, plotISRResults)$

Description:

Output variables:

- b, h are the final cross-section dimensions in case of a need of modification to comply with the restrictions criteria
- $cost\_elem\_col$ is the total construction cost of the element, considering both concrete and reinforcing steel
- Ac_sec_elem is the optimal reinforcement area
- Ef_sec_col is the optimal structural efficiency for the cross-section
- Mr_col are the resisting moments for both axis directions of the column cross-section: $[Mr_x, Mr_y]$
- t_value_x is the optimal ISR width t in the x-axis of the cross-section
- t_value_y is the optimal ISR width t in the y-axis of the cross-section
- cxy neutral axis depth of optimal design for both axis axis directions of the column's cross-section, corresponding to the critical load condition: $[cx, cy]$

Input variables:

- b, h initial given cross-section dimensions
- $fdpc = 0.85f'_c$: according to [ACI 318-19]
- β_1 is determined as following (73) in units Kg, cm

$$0.65 \leq (\beta_1 = 1.05 - \frac{f'_c}{1400}) \leq 0.85 \tag{45}$$

- $load_conditions$ are the load conditions applied to a cross-section: size = $[nloads, 4]$ as: $[load, Pu, Mx, My]$
- $ductility$ parameter than indicates the ductility demand: (1),(2),(3) for low, medium and high ductility
- rec is the concrete cover for both axis directions: $[cover_x, cover_y]$
- $optimaConvPlot$ is the parameters that indicates if the optima convergence plot is required or not. Option are: (1) the plot is required, (2) they plot is not required

- *plotISRResults* is the parameters that indicates if the ISR interaction diagrams are required or not. Options are: (1) they are required, (2) they are not required

Theory:
The structural efficiency is calculated with thefunction **eficienciaRec_ISR_Cols** (p.**??**).

The function can modify the initial given cross-section height in case the dimensions are too small to reach an acceptable design within the design reinforcement area limits established by code for any required ductility demand: low ductility, medium ductility or high ductility (46) and (47) respectively, controlled by the *duct* input argument.

Low and Medium ductility:

$$\frac{0.01bh}{2(b-2b_{rec})+2(h-2h_{rec})} \leq t \leq \frac{0.06bh}{2(b-2b_{rec})+2(h-2h_{rec})} \tag{46}$$

High ductility:

$$\frac{0.01bh}{2(b-2b_{rec})+2(h-2h_{rec})} \leq t \leq \frac{0.04bh}{2(b-2b_{rec})+2(h-2h_{rec})} \tag{47}$$

4.3 ISR analysis for isolated footings

4.3.1 Function: EvaluateISR1tFoot

Purpose: To determine the structural efficiency of a footing transversal cross-section subject to pure flexure.

Syntax:

$[maxef, mr] = EvaluateISR1tFoot(t_tension, b, h, ...$
$fy, fdpc, rec, beta1, axis, mu_real_axis)$

Description:

Output variables:

- $maxef$ is the structural efficiency of the reinforced footing transversal cross-section as $maxef = \frac{M_u}{M_R}$
- mr is the resistant bending moment of the reinforced footing transversal cross-section M_R

Input variables:

- h is the cross-section height
- *t_tension* is the ISR width
- b is the width of the footing transversal cross-section in analysis
- rec is the concrete cover
- β_1 is determined as following (73) in units Kg, cm

$$0.65 \leq (\beta_1 = 1.05 - \frac{f'_c}{1400}) \leq 0.85 \tag{48}$$

- $axis$ is the footing axis direction of analysis: (1) represents the axis direction in which the dimension L is the width of the transversal cross-section, for (2) B is the width of the transversal cross-section, in its reference system (see **Fig. 20**)
- fy is the yield stress of reinforcing steel
- *mu_real_axis* is the effective flexure distributed to the transversal cross-section from the contact soil pressures
- $fdpc$ is the reduced f'_c as $fdcp = 0.85f'_c$ according to code

Theory:

This function does not consider the steel in compression, given that for footings, this quantity is usually set as the minimum by temperature, therefore its resistance contribution is almost negligible. Thus, the resistant moment is calculated through a beam mechanism as (49), and the structural efficiency by $\frac{M_u}{M_R}$, where 0.9 represents the resistance reduction factor by code.

$$M_R = 0.9(T(d - \frac{a}{2})) \tag{49}$$

4.3.2 Function: SGD1tFootISR

Purpose: To determine the optimal steel area reinforcement of a footing transversal cross-section subject to uniaxial flexure.

Sintax:

$[pbest, bestEf, bestMr, best\_area] = SGD1tFootISR(b, h, ...$
$rec, fdpc, fy, steelAreaRange, betac, axis, mu\_real\_axis, plotOptimConv)$

Description:

Output variables:

- *pbest* is the optimal average reinforcement area percentage
- $bestEf$ is the structural efficiency of the optimal reinforcement option
- $bestMr$ is the resisting bending moment of the optimal reinforced cross-section
- *best_area* is the optima reinforcing area of the transversal cross-section

Input variables:

- *axis* is the footing axis direction of analysis: (1) represents the axis direction in which the dimension L is the width of the transversal cross-section, for (2) B is the width of the transversal cross-section, in its reference system (see **Fig. 20**)
- *mu_real_axis* is the effective bending moment load applied to the cross-section in question
- *plotOptimConv* is the parameter that indicates if the optima ISR convergence plot is required or not. Options are: (1) they are required, (2) they are not required

Theory:

The Steepest Gradient Descent method is used **Algorithm 2.1**. [Wahde, 2008]

5 Element rebar optimization functions

5.1 Rebar analysis for beams

5.1.1 Function: ISR1tRebarBeamsOptimization

Purpose: to design optimally a rebar distribution over a beam cross-section given the ISR.

Syntax:

$[sepbarsRestric, cbest, b, h, bestBarDisposition, bestCost, arrangement\_t1, arrangement\_t2, ...$
$maxEf, bestMr, area\_var\_t] = ISR1tRebarBeamsOptimization(E, b, h, fy, fc, b\_rec, ...$
$h\_rec, tma, conditions, t2, pu\_beams)$

Description:

Output variables:

- $sepbarsRestric$ is the parameter that indicates if the minimum rebar separation restriction of rebars in tension for the cross-section in question is being complied: (1) indicates that such restriction is not being complied, (0) indicates that such restriction is being complied
- $cbest$ is the depth of the neutral axis for the optimized beam reinforced cross-section considering the optimal rebar design
- b, h are the final cross-section dimensions in case they suffered modifications after the optimal design process
- $bestBarDisposition$ are the local coordinates of rebar disposition over the optimal designed cross-section
- $arrangement\_t1$ are the list of rebar type transformed from the ISR in tension: a vector consisting of one column of length $nbars$ in tension
- $arrangement\_t2$ are the list of rebar type transformed from the ISR in compression: a vector consisting of one column of length $nbars$ in compression
- $maxEf$ is the optimal final structural efficiency for the optimal designed beam cross-section considering the optimal rebar
- $bestMr$ is the optimal final bending resistance for the optimal designed beam cross-section considering the optimal rebar
- $area\_var\_t$ is a vector consisting of the total optimal rebar area in tension and compression (it can be considered later for the assessment of modification of inertia as a cracked section)

Input variables:

- E is the Elasticity Modulus of steel in $\frac{Kg}{cm}$
- fy is the yielding stress in $\frac{Kg}{cm}$
- $t2$ is the optimal ISR consisting of vector of two elements $[t\_tension, t\_compression]$
- pu_beams is the unitary cost of rebar assembly in beams considering an average of assembly performance (it is considered that various types of rebars are placed simultaneously in the beam element along its length)

Theory:

The design of its respective rebar are must be greater or equal than the ISR required area, to ensure that the required structural efficiency complies.

5.1.2 Function: RebarOptimalDesignBeams

Purpose: To determine an optimal rebar arrangement based on minimum area for a beam cross-section.

Syntax:

$[sepbarsRestric, cbest, b, h, maxEf, bestMr, area\_var\_t, nv\_t, arreglo\_t1, arreglo\_t2, list\_pac\_t1, ...$
$list\_pac\_t2, disposition\_rebar] = RebarOptimalDesignBeamsl(b, h, b_rec, h_rec, ...$
$sepMin, varDisponibles, t2, fdpc, E, fy, condiciones, betac)$

Description: The function employs a simple search algorithm to determine the rebar option with the less reinforcement area. The program has an alternative option of displaying rebars in two packs in case the minimum rebar separation restriction does not comply for rebars displayed individually. The function does not modify any of the beam cross-section dimensions in case the minimum separation restriction is not complied by any rebar option. At the end, the function computes the bending resistance and structural efficiency given the load conditions.

Output variables:

- $sepbarsRestric$ is the parameter that indicates if the minimum rebar separation restriction is being complied for the rebars in tension of the beam cross-section in question: (1) indicates the restriction is not being complied, (2) indicates such restriction is being indeed being complied
- $cbest$ Neutral axis depth for the beam reinforced cross-section with the optimum rebar option
- $maxEf$ Structural efficiency with the optimal rebar option, less than 1.0
- $bestMr$ Resistant bending moment with the optimal rebar option
- $area\_var\_t$ Vector consisting of the optimal rebar area in tension and compression, stated as $[arebar\_tension, arebar\_compression]$
- nv_t Vector consisting of the number of rebars in tension and compression, stated as $[nrebar\_tension, nrebar\_compression]$
- $arreglo\_t1, arreglo\_t2$ Vectors that contain the type of rebar for the optimal option both in tension and compression, respectively. The vectors size is of one column with nrebar rows containing a number between 1 and 7 according to the available commercial rebar types stated by default
- $list\_pac\_t1, list\_pac\_t2$ Vectors that contain a value either 1 or 2, which indicate the manner in which a rebar is displayed over the cross-section. Number 1 means that the rebar is laid out at the outer horizontal layer. Number 2 means that the rebar is laid out at the inner horizontal layer (see **Fig. 24**)
- $disposition\_rebar$ local coordinates of the optimal rebar distribution over the beam cross-section

Input variables:

- $varDisponibles$ Commercial available type of rebars database as indicated in **function** $ISR1t\_RebarBeams\_Optimization\_function$ (p.**??**)

Theory:

The rebar layout options are as shown next.

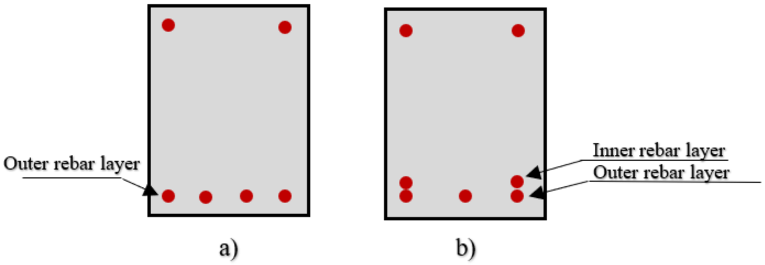


Figure 24: *Reinforcement bar options for a beam cross-section: a) in one pack, b) in two packs.*

The estimation of the minimum rebar separation constraint is determined as the max of (50):

$$sep_{min} = \begin{cases} 1in(2.54cm) \\ \frac{4}{3}d_{ag}, d_{ag} = \frac{2}{3}in \\ diam_{bar} \end{cases} \tag{50}$$

5.1.3 Function: EfcriticalRebarbeams

Purpose: To compute the bending resistance, structural efficiency and depth of neutral axis for the optimal reinforced beam cross-section with rebars.

Syntax:
$[maxef, Mrv, c] = EfcriticalRebarbeams(load\_conditions, b, E, fdpc, arrange\_t1, arrange\_t2, ...$
$rebarAvailable, d, h\_rec, beta1, disposition\_rebar)$

Description:

Output variables:

- $maxef$ structural efficiency for the optimal reinforced cross-section with rebars
- Mrv Resistant bending moment for the optimal reinforced cross-section with rebars
- c neutral axis depth for the optimal reinforced cross-section with rebars

Input variables:

- $fdpc$ reduced f'_c as $0.85f'_c$ according to the [ACI 318-19] code
- β_1 is determined as following (73) in units Kg, cm

$$0.65 \leq (\beta_1 = 1.05 - \frac{f'_c}{1400}) \leq 0.85 \tag{51}$$

- $disposition\_rebar$ local coordinates of rebars laid out over the beam cross-section
- $arrange\_t1, arrange\_t2$ Vectors that contain the type of rebar for the optimal option both in tension and compression, respectively. The vectors size is of one column with nrebar rows containing a number between 1 and 7 according to the available commercial rebar types stated by default

Theory:
Once the equilibrium condition is complied, with its respective neutral axis and nominal resistant bending moment with the **function bisectionMrRebarBeams** (p.52), the resistance reduction factor ϕ (52) is then applied as (53) according to the condition of *tension controlled* or *compression controlled* for the cross-section:

$$\phi = \begin{cases} 0.65 + (\epsilon_t - 0.002)\frac{250}{3} & [0.004 \leq \epsilon_t < 0.005] \\ 0.9, & [\epsilon_t > 0.005] \end{cases} \tag{52}$$

$$M_R = \phi Mn \tag{53}$$

5.1.4 Function: bisectionMrRebarBeams

Purpose: To determine the neutral axis depth and resistant bending moment of a reinforced beam cross-section taking on account the distribution of rebars over the cross-section with the aid of the bisection method as a root for the pre-established equilibrium condition $\sum F = 0$

Syntax:
$[raiz] = bisectionMrRebarBeams(c1, c2, fr, E, h, b, h\_rec, fdpc, beta, ea, arreglo\_t1, ...$
$arreglo\_t2, disposicion\_varillado, rebarAvailable)$

Description:

Output variables:

- $raiz$ vector that contains the neutral axis depth c, the sum of axial forces of equilibrium $\sum F_R = 0$ and the resistant bending moment a $[c, \sum F_R, M_R]$

Input variables:

- $c1, c2$ initial root values for the use of the bisection method. As a closed root method, it is recommended to use $c1 = 1x10^{-6}$ and $c2 = 2h$
- $arreglo\_t1, arreglo\_t2$ Vectors that contain the type of rebar for the optimal option both in tension and compression, respectively. The vectors size is of one column with nrebar rows containing a number between 1 and 7 according to the available commercial rebar types stated by default
- $disposicion\_varillado$ local coordinates of rebars laid out over the beam cross-section
- $rebarAvailable$ data base of the available commercial types of rebar

Theory: In order to calculate the resistant bending moment, the equilibrium condition of forces $\sum F = 0$ must be complied. For this function, it assumed that the acting axial load on the beam cross-section is too small and can be neglected. This axial load should be evaluated by the user, such that such axial load is smaller than the tenth part of the cross-section axial load resistance $P_{oc} < \frac{1}{10}(bh - A_s)f'_c$ according to the [ACI 318-19] code and other international codes.

The nominal resistant bending moment is determined as (26) for a given neutral axis depth value c. The neutral axis depth is restricted by a ductility strain requirement established by code **ACI 318-19** as (27). For more reference of the bisection method see [Chapra & Canale, 2015]

5.1.5 Function: eleMecanicosRebarBeams

Purpose: to compute the sum of resistant forces of a beam cross-section, considering the contribution of rebars over the cross-section and concrete in compression.

Syntax:

$eleMec = eleMecanicosRebarBeams(c, a, fdpc, h, b, h_rec, E, arreglo_t1, ...$
$arreglo_t2, disposition_rebar, rebarAvailable)$

Description:

Output variables:

- eleMec: vector that contains the output $[\sum F_s, \sum M_s; F_c, M_c]$

Input variables:

- $arreglo\_t1, arreglo\_t2$ Vectors that contain the type of rebar for the optimal option both in tension and compression, respectively. The vectors size is of one column with nrebar rows containing a number between 1 and 7 according to the available commercial rebar types stated by default
- $disposition_rebar$ local coordinates of rebars laid out over the beam cross-section
- $rebarAvailable$ data base of the available commercial types of rebar
- $fdpc = 0.85f'_c$: according to [ACI 318-19]

Theory:

The function considers the location of the Plastic Center (PC) the same as the Geometric Center (GC) (which is at a depth $\frac{h}{2}$), so that the resistant moment is calculated as (24), where $Fs_i = As_iE_y\epsilon_i$ for reinforcement steel.

5.1.6 Function: RebarDisposition1tBeams

Purpose: To compute the local coordinates of a rebar option according to its given data (number of rebars in tension and compression, cross-section dimensions, type of rebar in tension and compression and type of arrangement - two pack or one pack).

Syntax:

$[disposicion\_varillado] = RebarDisposition1tBeams(b, h, b\_rec, h\_rec, ...$
$varDisponibles, nv\_t, arreglo\_t1, arreglo\_t2, list\_pac\_t1, list\_pac\_t2, M\_u)$

Description:

Output variables:

- $disposicion\_varillado$ local coordinates of rebars laid out over the beam cross-section

Input variables:

- nv_t vector containing the number of rebars in tension and compression as $[nrebar\_tension, nrebar\_comp]$
- $arreglo\_t1, arreglo\_t2$ Vectors that contain the type of rebar for the optimal option both in tension and compression, respectively. The vectors size is of one column with nrebar rows containing a number between 1 and 7 according to the available commercial rebar types stated by default
- $list\_pac\_t1, list\_pac\_t2$ type of arrangement for each bar, either (1: one pack), (2: two pack). Are vectors of one column and n-rebar rows in tension and compression, respectively
- M_u maximum bending load over the cross-section

5.1.7 Function: EvaluateCostbeams

Purpose: To compute the unit linear cost of rebar assembly for a beam cross-section given an average unit cost in units $\frac{\$}{Kg}$.

Syntax:

$$cost = EvaluateCostbeams(nvHor1, nvHor2, arreglo_t1, arreglo_t2, pu, availableRebar)$$

Description:

Output variables:

- $cost$ unit linear cost of rebar for a beam cross-section in units $(\frac{\$}{cm})$

Input variables:

- $nvHor1, nvHor2$ number of rebars in tension and compression of a beam cross-section, respectively
- $arreglo\_t1, arreglo\_t2$ Vectors that contain the type of rebar for the optimal option both in tension and compression, respectively. The vectors size is of one column with nrebar rows containing a number between 1 and 7 according to the available commercial rebar types stated by default
- pu unit cost $(\frac{\$}{Kg})$ of rebar assembly for a beam cross-section
- $availableRebar$ database of available commercial types of rebar

5.1.8 Function: beamsISR

Purpose: to design optimally all three cross-sections (left, middle and right) along a beam element according to its moment distribution diagram, using the ISR analogy.

Syntax:

$[sepbarsRestric, b, h, inertia\_modif, dispositionBar\_Der, barArrangementDerComp, barArrangementDerTens, ...$
$dispositionBar\_Center, barArrangementCentralTens, barArrangementCentralComp, ...$
$dispositionBar\_Izq, barArrangementIzqTens, barArrangementIzqComp, ...$
$minAreaVar\_3sec, Ef\_elem\_sec\_t, bestCostVar, ef\_var, minAreaVar\_prom, Mr\_3section] = ...$
$beamsISR(pu_beams, span, b, h, h\_rec\_sections, fc, fy, condiciones, cols\_sym\_asym\_isr, duct, b\_rec, plots, ...$
$graphConvergencePlot)$

Description:

Output variables:

- $sepbarsRestric$ is the parameter that indicates if the rebar separation constraint of rebars in tension for each of the beam's cross-section is being complied: (1) means the restriction is not being complied in the design, (0) means the restriction is being complied
- $inertia_modif$ is the modified cross-section inertia based on a cracking mechanism for each of the designed cross-sections
- b, h are the final cross-section dimensions in case they suffered modifications after the optimal design process
- $dispositionBar\_Der$ are the local coordinates of rebar disposition over the optimal designed right cross-section
- $dispositionBar\_Center$ are the local coordinates of rebar disposition over the optimal designed central cross-section
- $dispositionBar\_Izq$ are the local coordinates of rebar disposition over the optimal designed left cross-section
- $arrangement\_t1$ are the list of rebar type transformed from the ISR in tension: a vector consisting of one column of length $nbars$ in tension
- $barArrangementDerComp, barArrangementDerTens$ are the list of rebar type transformed from the ISR in compression and tension, respectively, for the optimally designed right cross-section: a vector consisting of one column of length $nbars$ in compression and tension
- $barArrangementCentralComp, barArrangementCentralTens$ are the list of rebar type transformed from the ISR in compression and tension, respectively, for the optimally designed central cross-section: a vector consisting of one column of length $nbars$ in compression and tension
- $barArrangementIzqComp, barArrangementIzqTens$ are the list of rebar type transformed from the ISR in compression and tension, respectively, for the optimally designed left cross-section: a vector consisting of one column of length $nbars$ in compression and tension
- ef_var is the optimal final structural efficiency for each of the three optimal designed beam cross-sections considering the optimal rebar

- *Mr_3section* is the optimal final bending resistance for each of the three optimal designed beam cross-sections considering either the optimal rebar or the optimal ISR, according to the user preferences
- *bestCostVar* is the total final cost of reinforcement considering the three cross-section reinforcement as an average rebar area along the total span length of the beam element
- *minAreaVar_prom* is the average rebar area of all three cross-section (sum of steel in tension and compression)

Input variables:

- *fc* is the concrete f'_c in units $\frac{Kg}{cm^2}$
- *fy* is the reinforcement steel yielding stress in $\frac{Kg}{cm}$
- *load_conditions* are the load conditions for all three cross-sections: a vector consisting of one row and four columns as $[1 Mu_{left} Mu_{mid} Mu_{right}]$ in units $Ton - m$. The sign of each load should be included
- *pu_beams* is the unitary cost of rebar assembly in beams considering an average of assembly performance (it is considered that various types of rebars are placed simultaneously in the beam element along its length)

  span is the span length of the beam element

  b, h are the initial cross-section dimensions of the beam element

  h_rec_sections are the concrete height cover for each of the three cross-sections as a vector of six columns and one row as:

  $[rec_{l}eft_{u}p, rec_{l}eft_{l}ow, rec_{m}id_{u}p, rec_{m}id_{l}ow, rec_{r}ight_{u}p, rec_{r}ight_{l}ow]$
- *cols_sym_asym_isr* is the optimal design option: either "ISR" (when no optimal rebar design is required, but only the ISR) or "Standard" (when a rebar optimal design is required)
- *duct* is the ductility demand level of design: 1 - low ductility, 2 - medium ductility, 3 - high ductility
- *b_rec* is the lateral concrete cover of the beam element as: [*b_cover*] (as it is the same value for all three cross-sections)
- *plots* is the parameter that indicates if the plotting of results are required. Options are: (1) they are required, (2) they are not required
- *graphConvergencePlot* is the parameter that indicates if the plotting of optima convergence is required. Options are: (1) they are required, (2) they are not required

Theory:

The function encompasses all beam functions mentions in this section and applies them for the three main cross-sections of interest in a beam element along its length. The free-clash reinforcement criteria that the function considers is according to **Fig. 25**, where rec_{min} is the minimum height concrete cover and dv_{12} is the diameter of a #12 rebar type.

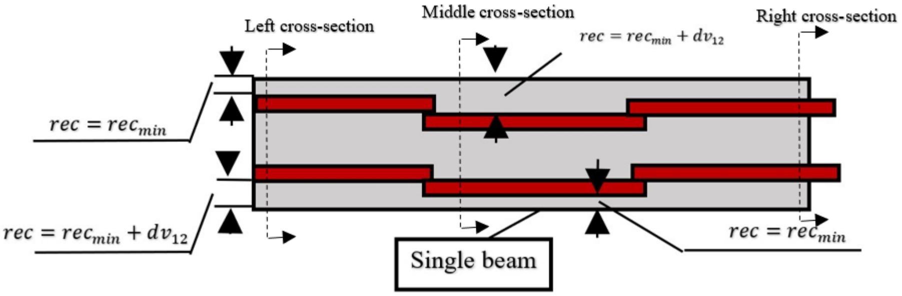


Figure 25: *Free-clash reinforcement criteria for a beam element.*

5.1.9 Function: ExportResultsBeam

Purpose: .

Syntax:

$ExportResultsBeam(directionData, dim\_beams\_collection, coordEndBeams, ...$
$disposition\_rebar\_beams3sec, nbarbeamsCollection, arrangemetbarbeams)$

Description: Computes the exportation of the design results of a beam element into a .txt file on a prescribed folder route.

Input variables:

- *directionData* is the folder disc location to save the results
- *dim_beams_collection* is the array containing the cross-section dimensions data of the beam element
- *coordEndBeams* is the array containing the coordinates of the initial end's cross-section centroid of the beam
- *disposition_rebar_beams3sec* is the array containing the local rebar coordinates of each of the three critical design cross-sections of the beam
- *nbarbeamsCollection* is the total number of rebars of each of the three design cross-sections, both in tension and compression. Size = $[1, 6]$ in format:

$$[nbarsLeft_{ten}, nbarsLeft_{com}, nbarsCenter_{ten}, nbarsCenter_{com}, nbarsRight_{ten}, nbarsRight_{com}]$$

- *arrangemetbarbeams* is the list of all the rebar types used in the element

---

5.2 Rebar analysis for columns

5.2.1 Function: ExportResultsColumn

Purpose: Computes the exportation of the design results of a column element into a .txt file on a prescribed folder route.

Syntax:

$ExportResultsColumn(directionData, dimColumnsCollection, ...$
$bestdisposicionRebar, nbarColumnsCollection, bestArrangement, coordBaseCols)$

Description:

Input variables:

- *directionData* is the folder disc location to save the results
- *dimColumnsCollection* is the array containing the cross-section dimensions data of the column element
- *coordBaseCols* is the array containing the coordinates of the column base cross-section's centroid
- *bestdisposicionRebar* is the array containing the local rebar coordinates of the column cross-sections
- *nbarColumnsCollection* is the total number of rebars of column cross-sections, both in tension and compression
- *bestArrangement* is the list of the rebar types used in the element

5.2.2 Function: isrColumnsSymAsym

Purpose: To determine an optimal reinforcement design, either with a pure ISR or with symmetric rebar.

Syntax:

$[Inertia\_xy\_modif, b, h, bestArrangement, best\_disposicion, cost\_elem\_col, ...$
$Ac\_sec\_elem, Ef\_sec\_col, Mr\_col] = isrColumnsSymAsym(pu\_cols, height, b, h, rec, fy, ...$
$fc, load\_conditions, cols\_sym\_asym\_isr, condition\_cracking, ductility, optimPlot, plotISRdiagram, plotRebarDesign)$

Description:

Output variables:

- $Inertia\_xy\_modif$ momentum of inertia of the optimal reinforced cross-section for both axis directions as $[Ix, Iy]$ considering the reinforcement with cracked or non-cracked section mechanisms
- b, h are the final cross-section dimensions in case of a need of modification to comply with the restrictions criteria
- *bestArrangement* is the list of rebar types for each rebar of the optima reinforcement option: size = $[nbars, 1]$ consisting of a number from 1 to 7 by default
- *best_disposicion* is the array consisting of the local rebar coordinates over the cross-section for the optimal rebar option
- *cost_elem_col* is the cost of the optimal rebar option (only steel is considered)
- *Ac_sec_elem* is the optimal rebar area
- Ef_sec_col is the critical structural efficiency for the optimal reinforcement option
- Mr_col are the resisting moments for both axis directions of the cross-section as $[M_{Rx}, M_{Ry}]$

Input variables:

- b, h initial given cross-section dimensions
- *pu_cols* is the database of reinforcement assembly and construction unit cost: format by default *pu_col* = $[PU_{\#4}, PU_{\#5}, PU_{\#6}, PU_{\#8}, PU_{\#9}, PU_{\#10}, PU_{\#12}]$;
- *height* is the total length of the column
- f'_c compressive concrete strength
- fy yield strength of reinforcement bars
- *load_conditions* load condition array: size = $[nloads, 4]$, in format $[nload, Pu, Mux, Muy]$
- *ductility* is demand ductility parameters, options are (1) for low ductility section requirements,(2) for medium ductility, (3) for high ductility
- *rec* are the concrete cover values for both cross-section axis directions: $[cover_x, cover_y]$

- $cols_sym_asym_isr$ is the reinforcement option parameters, options are: $"Symmetric"$ or $"ISR"$
- $condition_cracking$ is the cracking mechanisms to be considered, options are: $"Cracked"$ or $"Non-cracked"$
- $optimPlot$ is the parameters that indicates if the optima rebar area convergence is required or not. Options are: (1) they are required, (2) they are not required
- $plotsISRdiagrams$ is the parameters that indicates if the optima ISR interaction diagrams are required or not. Options are: (1) they are required, (2) they are not required
- $plotRebarResults$ is the parameters that indicates if the rebar design results are required or not. Options are: (1) they are required, (2) they are not required

Theory:

Given that the reinforcement is symmetric (also for ISR reinforcement), it is considered that the Plastic Center (PC) of the section is located at the same place of the Geometric Center (GC). The **function** *isr_columns* is used for determination of the optimal ISR (see p. **??**, then if symmetric rebar reinforcement is required the **function** *optimalrebar_cols_sym* is used. For both cases (ISR or $Symmetric$) the **function** $CrackingColumnsSym$ is used to transform the cross-section inertia considering the steel reinforcement (weather if it is a $Cracked$ section or a $Non-cracked$ one).

5.2.3 Symmetric reinforcement

5.2.4 Function: optimalrebarColsSym

Purpose: To determine an optimal rebar design.

Syntax:

$[Mr\_col, h, Inertia\_xy\_modif, bestArea, lowestCost, ovMostEc, nvEc, ...$
$maxEfEc, bestArrangement, best\_disposicion] = optimalrebarColsSym(b, h, ...$
$rec, act, sepMin, E, npuntos, fdpc, beta1, pu\_col\_sym, load\_conditions, condition\_cracking, plotRebarDesign)$

Description:

Output variables:

- Mr_col are the final resistant bending moment for both axis directions of the optimal designed cross-section
- h modified cross-section height in case it is modified through the optimization process to comply the given restrictions of min separation of rebars
- $Inertia\_xy\_modif$ momentum of inertia of the bar reinforced cross-section for both axis directions, by the computation of the cracking mechanisms according to the parameter $condition\_cracking$
- $bestArea$ is the optimal rebar area
- $lowestCost$ is the cost of the optimal design option
- $ovMostEc$ is the type of rebar corresponding to the optimal rebar design option
- $nvEc$ is the total number of rebars over the cross-section corresponding to the optimal design option
- $maxEfEc$ is the critical structural efficiency corresponding to the optimal most economic design option $maxEfEc < 1.0$
- $bestArrangement$ is the list of rebar type of each rebar: size $[nbars, 1]$ (a number from 1 to 7 by default)
- $best\_disposicion$ is an array containing the local coordinates of position of each rebar over the cross-section corresponding to the optimal rebar design option

Input variables:

- rec concrete cover of cross-section for both axis direction: $[cover_x, cover_y]$
- act optima ISR reinforcement area
- $sepMin$ min separation of rebars constraint
- E Elasticity Modulus of reinforcement steel $E = 2.0x10^6 \frac{Kg}{cm^2}$
- $npntos$ number of points to compute for the interaction diagram
- $load\_conditions$ load conditions for the column cross section: size $= [nload, 4]$ in format $[nload, P_u, Mu_x, Mu_y]$
- $fdpc$ is the f'_c reduced with the factor 0.85 according to the **ACI 318-19** code

- *beta*1 is determined as following (73) in units Kg, cm

$$0.65 \leq (\beta_1 = 1.05 - \frac{f'_c}{1400}) \leq 0.85 \tag{54}$$

- *pu_col* is the database of reinforcement assembly and construction unit cost: format by default $pu\_col = [PU_{\#4}, PU_{\#5}, PU_{\#6}, PU_{\#8}, PU_{\#9}, PU_{\#10}, PU_{\#12}]$;

- *condition_cracking* parameter that indicates which cross-section cracking mechanism will be consider, either $Cracked$ or $Non - cracked$. If the condition $Non - cracked$ is set, then the cracking mechanism will be neglected by all means

- *plotRebarDesign* is the parameters that indicates if the rebar design results are required or not. Options are: (1) they are required, (2) they are not required

Theory:
The optimization process is based on simple search or exhaustive search, given the limited number of possibilities for reinforcement with only one type of rebar allowed. The restriction to accept or not a design is based entirely on the rebar separation constraint, taken as the greater of (55)

$$sep_{min} = \begin{cases} \frac{3}{2}d_b \\ \frac{4}{3}d_{ag}, d_{ag} = \frac{3}{4}in \\ 4cm \end{cases} \tag{55}$$

5.2.5 Function: RebarDisposition

Purpose: To compute the local position coordinates of a symmetric rebar design option.

Syntax:

$[dispositionRebar] = disposicionVarillado(b, ...$
$h, rec, dv, nv, varCos, varSup);$

Description:

Output variables:

- *dispositionRebar* are the local position coordinates of the symmetric rebar option

Input variables:

- b, h given cross-section dimensions
- dv, nv are the rebar diameter of the current option, the number of rebars
- $varCos, varSup$ are the number rebars vertically of the cross-section (along the cross-section h height dimension) and the number of rebars horizontally (along the cross-section b width dimension), respectively
- rec is the concrete cover for both cross-section axis directions: $[cover_x, cover_y]$

5.2.6 Function: diagramasDisposicion

Purpose: To compute the interaction diagram of a symmetric rebar option, as well as the structural efficiency given certain load conditions.

Syntax:

$[diagrama, mexef, eficiencia, cxy] = diagramasDisposicion(As, b, h, E, npuntos, ...$
$fdpc, nv, beta, ov, av, disposicion\_varillado, load\_conditions)$

Description:

Output variables:

- $diagrama$ is the interaction diagram data
- $maxef$ is the critical structural efficiency corresponding to the critical load condition
- $eficiencia$ is a table containing the structural efficiency analysis data: size = $[nload, 8]$, in format: $[nload, P_u, Mu_x, Mu_y, P_R$
- cxy are the neutral axis depth values corresponding to the critical load condition, for both axis directions: $[c_x, c_y]$

Input variables:

- b, h are the cross section dimensions of the column
- av, ov, nv are the rebar area and number of rebar of the current rebar option, and the number of rebars, respectively
- $fdpc$ equal to $0.85f'_c$ according to code
- $load\_conditions$ is the array containing the load conditions: size = $[nload, 4]$ in format $[nload, Pu, Mux, Muy]$
- β is determined as following (73) in units Kg, cm

$$0.65 \leq (\beta_1 = 1.05 - \frac{f'_c}{1400}) \leq 0.85 \tag{56}$$

Theory:

The max compression resistance of the column cross-section is determined as (66) where A_c is the concrete net cross-section area and A_s is the total reinforcement area. On the other hand, the max tension resistance is determined as (67).

$$P_{oc} = 0.85f'_c(A_c - A_s) + f_y(A_s) \tag{57}$$

$$P_{ot} = f_y(A_s) \tag{58}$$

For the computation of the interaction diagram, the resistance factors of $F_R = 0.65$ and $F_R = 0.75$ are applied respectively for a compression controlled and tension controlled cross-section case, according to the **ACI 318-19** and **NTC-17** codes. **Fig. 23**.

5.2.7 Function: eficienciaRec

Purpose: To compute the structural efficiency of a symmetrically reinforced column cross-section.

Syntax:

$[maxef, eficiencia, cxy] = eficienciaRec(diagrama, ...$
$load\_conditions, pot, poc, c\_vector\_bar)$

Description: .

Output variables:

- $maxef$ is the critical structural efficiency of the column cross-section given different load conditions
- $eficiencia$ is the resume table of results consisting of $nload\_conditions$ rows and eight columns as:

$$[P_u, M_{ux}, M_{uy}, P_{Rx}, P_{Ry}, M_{Rx}, M_{Ry}, Eff]$$

- cxy is a vector containing the neutral axis depth of each cross-section direction according to the most critical load condition as $[cx, cy]$

Input variables:

- $diagrama$ is the interaction diagram data
- $load\_conditions$ is the array containing the load conditions: size = $[nload, 4]$ in format $[nload, Pu, Mux, Muy]$
- pot, poc are the max resistant axial force in tension of reinforcement steel (concrete is not considered) and compression of the whole reinforced cross-section area (both concrete area and rebar area are considered)
- c_vector_bar is the array containing the neutral axis depth values for both cross-section axis directions for all interaction diagram points: size = $[npoints + 2, 2]$

Theory:

The structural efficiency Ef is computed using the Bresler's formula with the *inverse load method* (68) for axial load values in which $\frac{P_u}{P_{oc}} \geq 0.1$ is complied, so that $Ef = [\frac{P_n}{P_{oc}} \leq 1.0]$.

$$\frac{1}{P_n} = \frac{1}{P_{Rx}} + \frac{1}{P_{Ry}} - \frac{1}{P_{ot}} \tag{59}$$

On the other hand, for small values of the axial load such that $\frac{P_u}{P_{oc}} \leq 0.1$ is complied, the *Contour load method* applies through the *bidirectional interaction equation* (69):

$$Ef = \frac{M_{nx}}{M_{Rx}} + \frac{M_{ny}}{M_{Ry}} \leq 1.0 \tag{60}$$

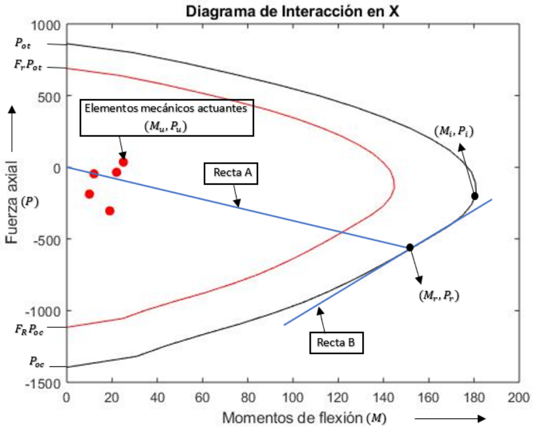


Figure 26: *Interaction diagram in the Cartesian plane as reference for comprehension of the application of analytical geometry for the computation of the structural efficiency of reinforced column cross-sections.*

Analytic geometry is applied for the determination of the resistance corresponding to each load condition P_R, M_{Rx}, M_{Ry} (see **Fig. 27**) as (70) and (71):

$$M_r = \frac{P_{i+1} + (\frac{P_i - P_{i+1}}{M_{i+1} - M_i})}{\frac{P_u}{M_u} - \frac{P_{i+1} - P_i}{M_{i+1} - M_i}} \tag{61}$$

$$P_r = \frac{P_u}{M_u} M_r \tag{62}$$

5.2.8 Function: bisectionMrSymRebarCols

Purpose: To determine the neutral axis depth, axial and bending resistance from the interaction diagram of a reinforced concrete column cross-section for each for its points with the aid of the bisection root method.

Syntax:

$[root] = bisectionMrSymRebarCols(cUno, cDos, fr, E, h, b, fdpc, beta1, ...$
$ea, nv, ov, av, rebar\_disposition)$

Description: .

Output variables:

- $root$ is a vector containing the neutral axis depth, axial resistant force and bending resistance of a reinforced column cross-section as $[c, F_R, M_R]$

Input variables:

- $cUno, cDos$ are the initial values of the neutral axis to commence iterations
- fr is the axial force resistance corresponding to the bending moment resistance for which the equilibrium condition $\sum F = 0$ is established to extract its corresponding bending moment resistance and neutral axis depth from the interaction diagram
- E Elasticity modulus of steel $(4200\frac{Kg}{cm^2})$
- b, h cross-section dimensions
- $fdpc$ is the f'_c reduced with the factor 0.85 according to code
- $\beta 1$ is determined as following (73) in units Kg, cm

$$0.65 \leq (\beta_1 = 1.05 - \frac{f'_c}{1400}) \leq 0.85 \tag{63}$$

- ea is the approximation error to terminate the root bisection method
- nv is the number of rebars to be placed over the cross-section
- ov, av are the type of rebar in eighth of inches $(\#\frac{ov}{8}in)$ and the cross-section area of each rebar in cm^2 equal to $\frac{\pi}{4}(\frac{ov}{8}(2.54))^2$
- $rebar\_disposition$ are the local coordinates of rebars over the cross-section

Theory:

The root bisection method is employed. For more reference of this method, see [Chapra & Canale, 2015].

5.2.9 Function: eleMecanicosRebarCols

Purpose: To compute the sum of resistant forces of a reinforced column cross-section considering the distribution of rebars over the cross-section and concrete zone in compression.

Syntax:

$$[eMecVar] = eleMecanicosRebarCols(disposicion_varillado, nv, ov, av, b, h, c, fdpc, E, beta1)$$

Description: .

Output variables:

- $eMecVar$ vector that contains the output $[\sum F_s, \sum M_s; F_c, M_c]$

Input variables:

- $disposition_rebar$ are the local coordinates of rebars over the cross-section
- nv is the number of rebars
- ov, av are the type of rebar in eighth of inches ($\# \frac{ov}{8} in$) and the cross-section area of each rebar in cm^2 equal to $\frac{\pi}{4}(\frac{ov}{8}(2.54))^2$
- E Elasticity modulus of steel ($4200 \frac{Kg}{cm^2}$)
- b, h cross-section dimensions
- $fdpc$ is the f'_c reduced with the factor 0.85 according to code
- $\beta 1$ is determined as following (73) in units Kg, cm

$$0.65 \leq (\beta_1 = 1.05 - \frac{f'_c}{1400}) \leq 0.85 \tag{64}$$

Theory:

The function considers the location of the Plastic Center (PC) the same as the Geometric Center (GC) (which is at a depth $\frac{h}{2}$) **Fig. 22**, so that the resistant moment is calculated as (41), where $Fs_i = As_i E_y \epsilon_i$ for reinforcement steel.

5.2.10 Asymmetric reinforcement

5.2.11 Function: PSOAsymmetricRebar

Purpose: To determine an optimal arrangement of rebars asymmetrically over a column cross-section using the PSO algorithm based on a given 1t-ISR.

Syntax:

$[Mr\_col, h, Inertia\_xy\_modif, bestPerformance, best\_cost, bestnv, bestPerformanceEf, ...$
$bestArrangement, best\_disposicion] = PSOAsymmetricRebar(t4, b, h, rec, sepMin, Ac\_t\_elem, npuntos, fdpc, betac, ...$
$pu\_asym\_cols, load_conditions, condition\_cracking, plotRebarDesign, rebarOptimConv)$

Description: .

Output variables:

- Mr_col are the resisting bending moments for both axis directions for the optimally desined cross-section: $[M_{Rx}, M_{Ry}]$
- h is the increased height dimension of the cross-section; in case it suffers modification through the optimization design process to comply the corresponding restrictions
- $Inertia\_xy\_modif$ are the modified inertia momentums for both axis directions of the optimally reinforced cross-section: under cracking criteria (non-cracked or cracked)
- $bestPerformance$ is the optimal rebar area for the cross-section
- $best\_cost$ is the final construction cost for the optimally designed cross-section (considering only rebar volumes and assembly)
- $bestnv$ is the total number of rebars to be placed over the optimally designed cross-section
- $bestPerformanceEf$ is the critical structural efficiency for the optimally designed cross-section
- $bestArrangement$ is the list of rebar types for each of the rebars to be placed over the optimally designed cross-section
- $best_disposicion$ are the local coordinates of the optimal rebar option

Input variables:

- $t4$ is the optimal ISR from which the optimal rebar option is determined as: $[t1, t2, t3, t4]$ (see **Fig. 2**)
- b, h cross-section dimensions
- $npuntos$ number of points to be computed for the definition of the interaction diagram
- $fdpc$ is the f'_c reduced with the factor 0.85 according to code
- $\beta 1$ is determined as following (73) in units Kg, cm

$$0.65 \leq (\beta_1 = 1.05 - \frac{f'_c}{1400}) \leq 0.85 \tag{65}$$

- $sepMin$ is the minimum rebar separation constriction (55):
- rec are the concrete cover values for each axis direction of the cross-section
- pu_asym_cols are the unit cost data for the reinforcing bar volumes and assembly as an array of size: $[2, 7]$ by default, for which the first row corresponds to unit-cost values for each rebar type in case only one type of rebar results as an optimal design, and the second row consisting of only one unit cost in case more than one different type of rebar results as an optimal design (assuming an average unit-cost of all types of rebars)
- $load\_conditions$ is the load condition array: size = $[nloads, 4]$ in format $[nload, Pu, Mux, Muy]$
- $condition\_cracking$ is the parameter to indicate the cracking mechanism to be considered; options are $''Cracked''$ or $''Non-cracked''$
- $plotRebarDesign$ is the parameters that indicates if the rebar design results are required or not. Options are: (1) they are required, (2) they are not required
- $rebarOptimConv$ is the parameters that indicates if the optima rebar convergence is required or not. Options are: (1) they are required, (2) they are not required

Theory:

This function deploys the PSO algorithm **Algorithm 2.2** to find the optimal rebar option of as much of four different types of rebars (1 per each cross-section face) **Fig. 8**. For more reference of this method and other stochastic ones see [Wahde, 2008].

5.2.12 Function: dispositionRebarAsymmetric

Purpose: To compute the local coordinates of an asymmetric rebar option.

Syntax:

$[disposition\_rebar, separation\_hor1, separation\_hor2, ...$
$separation\_ver1, separation\_ver2] = dispositionRebarAsymmetric(b, ...$
$h, sepMin, rec, nv, number\_rebars\_sup, number\_rebars\_inf, number\_rebars\_izq, ...$
$number\_rebars\_der, RebarAvailable, op1, op2, op3, op4)$

Description: .

Output variables:

- $disposition\_rebar$ are the local coordinates of the optimal rebar option
- $separation\_hor1, separation\_hor2, ...separation\_ver1, separation\_ver2$ resultant rebar separation to be compared with the minimum

Input variables:

- b, h cross-section dimensions
- $sepMin$ is the minimum rebar separation constriction (55):
- rec are the concrete cover values for each axis direction of the cross-section
- $RebarAvailable$ rebar database consisting of an array of size $[7, 3]$ by default in format: $[\#rebar, diam, unit-weight]$
- $number\_rebars\_sup, number\_rebars\_inf, number\_rebars\_left, ...number\_rebars\_right$ number of rebars to placed on each of the cross-section boundaries
- $op1, op2, op3, op4$ resultant types of rebar for each of the four cross-section boundaries (upper boundary, lower boundary, left side and right side, respectively)

5.2.13 Function: EvaluateAsymmetric

Purpose: To compute the interaction diagram of an asymmetrically reinforced cross-section design option.

Syntax:

$[maxef, diagramaInteraccion, efficiency, cp\_axis, cxy] = ...$
$EvaluateAsymmetric(load\_conditions, npoints, position, b, h, ...$
$fy, fdpc, beta, E, number\_rebars\_sup, number\_rebars\_inf, number\_rebars\_left, ...$
$number\_rebars_right, rebarAvailable, dispositionRebar)$

Description: .

Output variables:

- $maxef$ is the critical structural efficiency according to the load conditions applied
- $diagramaInteraccion$ is the array containing the interaction diagram data
- $efficiency$ is the resume table of structural efficiency analysis for each load condition
- cp_axis are the central plastic locations for each axis direction of the cross-section: $[CPx, CPy]$
- cxy are neutral axis depth values for each axis direction of the cross-section

Input variables:

- b, h cross-section dimensions
- $rebarAvailable$ rebar database consisting of an array of size $[7, 3]$ by default in format: $[\#rebar, diam, unit - weight]$
- $number\_rebars\_sup, number\_rebars\_inf, number\_rebars\_left, ...number\_rebars\_right$ are the number of rebars to be placed over each of the four cross-section boundaries: upper boundary, lower boundary, left boundary, right boundary (the height dimensions defines the upper and lower boundary).
- $dispositionRebar$ are the local rebar coordinates
- $position$: are the variables to optimize in the PSO algorithms. In this case: $op1, op2, op3, op4$ as a vector of size $[1, 4]$ which are the resultant types of rebar for each of the four cross-section boundaries (upper boundary, lower boundary, left side and right side, respectively)
- $load\_conditions$ are the load conditions: vector of size $[nloads, 4]$ in format $[nload, Pu, Mux, Muy]$
- $npoints$ number of points to be computed for the definition of the interaction diagram

Theory:

The max compression resistance of the column cross-section is determined as (66) where A_c is the concrete net cross-section area and A_s is the total reinforcement area. On the other hand, the max tension resistance is determined as (67).

$$P_{oc} = 0.85 f_c'(A_c - A_s) + f_y(A_s) \tag{66}$$

$$P_{ot} = f_y(A_s) \tag{67}$$

For the computation of the interaction diagram, the resistance factors of $F_R = 0.65$ and $F_R = 0.75$ are applied respectively for a compression controlled and tension controlled cross-section case, according to the **ACI 318-19** and **NTC-17** codes. **Fig. 23**.

5.2.14 Function: eficienciaRecAsymm

Purpose: To compute the structural efficiency of an asymmetrically reinforced column cross-section.

Syntax:

$[maxef, eficiencia, cxy] = eficienciaRecAsymm(diagrama, ...$
$load\_conditions, pot, poc, c\_vector\_bar)$

Description: .

Output variables:

- $maxef$ is the critical structural efficiency of the column cross-section given different load conditions
- $eficiencia$ is the resume table of results consisting of $nload\_conditions$ rows and eight columns as:

$$[P_u, M_{ux}, M_{uy}, P_{Rx}, P_{Ry}, M_{Rx}, M_{Ry}, Eff]$$

- cxy is a vector containing the neutral axis depth of each cross-section direction according to the most critical load condition as $[cx, cy]$

Input variables:

- $diagrama$ is the interaction diagram data
- $load\_conditions$ is the array containing the load conditions: size $= [nload, 4]$ in format $[nload, Pu, Mux, Muy]$
- pot, poc are the max resistant axial force in tension of reinforcement steel (concrete is not considered) and compression of the whole reinforced cross-section area (both concrete area and rebar area are considered)
- c_vector_bar is the array containing the neutral axis depth values for both cross-section axis directions for all interaction diagram points: size $= [npoints + 2, 2]$

Theory:

The structural efficiency Ef is computed using the Bresler's formula with the *inverse load method* (68) for axial load values in which $\frac{P_u}{P_{oc}} \geq 0.1$ is complied, so that $Ef = [\frac{P_n}{P_{oc}} \leq 1.0]$.

$$\frac{1}{P_n} = \frac{1}{P_{Rx}} + \frac{1}{P_{Ry}} - \frac{1}{P_{ot}} \tag{68}$$

On the other hand, for small values of the axial load such that $\frac{P_u}{P_{oc}} \leq 0.1$ is complied, the *Contour load method* applies through the *bidirectional interaction equation* (69):

$$Ef = \frac{M_{nx}}{M_{Rx}} + \frac{M_{ny}}{M_{Ry}} \leq 1.0 \tag{69}$$

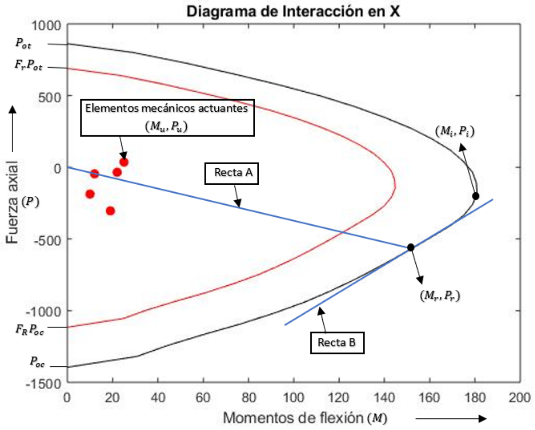


Figure 27: *Interaction diagram in the Cartesian plane as reference for comprehension of the application of analytical geometry for the computation of the structural efficiency of reinforced column cross-sections.*

Analytic geometry is applied for the determination of the resistance corresponding to each load condition P_R, M_{Rx}, M_{Ry} (see **Fig. 27**) as (70) and (71):

$$M_r = \frac{P_{i+1} + (\frac{P_i - P_{i+1}}{M_{i+1} - M_i})}{\frac{P_u}{M_u} - \frac{P_{i+1} - P_i}{M_{i+1} - M_i}} \tag{70}$$

$$P_r = \frac{P_u}{M_u} M_r \tag{71}$$

5.2.15 Function: bisectionMrVarAsymm

Purpose: To determine the neutral axis depth, axial and bending resistance from the interaction diagram of a reinforced concrete column cross-section with asymmetric reinforcement, for each for its points with the aid of the bisection root method.

Syntax:

$[root] = bisectionMrVarAsymm(cUno, cDos, fr, E, h, b, fdpc, beta, ea, nv, ...$
$number\_rebars\_sup, number\_rebars\_inf, number\_rebars\_left, ...$
$number\_rebars\_right, rebarAvailable, op1, op2, op3, op4, ...$
$dispositionRebar, cp);$

Description: .

Output variables:

- $root$ is a vector containing the neutral axis depth, axial resistant force and bending resistance of a reinforced column cross-section as $[c, F_R, M_R]$

Input variables:

- $cUno, cDos$ are the initial values of the neutral axis to commence iterations
- fr is the axial force resistance corresponding to the bending moment resistance for which the equilibrium condition $\sum F = 0$ is established to extract its corresponding bending moment resistance and neutral axis depth from the interaction diagram
- E Elasticity modulus of steel $(4200 \frac{Kg}{cm^2})$
- b, h cross-section dimensions
- $fdpc$ is the f'_c reduced with the factor 0.85 according to code
- β is determined as following (73) in units Kg, cm

$$0.65 \leq (\beta_1 = 1.05 - \frac{f'_c}{1400}) \leq 0.85 \tag{72}$$

- ea is the approximation error to terminate the root bisection method
- nv is the number of rebars to be placed over the cross-section
- $number\_rebars\_sup, number\_rebars\_inf, number\_rebars\_left, number\_rebars\_right$ are the number of rebars to be placed for each of the cross-section boundaries
- $dispositionRebar$ are the local coordinates of rebars over the cross-section
- $rebarAvailable$ data base of commercial available rebars. An array of size $[7, 3]$ by default; in format $[\#, diam, unit - weight]$

- $op1, op2, op3, op4$ types of rebar to be placed for each of the four boundaries of the cross-section (upper boundary, lower boundary, left boundary and right boundary)
- cp Plastic Center location for each axis-direction of the column cross-section

Theory:

The root bisection method is employed. For more reference of this method, see [Chapra & Canale, 2015].

5.2.16 Function: eleMecanicosVarAsymm

Purpose: To compute the sum of resistant forces of an asymmetrically reinforced column cross-section considering the distribution of rebars over the cross-section and concrete zone in compression.

Syntax:

$[eMecVar] = eleMecanicosVarAsymm(dispositionRebar, nv, number_rebars_sup, ...$
$number\_rebars\_inf, number\_rebars\_left, number\_rebars\_right, rebarAvailable, op1, op2, ...op3, op4, b, h, c, fdpc, E, beta, cp)$

Description: .

Output variables:

- $eMecVar$ vector that contains the output $[\sum F_s, \sum M_s; F_c, M_c]$

Input variables:

- $dispositionRebar$ are the local coordinates of rebars over the cross-section
- nv is the total number of rebars
- $number\_rebars\_sup, number\_rebars\_inf, number\_rebars\_left, number\_rebars\_right$ are the number of rebars to be placed in each cross-section boundary: upper boundary, lower boundary, left boundary and right boundary, respectively
- $op1, op2, op3, op4$ are the type of rebar to be placed over each of the cross-section's boundaries
- E Elasticity modulus of steel $(4200 \frac{Kg}{cm^2})$
- b, h cross-section dimensions
- $fdpc$ is the f'_c reduced with the factor 0.85 according to code
- $\beta 1$ is determined as following (73) in units Kg, cm

$$0.65 \leq (\beta_1 = 1.05 - \frac{f'_c}{1400}) \leq 0.85 \tag{73}$$

- cp is the Plastic Center location for each of the cross-section axis directions

Theory:

Given that the cross-section is reinforced asymmetrically, the function considers the location of the Plastic Center (PC) (computed as (74)) aside of the location of the Geometric Center (GC) (which is at a depth $\frac{h}{2}$ for the latter), so that the resistant moment is calculated as (75), where $Fs_i = As_i E_y \epsilon_i$ for reinforcement steel. For the computation of the Plastic Center the **Fig. 28** is considered as reference:

$$y = \frac{0.85 f'_c \frac{bh^2}{2} + \sum_{i=1}^{i=nbars} A_{s_i} f_u d_i}{0.85 f'_c bh + \sum_{i=1}^{i=nbars} A_{s_i} f_y} \tag{74}$$

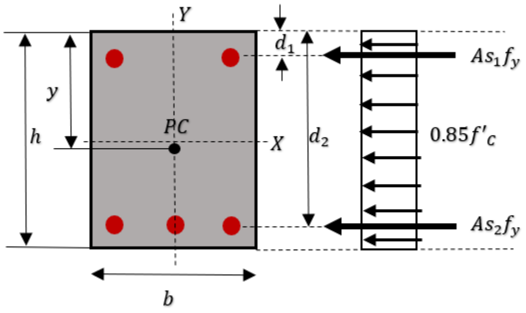


Figure 28: *Design mechanisms under flexure-compression for a symmetrically reinforced cross-section.*

$$M_n = \sum_{i=1}^{i=nbars} F_i = \sum_{i=1}^{i=nbars} A_{s_i} E_y \epsilon_i (y - d_i) + \beta_i a b f_c''(y - \frac{a}{2}) \tag{75}$$

5.3 Rebar analysis for isolated footings

5.3.1 Function: RebarOptionsFootings

Purpose: To determine an optimal rebar option for a transversal cross-section of a rectangular footing.

Syntax:

$[dimb, acRebar, nv, s, arrangement] = ...$
$RebarOptionsFootings(ac, dimb, RebarAvailable, sepMinCode)$

Description: .

Output variables:

- *dimb* final transversal cross-section width dimension (in case is augmented to comply the max-min rebar area or minimum rebar separation constraints)
- *acRebar* rebar area approximate to the given ISR area
- *nv* number of rebars
- *arrangement* is the rebar type to be used from the available commercial ones

Input variables:

- *dimb* transversal cross-section width
- *ac* ISR area
- *RebarAvailable* commercial available rebar database: by default size = $[7, 5]$ in format $[noption, \#, diam, rebarArea, lineal - weight]$
- *sepMinCode* minimum separation by code according to the ACI 318-19 code as $1.0inch$ (or specified higher for practical construction reasons)

Theory:

The function uses a *Simple search* method (or exhaustive search) to find the optimal rebar arrangement given the limited number of potential solutions. The transversal cross-section width dimension may be modified in case the rebar separation constraint is not comply for any potential solution.

5.3.2 Function: dispositionRebarSquareFootings

Purpose: To compute the local rebar coordinates over a transversal cross-section of a square isolated footings for which the rebar distribution is placed uniformly over both transversal cross-sections.

Syntax:

$[dispositionRebar] = dispositionRebarSquareFootings(b, h, rec, ...$
$rebarAvailable, nvt, RebarArrangement1, RebarArrangement2, axis)$

Description: .

Output variables:

- $dispositionRebar$ is the array containing the local rebar coordinates over the transversal cross-section

Input variables:

- b, h are the transversal cross-section dimensions
- rec is the concrete cover
- $rebarAvailable$ are the commercial available rebar database: size = $[7, 5]$ by default, in format $[noption, \#rebar, diam, rebar - area, lineal - weight]$
- nvt is a vector containing the quantity of rebars both in compression and tension as $[nb_{tension}, nb_{compression}]$
- $RebarArrangement1, RebarArrangement2$ are the arrays containing the rebar type used both in tension and compression, respectively. Size = $[nbars, 1]$ in format $[\#rebar_1, ..., \#rebar_n]$

Theory:

This function only applies for square isolated footings in which the rebar distribution is placed uniformly on both transversal cross-sections, according to code.

5.3.3 Function: dispositionRebarRectangularFootings

Purpose: To compute the local rebar coordinates over a transversal cross-section of a rectangular isolated footing for which the rebar distribution is placed non-uniformly over the larger transversal cross-sections.

Syntax:

$[dispositionRebar, arrangementfinal] = dispositionRebarRectangularFootings(b, h, rec, ...$
$rebarAvailable, nv\_t, RebarArrangement1, RebarArrangement2, axis, largerDim, dim\_zap)$

Description: .

Output variables:

- $dispositionRebar$ is the array containing the local rebar coordinates over the transversal cross-section

Input variables:

- b, h are the transversal cross-section dimensions
- rec is the concrete cover
- $rebarAvailable$ are the commercial available rebar database: size = $[7, 5]$ by default, in format $[noption, \#rebar, diam, rebar - area, lineal - weight]$
- nvt is a vector containing the quantity of rebars both in compression and tension as $[nb_{tension}, nb_{compression}]$
- $RebarArrangement1, RebarArrangement2$ are the arrays containing the rebar type used both in tension and compression, respectively. Size = $[nbars, 1]$ in format $[\#rebar_1, ..., \#rebar_n]$
- $axis$ is the axis direction in question: (1) axis ALONG the L footing dimension, (2) axis ALONG the B footing dimension
- $largerDim$ is the parameter that indicates which transversal cross-section is in analysis: for rectangular footings (1) means that the larger dimension is the L dimensions, (2) means that the larger dimension is the B dimension. If the footing is squared, then the parameters takes the value of (3)
- dim_zap are the two transversal cross-section dimensions of the footing: in format $[B, L]$

Theory:

This function computes the distribution of rebar over the larger transversal cross section of a rectangular isolated footing in which the rebar distribution is placed uniformly according to code (see **Fig. 29** and **Fig. 30**).

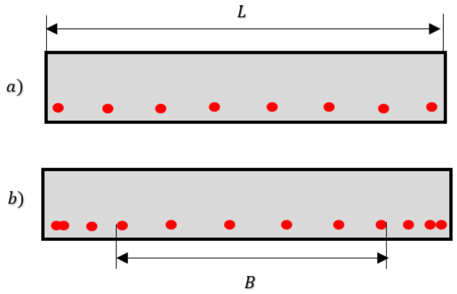


Figure 29: *Potential rebar solutions for transversal cross-sections of rectangular isolated footings.*

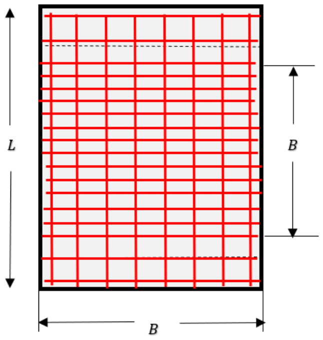


Figure 30: *Transversal reinforcement in a rectangular isolated footing.*

5.3.4 Function: EfcriticalFootings

Purpose: To compute the structural efficiency of a rebar option, considering only the steel in tension for a transversal cross-section of an isolated footing in which the steel in compression is designed only by temperature.

Syntax:

$$[maxef, mr] = EfcriticalFootings(bz, fdpc, actension, d, fy, mu\_real\_axis)$$

Description: .

Output variables:

- $maxef, mr$ is the structural efficiency of the designed transversal cross-section and the resisting bending moment, respectively

Input variables:

- bz width dimension of the transversal cross-section in analysis (cm)
- d effective footing height $h - cover$
- $actension$ is the quantity of rebar area in tension (cm^2)
- $fdpc$ is the f'_c used reduced by the 0.85 factor defined by code ($\frac{Kg}{cm^2}$)
- fy is the yield strength of the rebar steel ($\frac{Kg}{cm^2}$)
- mu_real_axis is the demanding bending moment

Theory:

The function only considers the steel rebar in tension, so that the equation (49) applies.

5.3.5 Function: EvaluateCostRebarFoot

Purpose: To compute the construction cost of a rebar option for an isolated footing transversal cross-section.

Syntax:

$cost = EvaluateCostRebarFoot(be, RebarAvailable, arrangement1, arrangement2, pu)$

Description: .

Output variables:

- *cost* is the construction cost of a rebar option over an isolated footing transversal cross-section

Input variables:

- *be* longitudinal dimension perpendicular to the transversal cross-section in analysis (length of rebars)
- $arrangement1, arrangement2$ are the rebar types used for the steel in tension and compression, respectively
- *pu* is the unit construction assembly cost of rebars in isolated footings: units $\frac{\$}{Kg}$. The unit cost is considered using an average value of all rebar types's assembly performances (assuming that in an isolated footing as much as four different types of rebar may be placed)
- *RebarAvailable* is the array containing the available commercial rebar database: size = $[7,5]$ by default. In format: $[noption, \#rebar, diam, area, lineal - weight]$

5.3.6 Function: EvaluateCostISRFoot

Purpose: To compute the estimated construction cost given an ISR data.

Syntax:

$cost = EvaluateCostISRFoot(be, rec, act, acmin, pu)$

Description: .

Output variables:

- *cost* is the estimated construction cost of an ISR over an isolated footing transversal cross-section

Input variables:

- *be* longitudinal dimension perpendicular to the transversal cross-section in analysis (length of rebars)
- *act* is the steel area reinforcement in tension over the transversal cross-section (lower boundary)
- *acmin* is the minimum steel area reinforcement by temperature over the zone in compression of the transversal cross-section (upper boundary)
- *rec* is the concrete cover in *cm*
- *pu* is the unit construction assembly cost of rebars in isolated footings: units $\frac{\$}{Kg}$. The unit cost is considered using an average value of all rebar types's assembly performances (assuming that in an isolated footing as much as four different types of rebar may be placed)

5.3.7 Function: isrFootings

Purpose: To determine an optimal rebar option of both transversal cross-section of an isolated footing subject to eccentric biaxial loads.

Syntax:

$[hmodif, mu\_axis, barDispositionFootings, arrangement\_bar\_footings, ...$
$nbars\_footings, AcBar, bestCost\_elem, list\_ef\_footings, list\_mr\_footings] = ...$
$isrFootings(pu\_steel\_footings, h, be, le, rec, fc, fy, load\_conditions, dimCol, ...$
$cols_sym_asym_isr, ductility, optimConv, PlotRebarDesign)$

Description: .

Output variables:

- $hmodif$ is the final modified design of the footing height
- mu_axis are the effective distributed bending moments for both transversal cross-section of the footing in analysis
- $barDispositionFootings$ is the collection of local rebar coordinates of both transversal cross-section of the footing. Size = $[total_{bars}, 2]$
- $arrangement\_bar\_footings$ is the list that contains the rebar types of all rebars of the optimal designed footing. A number between $1 - 7$ by default, comprising the 7 rebar types commercially available by default.
- $nbars_footings$ is the list containing the total number of rebars used both in tension and compression for both transversal cross-sections of the footing
- $AcBar$ is the list containing the quantity of reinforcement area used for both transversal cross-sections as the sum of the area in compression (or temperature) and tension
- $bestCost$ is the total assembly cost of the optimal reinforcement option, either with an ISR or with rebars
- $list_ef_footings$ is the list of final structural efficiencies for both transversal cross-sections of the footing. Size $= [1, 2]$
- $list_mr_footings$ is the list of final resistant bending moments for both of the transversal cross-sections. Size $=$ $[1, 2]$

Input variables:

- be, le are the initial given transversal cross-section width dimensions
- h is the initial given footing height dimension
- $pu\_steel\_footings$ is the unit construction assembly cost of rebars in isolated footings: units $\frac{\$}{Kg}$. The unit cost is considered using an average value of all rebar types's assembly performances (assuming that in an isolated footing as much as four different types of rebar may be placed)

- fc is the minimum steel area reinforcement by temperature over the zone in compression of the transversal cross-section (upper boundary)
- fy is the yield stress of the reinforcing steel
- *load_conditions* are the eccentric biaxial load conditions applied to the footing through the column that supports
- rec is the concrete cover in cm
- $dimCol$ are the columns cross-section dimensions that the footing supports
- *cols_sym_asym_isr* is the parameter that indicates if only an ISR optimal design is required or an optimal rebar optimization design process. Options are $''ISR''$ or $Symmetric$
- *ductility* is the ductility parameter that indicates the level of ductility required for the transversal cross-sections of the footing through the max-min quantity of steel area. (1) low ductility, (2) medium ductility, (3) high ductility
- $optimConv$ is the parameters that indicates if the optima ISR convergence plots are required or not. Options are: (1) they required, (2) they are not
- $PlotRebarDesign$ is the parameters that indicates if the optima rebar design plots are required or not. Options are: (1) they required, (2) they are not

Theory:

The **function** $RealPressuresFoot$ is used to distribute the soil pressure at the bottom of the footing in order to determine later on the effective acting bending moments for both transversal cross-sections with the **function** *Moment_Distribution_Footings*. Then once these bending moments have been determined the optimal design begins, first with the determination of an optimal reinforcement area for both transversal cross-section through the ISR by using the **function** *SGD_1tFoot_ISR* and second with the determination of an optimal rebar arrangement (if required) with the **function** $RebarOptionsFootings$. At the end, the computation of the construction reinforcement cost is carried out with the **functions** $EvaluateCostRebarFoot$ or $EvaluateCostISRFoot$ given the case.

The ductility demand for the transversal cross-sections is considered, with which the max-min reinforcing steel area is controlled as: low ductility, medium ductility or high ductility (76) and (77) respectively..

Low and Medium ductility:

$$\frac{0.7bd}{b-2rec}\frac{\sqrt{0.85f'_c}}{f_y} \leq t \leq \frac{0.9(0.85f'_c)}{(b-2rec)f_y}\frac{bd(6000\beta_1)}{(f_y+6000)} \tag{76}$$

High ductility:

$$\frac{0.7bd}{b-2rec}\frac{\sqrt{0.85f'_c}}{f_y} \leq t \leq \frac{0.75(0.85f'_c)}{(b-2rec)f_y}\frac{bd(6000\beta_1)}{(f_y+6000)} \tag{77}$$

In case, the rebar constraints such as the minimum rebar separation are not complied, then there is modification of the transversal cross-section dimensions and the optimal design process is carried out again until all design restrictions and constraints are complied.

5.3.8 Function: ExportResultsIsolFootings

Purpose: Computes the exportation of the design results of an isolated footing element into a .txt file on a prescribed folder route.

Syntax:

$ExportResultsIsolFootings(directionData, bestDispositionFootings, ...$
$dimensionFootingCollection, nbarsFootingsCollection, typesRebarFooting, ...$
$coordBaseFooting)$

Description:

Input variables:

- $directionData$ is the folder disc location to save the results
- $dimensionFootingCollection$ is the array containing the isolated footing transversal cross-section dimensions data
- $coordBaseFooting$ is the array containing the coordinates of the isolated footing base
- $bestDispositionFootings$ is the array containing the local rebar coordinates of the isolated footing transversal cross-sections
- $nbarsFootingsCollection$ is the total number of rebars on the transversal cross-sections, both in tension and compression
- $typesRebarFooting$ is the list of the rebar types used in the element

6 Design-Analysis of 2D frames

6.0.1 Function: DesignRCPlaneFrameBCIF

Purpose: To design the reinforcement of the elements of a plane frame concrete structure composed of rectangular beams, rectangular columns and rectangular isolated footings.

Syntax:

$[totalWeightStruc, wsteelColsTotal, pac\_cols\_elem, sectionRestrictions, ...$
$Mp, dimensions, displacementsRightLeft, unitWeightElem, wsteelConcBeamsElem, ...$
$wsteelConcColsElem, wsteelConcFootingsElem, hefootings, dimFoot, ...$
$totalCostStruc, inertiaElem, wsteelStructure] = DesignRCPlaneFrameBCIF...$
$(nnodes, bc, ni, nf, Edof, coordxy, puBeams, type\_elem, puCols, nbars, np, ...$
$lenElem, coordBaseCols, fcElem, inertiaElem, qadm, FSfootings, nodesSupportColumns, ...$
$puSteelFootings, dimensions, Eelem, fcbeams, fccols, fcfootings, fglobal, ...$
$qbary, areaElem, ForcesDOFseismic, floorElem, colsSymAsymISR, ...$
$conditionCracking, duct, elem\_cols, elem\_beams, recxyCols, directionData, plotAnalysisResults)$

Description:

Output variables:

- $totalWeightStruc$ is the total weight of the structural frame, considering the reinforcing steel and the concrete volume
- $wsteelColsTotal$ is the sum of reinforcing steel weight of each of the columns composing the structural frame
- $pacColsElem$ is a vector containing the percentage steel area of each column. Size = $[ncolumns, 1]$
- $sectionRestrictions$ is a parameter that indicates weather or not the dimension constraints of both beam and column elements are being complied. If the parameter's value is equal to 1, then the dimension restrictions are not being complied, if its value is equal to 0 all dimension restrictions are being complied
- Mp are the resisting bending moments at the ends of each element composing the structural frame (Plastic Moments)
- $dimensions$ are the new modified cross-section dimensions of the elements
- $displacementsRightLeft$ is an array containing the node displacements of the structural frame extracted from both structural static linear analysis with lateral seismic loads to the right and to the left. Size = $[NDOF, 2]$ in format $[dip_{right}, dip_{left}]$
- $unitWeightElem$ Is the array containing the unit-self-weight of each structural element considering both the concrete and steel reinforcement
- $wsteelConcBeamsElem$ is the vector containing the total weight of each of the beam elements, considering both the concrete volume and the steel reinforcement
- $wsteelConcColsElem$ is the vector containing the total weight of each of the column elements, considering both the concrete volume and steel reinforcement
- $wsteelConcFootingsElem$ is the vector containing the total weight of each of the footing elements, considering both the concrete volume and steel reinforcement

- $hefootings$ is the vector containing the final designed height dimensions of the isolated footings
- $dimFoot$ is the vector containing the transversal cross-section dimensions of all footings. Size = $[nfootings, 2]$ in format $[Be, Le]$
- $totalCostStruc$ Is the total construction cost of the structural frame considering the reinforcing steel design and concrete volumes
- $inertiaElem$ are the final cross-section inertia momentums of each structural element after applying a cracked or non-cracked cross-section mechanism
- $wsteelStructure$ is the total weight of steel reinforcement of the structural frame

Input variables:

- $nnodes$ is the number of nodes of the structural frame
- bc is the boundary condition vector of the restricted DOF with their pre-established displacement condition. Size = $[DOF, 2]$ in format $[nDOF, displacement]$
- ni, nf are the list of initial and final nodes ID's, respectively, for each of the elements composing a structural frame in the order pre-established
- $Edof$ is the topology array. Size = $[nbars, 7]$ in format $[no_{bar}, DOF_{initial-node}, DOF_{final-node}]$
- $coordxy$ is an array containing the node coordinates. Size = $[n_{nodes}, 2]$ in format $[x, y]$
- $puBeams$ is the unit construction cost of steel reinforcement assembly for a beam element, as an average of all assembly performances for each type of rebar commercially available (considering that as many as 6 different types of rebar may be placed in a beam element)
- $type\_elem$ is an array containing the element ID labels of each of the elements composing a structural frame: either $''Beam''$ or $''Col''$. Size = $[nbars, 2]$ in format $[nbar,'' label'']$
- $puCols$ are the unit construction cost data for the reinforcing bar assembly. For symmetric reinforcement format is default $pu\_col = [PU_{\#4}, PU_{\#5}, PU_{\#6}, PU_{\#8}, PU_{\#9}, PU_{\#10}, PU_{\#12}]$; and for asymmetric reinforcement as an array of size: $[2, 7]$ by default for asymmetric reinforcement, for which the first row corresponds to unit-cost values for each rebar type in case only one type of rebar results as an optimal design, and the second row consisting of only one unit cost in case more than one different type of rebar results as an optimal design (assuming an average unit-cost of all types of rebars)
- $nbars$ is the number of bars or elements composing a structural frame
- np are the number of points of analysis for the computation of the mechanic elements distribution of an element
- $lenElem$ is an array containing the length of each structural element composing a structural frame
- $coordBaseCols$ are the coordinates of each of the columns' base point. Size = $[ncolumns, 2]$ in format $[x_{bc}, y_{bc}]$
- $fcElem$ is the vector containing the f'_c used for each of the elements composing the structural frame
- $inertiaElem$ is the vector containing the cross-section inertia momentum of each of the elements composing the structural frame
- $qadm$ is the max admissible bearing load of the soil supporting the footings of the structural frame. Is used for the design of the footings

- $FSfootings$ is the Safety Factor used for the design of the footings
- *nodesSupportColumns* is an array containing the nodes in which each of the base columns are supported in contact with their respective isolated footing. Size = $[2, nfootings]$ in format:

$$[node; \\ columnID]$$

- *puSteelFootings* is the unit construction cost of steel reinforcement assembly for an isolated footing element, as an average of all assembly performances for each type of rebar commercially available (considering that as many as 4 different types of rebar may be placed in an isolated footing element)
- *dimensions* is the array containing the cross-section dimensions of all elements composing the structural frame. Size = $[nbars, 2]$ in format $[b_e, h_e]$
- *Eelem* is the yield stress of each of the structural elements' material (is a function of the f'_c used for each element)
- $fcbeams, fccols, fcfootings$ is the f'_c used for beam, columns and isolated footings elements, respectively
- $fglobal$ is the vector containing the punctual node forces applied to the structural frame. Size = $[NDOF, 1]$
- *qbary* is the array containing the value of the distributed load acting downwards over each of the elements (applies specially for beams)
- *areaElem* is the vector containing the cross-section area of all of the structural elements composing the structural frame
- $ForcesDOFseismic$ is the list of DOF on which the lateral equivalent base shear forces are acting
- $floorElem$ is an array containing the elements' IDs that correspond to each of the floors of the structural element. Size = $[nfloors, nmaxElements + 1]$. It is used to compute the weight, mass and stiffness of each floor. Format $[nfloor, elem_1, ..., elem_n]$. In case one floor has less number of elements than the others, the array should be completed by zeros
- $colsSymAsymISR$ is the parameter that indicates what type of reinforcement design is to be carried out (mainly for columns, although it also defines the type of design for the other element types). Options are $''ISR'','' Symmetric'','' Asymmetric''$
- *conditionCracking* is the parameter that indicates what type of cracking mechanism is considered for the computation of the inertia momentums of each of the elements' cross-section. Options are: $''Cracked''$ and $''Non-cracked'')$
- $elem\_cols, elem\_beams$ are the vector containing the elements' IDs or numbers corresponding to the type of structural element. Only for columns and beams, respectively
- *recxyCols* is a vector containing the concrete cover for both axis directions of a rectangular column cross-section as $cover_x, cover_y$
- *directionData* is the disc route direction to save the design data
- *plotAnalysisResults* is the parameters that indicates if the linear static structural analysis results are required or not. Options are: (1) they are required, (2) they are not required

Theory:

The function designs optimally the reinforcement in all the beam elements, columns elements and isolated footing elements composing the structural frame. It used the **function** *PlaneFrameStaticLinearAnalysis* to perform a linear static analysis to obtain the mechanical elements on each bar for their design. Only longitudinal reinforcement is designed, by using the **functions** *beams_isr*, *isr_columns_sym_asym* and *isr_footings* for beams, columns and isolated footings, respectively. The design results are exported to .txt files so that they can be used by the VISUAL-CALRECOD library for the 3D visualization through ANSYS SpaceClaim.

6.0.2 Function: PlaneFrameStaticLinearAnalysis

Purpose: To execute a linear static analysis of plane frame, and compute the mechanic element diagrams for each of its elements.

Syntax:

$[displacements, reactions, Ex, Ey, esbarsnormal, esbarsshear, esbarsmoment] = ...$
$PlaneFrameStaticLinearAnalysis(nnodes, nbars, Eelem, areaElem, inertia, bc, ...$
$fglobal, ni, nf, qbary, Edof, np, coordxy, plotAnalysisResults)$

Description:

Output variables:

- *displacements* is a vector containing the EDOF displacements. Size: $[nEDOF, 1]$
- *reactions* is the EDOF reaction forces. Size $[nEDOF, 1]$
- Ex, Ey are the element end nodes' coordinates
- *esbarsnormal* are the normal mechanic elements for each bar. Size: $[np, nbars]$
- *esbarsshear* are the shear mechanic elements for each bar. Size: $[np, nbars]$
- *esbarsmoment* are the bending moment mechanic elements for each bar. Size: $[np, nbars]$

Input variables:

- *nnodes* are the number of nodes of the structure
- *nbars* are the number of structural elements of the structural frame (neglecting the footings)
- *Eelem* is the vector containing the elasticity modulus for each bar
- *areaElem* is the vector containing the cross-section area of each element
- *inertia* is the vector containing the cross-section inertia momentum of each element
- *bc* is the array containing the boundary conditions for the respective prescribed (or restricted) DOF. Size: $[nRestrictedDOF, 2]$ in format $[DOF, prescribed - displacement]$
- *fglobal* is the global external force vector for each DOF. Size: $[NDOF, 1]$
- ni, nf are the vectors containing the initial and final nodes for each element. Size: $[nbars, 1]$ for each
- *qbary* are the distributed vertical loads on the elements. Array of size: $[nbars, 2]$ in format $[nbar, load]$
- *Edof* is the topology matrix, consisting of the DOF for each element. The DOF of the initial node are placed first as: $[nbar, DOFxi, DOFyi, DOF\theta i, DOFxf, DOFyf, DOF\theta f]$
- *np* are the number of points for the evaluation of the mechanic elements for each structural element
- *coordxy* is the array containing the node coordinates. Size = $[nNodes, 2]$ in format $[xi, yi]$

- *plotAnalysisResults* is the parameter that indicates if the structural analysis results should be plotted or not. Options are: (1) the plots are required, (2) the plots are not required

Theory:

For this function, the library CALFEM must be uploaded (see [CALFEM, 2004]).

7 Graphic functions

7.1 Reinforcement of beam cross-sections

7.1.1 Function: beamReinforcedSection

Purpose: To plot the reinforcement of a designed beam cross-section.

Syntax:

$$beamReinforcedSection(h, b, disposition\_rebar, barTypes1, barTypes2)$$

Description:

Input variables:

- $barTypes1, barTypes2$ Vectors that contain the type of rebar for the optimal option both in tension and compression, respectively. The vectors size is of one column with nrebar rows containing a number between 1 and 7 according to the available commercial rebar types stated by default
- $disposition\_rebar$ local coordinates of rebars over the cross-section

7.1.2 Function: MidLeftRightBeamReinforcedSections

Purpose: To plot the reinforcement of middle-span, left-span and right-span designed beam cross-sections.

Syntax:

$MidLeftRightBeamReinforcedSections(h, b, disposition\_rebarMid, ...$
$disposition\_rebarLeft, disposition\_rebarRight, barTypes1Mid, barTypes2Mid, ...$
$barTypes1Left, barTypes2Left, barTypes1Right, barTypes2Right)$

Description:

Input variables:

- $barTypes1Mid, barTypes2Mid$ Vectors that contain the type of rebar of the middle-span cross-section of a beam element for both in tension and compression, respectively. The vectors size is of one column with nrebar rows containing a number between 1 and 7 according to the available commercial rebar types stated by default
- $barTypes1Left, barTypes2Left$ Vectors that contain the type of rebar of the left-span cross-section of a beam element for both in tension and compression, respectively. The vectors size is of one column with nrebar rows containing a number between 1 and 7 according to the available commercial rebar types stated by default
- $barTypes1Right, barTypes2Right$ Vectors that contain the type of rebar of the right-span cross-section of a beam element for both in tension and compression, respectively. The vectors size is of one column with nrebar rows containing a number between 1 and 7 according to the available commercial rebar types stated by default
- $disposition\_rebarMid$ local coordinates of rebars over the middle span cross-section
- $disposition\_rebarLeft$ local coordinates of rebars over the left span cross-section
- $disposition\_rebarRight$ local coordinates of rebars over the right span cross-section

7.2 Interaction diagrams and rebar layout of column cross-sections

7.2.1 Function: diagramISR

Purpose: To graph the interaction diagram of a reinforced column cross-section.

Syntax:

$diagramISR(diagramInteraction, conditions)$

Description:

Input variables:

- $conditions$ load conditions in format of $n\_load\_conditions$ rows and four columns $[n\_load, P_u, Mu_x, Mu_y]$
- $diagramInteraction$ interaction diagram data computed by using the **function** $width\_efficiency$ (p. 41).

7.2.2 Function: diagramsFinalRebarCols

Purpose: To graph the interaction diagram of a reinforced column cross-section as well as the reinforced cross-section rebar layout.

Syntax:

$diagramsFinalRebarCols(load\_conditions, diagrama, disposicion\_varillado, ...h, b, arregloVar)$

Description:

Input variables:

- *conditions* load conditions in format of *n_load_conditions* rows and four columns $[n\_load, P_u, Mu_x, Mu_y]$
- *diagrama* interaction diagram data computed by using the **function** *diagramasDisposicion* (p. 65)
- *disposicion_varillado* position local coordinates of a rebar design option
- b, h cross-section dimensions
- $arregloVar$ is the list of rebar types for all rebars: a list of size nbars consisting of a number between 1 to 7 by default

7.3 Reinforcement of isolated rectangular footing cross-sections

7.3.1 Function: ReinforcedSectionsFooting

Purpose: To plot both reinforced concrete sections of a rectangular isolated footing.

Syntax:

$ReinforcedSectionsFooting(h, be, le, dispositionRebar1, ...$
$dispositionRebar2, barTypes1B, barTypes2B, barTypes1L, barTypes2L)$

Description:

Input variables:

- h footing height
- be, le transversal cross-section dimensions
- $dispositionRebar1, dispositionRebar2$ local rebar coordinates over the transversal cross-section with the be and le dimension, respectively
- $barTypes1B, barTypes1L$ types of rebar in tension for both transversal cross-sections, be and le dimension, respectively
- $barTypes2B, barTypes2L$ types of rebar in compression for both transversal cross-sections, be and le dimension, respectively

8 Structural Frame System Functions

8.0.1 Function: WeightStruc

Purpose: To compute the weight of a reinforced concrete plane frame structure as well as all of its elements individually. All units are in Kg, cm

Syntax:

$[wsteelColsTotal, pacColsElem, wsteelConcBeams, wsteelConcCols, ...$
$wsteelConcFootings, volbeams, volcols, volfoot, wsteelStruc, wconcStruc, wbeams, ...$
$wcols, wfootings, weightStructure] = WeightStruc(elem\_cols, elem\_beams, ...$
$lenElem, areaElem, areaBarbeams, areaBarFootings, hfootings, nbeams, ...$
$ncols, steelareaCols, nfootings, dim\_zap)$

Description:

Output variables:

- $wsteelColsTotal$ Sum of the total weight of reinforcing steel of each column
- $pacColsElem$ Vector containing the percentage area of steel reinforcement on each of the columns. Size = $[ncolumns, 1]$
- $wsteelConcBeams$ Sum of the weight of both steel reinforcement and concrete of each of the beams
- $wsteelConcCols$ Sum of the weight of both steel reinforcement and concrete of each column
- $wsteelConcFootings$ Sum of the weight of both steel reinforcement and concrete of each isolated footing
- $volbeams, volcols, volfoot$ are sum of the volume of concrete of each beam, column and isolated footing, respectively
- $wsteelStruc$ Total weight of steel reinforcement of the whole structural frame
- $wconcStruc$ Total weight of concrete of the whole structural frame
- $wbeams, wcols, wfootings$ sum of the total weight of each type of element; beams, columns and isolated footings, respectively
- $weightStructure$ Total weight of the structure, considering both steel reinforcement and concrete volumes

Input variables:

- $elem\_cols$ is the vector containing the element number code of those elements identified as columns
- $elem\_beams$ is the vector containing the element number code of those elements identified as beams
- $lenElem$ is a vector containing the length of each element
- $areaElem$ is the vector containing the cross-section area of each element
- $areaBarbeams$ is the vector containing the quantity of steel rebar area of each beam element

- $areaBarFootings$ is the vector containing the quantity of steel rebar area of each isolated footing element
- $hfootings$ is the vector containing the design height dimension of each isolated footing
- $nbeams, ncols$ are the total number of beam and column elements, respectively
- $steelareaCols$ is the vector containing the total rebar area for each column element
- $nfootings$ is the number of isolate footing elements
- $dimFootings$ is the vector containing the transversal design cross-sections of each isolated footing

8.0.2 Function: CostStruc

Purpose: To compute the total construction cost of a reinforced concrete plane frame, considering only reinforcing steel and concrete volumes.

Syntax:

$totalCostStruc = CostStruc(costSteelBeams, costSteelCols, ...$
$costSteelFootings, fcbeams, fccols, fcfootings, vol\_beams, ...$
$vol\_cols, vol\_footings)$

Description:

Output variables:

- $totalCostStruc$ is the total construction cost estimated for the structural frame, considering steel reinforcement and concrete volumes

Input variables:

- $totalCostStruc$ Is the total construction cost of the structural frame, considering both reinforcing steel and concrete volumes
- $costSteelBeams$ is the vector containing the construction assembly cost of steel reinforcement of each beam
- $costSteelCols$ is the vector containing the construction assembly cost of steel reinforcement of each column
- $costSteelFootings$ is the vector containing the construction assembly cost of steel reinforcement of each isolated footing
- $fcbeams, fccols, fcfootings$ are the fc'_c in $\frac{Kg}{cm^2}$ used for beams, columns and isolated footings
- $volbeams, volcols, volfootings$ are the total concrete volumes used for each type of element, by using the **function** $WeightStruc$

8.0.3 Function: AdjustDimElemFrames

Purpose: To adjust cross-section dimensions of elements accordingly to make them uniform.

Syntax:

$[wsteelColsTotal, pacColsElem, wsteelConcBeams, wsteelConcCols, ...$
$wsteelConcFootings, volbeams, volcols, volfoot, wsteelStruc, wconcStruc, wbeams, ...$
$wcols, wfootings, weightStructure] = WeightStruc(elem\_cols, elem\_beams, ...$
$lenElem, areaElem, areaBarbeams, areaBarFootings, hfootings, nbeams, ...$
$ncols, steelareaCols, nfootings, dim\_zap)$

Description:

Output variables:

- *dimensions* are the new modified cross-section dimensions

Input variables:

- *nbars* number of elements composing the structural frame
- *type_elem* is the identification vector containing the label for each elements as $''Beam''$ or $''Col''$ for each of the elements. Size $[nbars, 2]$ in format $[nelem, label]$
- *nnodes* total number of nodes of the structural frame to be reviewed
- *dimensions* is the array containing the cross-section dimensions data for each of the elements. Size $= [nbars, 2]$ in format $[b_e, h_e]$
- *coord_base_cols* is the array containing the coordinates of the lower ends' centroid of each column. Size $= [ncolumns, 2]$ in format $[x_c, y_c]$
- ni, nf vectors containing the ID of initial nodes and final nodes of each element, respectively, in the order pre-established

Theory:

Each node is reviewed so that the upper column has equal or smaller cross-section dimensions than the lower one. The intersecting beam's cross-section are also reviewed so that its width dimension is smaller or equal than the width of the upper column (in case two columns intersect). See **Fig. 31**

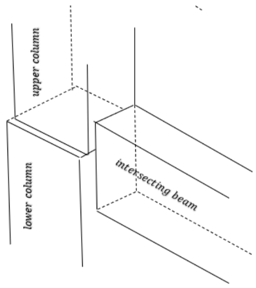


Figure 31: *Dimensions design criteria for beam-column nodes.*

Note: this function must be entered into a loop of at least 2 times the number of floors of the structural frame in question, given that the node revision is carried out in order per node-ID.

9 Visual CALREDOC

As a complement of the CALREDOC MatLab toolbox, the development of a library in ANSYS SpaceClaim using Python language has been carried out as well. Such library aids the visualization of any resulting design using the CALREDOC Toolbox through visual programming, so that a detailed evaluation of any design may be better assessed as CAD results.

So far, the Visual CALREDOC library computes the visualization of 2D Concrete Frames and its elements' longitudinal rebar (composed only beams, columns and isolated footings). It is expected that in the upcoming years and versions the library can be able to compute the visualization of transversal reinforcement (stirrups), and 3D frames, applicable for other types of structural elements (such as slabs, continuing footings, etc.).

9.1 General functions

9.1.1 Function: archivomatriz

Purpose: to transform an open file content into a matrix array.

Syntax:

$archivo\_matriz(variable\_archivo, separador):$

Description:

Output variables:

- *matriz_archivo*: resultant matrix array from the read file

Input variables:

- *variable_archivo* is the opened file to read with the function $open(dirCarpeta\_rec).read()$
- *separador* is the text separator format (for this case a blank space ' ')

9.1.2 Function: designdatareading

Purpose: to read a .txt file, and import it as a numerical array.

Syntax:

$$design\_data\_reading(direccion\_carpeta\_principal, name\_file, separador):$$

Description:

Output variables:

- *design_data* numeric array resultant from the read .txt file

Input variables:

- *direccion_carpeta_principal* route of .txt file location in disc
- $name\_file$ is the name of the .txt file
- *separador* is the text separator format (for this case a blank space ' ')

9.1.3 Function: dimelem

Purpose: to read a .txt file, containing the dimensions of a structural element type.

Syntax:

$dim\_elem(direccion\_carpeta\_principal, namefile\_entrydata\_elem, separador):$

Description:

Output variables:

- $nelem$ numeric array that contains the number of elements of a certain type
- $entry\_elem$ is the data that contain the .txt file, imported as a numerical array

  For beam elements, the data in the .txt file should be in the following format: $[b, h, span-length, sup-left-cover, inf-left-cover, sup-mid-cover, inf-mid-cover, sup-right-cover, inf-right-cover]$

  For column elements, the data in the .txt file should be in the following format:

  $[b, h, length, height-cover, width-cover]$

  For footing elements, the data in the .txt file should be in the following format:

  $[B, L, h, height-cover]$

Input variables:

- $direccion\_carpeta\_principal$ route of .txt file location in disc
- $namefile\_entrydata\_elem$ is the name of the .txt file
- $separador$ is the text separator format (for this case a blank space ' ')

9.2 Beams

9.2.1 Function: RectangularPrismaticSolidBeams

Purpose: to compute the visualization of prismatic solid rectangular beams on the 2D-XY space.

Syntax:

$Rectangular\_Prismatic\_Solid\_Beams(nbeams, coord\_desp\_beams, entrada\_Beams)$

Description:

Input variables:

- $nbeams, entrada\_Beams$ are the result of using the function *dim_elem* (p. 110)
- *coord_desp_beams* are the base directional axis location of each element, as a result of using the function *design_data_reading* (p. 109)

9.2.2 Function: RebarRectangularSolidBeams

Purpose: to compute the visualization of longitudinal prismatic rebar in rectangular solid beam elements (the X-axis is the longitudinal axis).

Syntax:

$Rebar\_Rectangular\_Solid\_Beams(nbeams, coord\_desp\_beams, entrada\_Beams, ...$
$rebar\_disp\_beams, arrangement\_rebar\_beams, varDisponibles, nrebar\_beams)$

Description:

Input variables:

- *nbeams*, *entrada_Beams* are the result of using the function *dim_elem* (p. 110)
- *coord_desp_beams* are the base directional axis location of each beam element, as a result of using the function *design_data_reading* (p. 109)
- *rebar_disp_beams* are the local coordinates (X,Y) of the rebar over the three cross-section of each beam element
- *arrangement_rebar_beams* is the list of the type of rebar for all of three cross-section of each beam
- *varDisponibles* are the available commercial rebar types as a data base consisting of an array vector of length 7 (by default) for which each vector element corresponds to the rebar type diameter
- *nrebar_beams* is a vector array of *nbeams* rows and six columns (two of which correspond to each of the three cross-section of a beam)

9.3 Columns

9.3.1 Function: RectangularPrismaticSolidColumns

Purpose: to compute the visualization of prismatic solid rectangular columns on the 3D space. The Y-axis direction is the vertical direction.

Sintax:

$Rectangular\_Prismatic\_Solid\_Columns(ncols, entrada\_Col, coord\_desp\_cols)$

Description:

Input variables:

- $ncols, entrada\_Col$ are the result of using the function *dim_elem* (p. 110)
- *coord_desp_cols* are the base directional axis location of each element, as a result of using the function *design_data_reading* (p. 109)

9.3.2 Function: RebarRectangularSolidColumns

Purpose: to compute the visualization of longitudinal prismatic rebar in rectangular prismatic columns.

Syntax:

$Rebar\_Rectangular\_Solid\_Columns(ncols, coord\_desp\_cols, entrada\_Col,$ $rebar\_disposition\_columns, ...$ $arrangement\_rebar\_cols, varDisponibles, nrebar\_cols)$

Description:

Input variables:

- $ncols, entrada\_Col$ are the result of using the function dim_elem (p. 110)
- $coord\_desp\_cols$ are the base directional axis location of each element, as a result of using the function $design\_data\_reading$ (p. 109)
- $rebar\_disposition\_columns$ are the local coordinates (X,Y) of the rebar over each column cross-section
- $arrangement\_rebar\_cols$ is the list of the type of rebar for each column
- $varDisponibles$ are the available commercial rebar types as a data base consisting of an array vector of length 7 (by default) for which each vector element corresponds to the rebar type diameter
- $nrebar\_cols$ is a vector array of size $ncols$ for which each vector element correspond to the total number of rebars of each column

9.4 Isolated Footings

9.4.1 Function: IsolatedFootings

Purpose: to compute the visualization of prismatic solid rectangular isolated footings on the 3D space. The Y-axis direction is the vertical direction.

Syntax:

$Isolated\_Footings(nfootings, coord\_desp\_footings, entrada\_Footings)$

Description:

Input variables:

- $nfootings, entrada\_Footings$ are the result of using the function *dim_elem* (p. 110)
- *coord_desp_footings* are the base directional axis location of each element, as a result of using the function *design_data_reading* (p. 109)

9.4.2 Function: RebarRectangularIsolatedFootings

Purpose: to compute the visualization of longitudinal prismatic rebar in rectangular isolated footing elements (the X-axis and Z-axis are each of the longitudinal axis).

Syntax:

$Rebar\_Rectangular\_Isolated\_Footings(nfootings, coord\_desp\_footings, entrada\_Footings, ...$
$rebar\_disp\_footings, arrangement\_rebar\_footings, varDisponibles, nrebar\_footings)$

Description:

Input variables:

- $nfootings, entrada\_Footings$ are the result of using the function *dim_elem* (p. 110)
- *coord_desp_footings* are the base directional axis location of each footing element, as a result of using the function *design_data_reading* (p. 109)
- *rebar_disp_footings* are the local coordinates (X,Y) of the rebar over the two transversal cross-section of each footing element
- *arrangement_rebar_footings* is the list of the type of rebar for all rebars of the two transversal cross-sections of each footing
- *varDisponibles* are the available commercial rebar types as a data base consisting of an array vector of length 7 (by default) for which each vector element corresponds to the rebar type diameter
- *nrebar_footings* is a vector array of *nfootings* rows and four columns (two of which correspond to each of the two transversal cross-section of a footing)

10 User's Manual Examples

10.1 Ex1: Design of rectangular beams

Let us consider a reinforced concrete beam of length equal to $500cm$ and cross-section dimensions of $30x60cm$, made of concrete $f'_c = 280\frac{Kg}{cm^2}$ subject to load conditions of $33Ton \cdot m, 29Ton \cdot m, 31Ton \cdot m$ for all of its three main cross-sections along its length (left, middle, right), respectively. A lateral concrete cover of $3cm$ with a high demand of ductility.

MatLab code:

```
clc
clear all

pu_beams=38.6; % unit construction assembly cost of steel reinforcement
duct=3; % high ductility demand
span=500; % cm
b=30; % width (cm)
h=60; % height (cm)

h_rec_sections=[5 5 3 3 5 5]; % [rec_left_up, rec_left_low, rec_mid_up,
                              % rec_mid_low, rec_right_up, rec_right_low]

b_rec=3; % lateral concrete cover
fc=280; % Kg/cm2
fy=4200; % Yield stress of steel reinforcement (Kg/cm2)
load_conditions=[1 -33.0 29.0 -31.0]; %Ton-m
cols_sym_asym_isr="Symmetric";

[sep_bars,b,h,inertia_modif,dispositionBar_Der,barArrangementDerComp,...
    barArrangementDerTens,dispositionBar_Center,barArrangementCentralTens,...
    barArrangementCentralComp,dispositionBar_Izq,barArrangementIzqTens,...
    barArrangementIzqComp,minAreaVar_3sec,Ef_elem_sec_t,bestCostVar,ef_var,...
    minAreaVar_prom,Mr_3section]=beamsISR(pu_beams,span,b,h,h_rec_sections,...
    fc,fy,load_conditions,cols_sym_asym_isr,duct,b_rec)
```

Results:

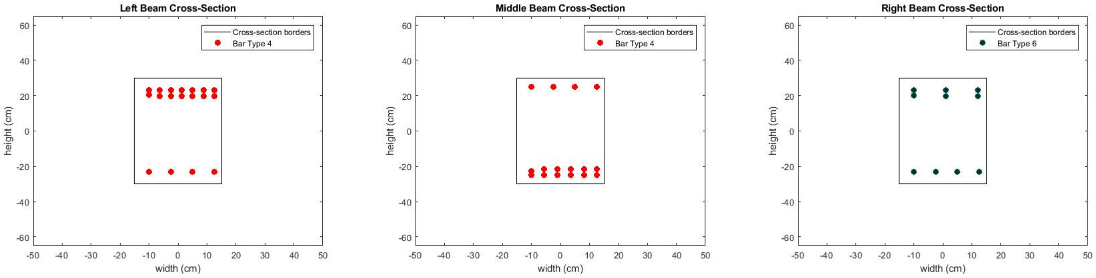


Figure 32: *Reinforced concrete cross-section designs for the beam along its length (left, middle, right), respectively.*

10.2 Ex2: Design of rectangular columns

Let us consider the reinforcement design of a columns with cross-sections $b = 60cm, h = 60cm$ of $height = 400cm$ made of concrete $f'_c = 280\frac{Kg}{cm^2}$ subject to load conditions of $Pu = 15Ton$, $Mu_x = 32Ton \cdot m$, $Mu_y = 8Ton \cdot m$ with concrete cover in the vertical and horizontal cross-section direction of $4cm$. A cross-section *Cracked* mechanisms will be considered in case it is necessary. It is required a high ductile behaviour of the cross-section. No slenderness effects are considered.

10.2.1 Asymmetrical reinforcement

MatLab code:

```
clear all
clc

cols_sym_asym_isr="Asymmetric"; % Distribution of reinforcement
if cols_sym_asym_isr=="Symmetric"
    pu_cols=[29.19, 29.06, 28.93, 28.93, 28.93, 28.93, 28.93]; % symmetric rebar
elseif cols_sym_asym_isr=="Asymmetric"
    pu_cols=[32.23, 32.10, 31.96, 31.96, 31.96, 31.96, 31.96;
            36.0, 36.0, 36.0, 36.0, 36.0, 36.0, 36.0]; % asymmetric rebar

elseif  cols_sym_asym_isr=="ISR"
    pu_cols=[28.93];
end

height=400; % column length or height (cm)
b=60; % width cross-section (cm)
h=60; % height cross-section (cm)
fy=4200; % yield stress of rebars
fc=280; % Kg/cm2
load_conditions=[1 15 32 8]; % [nload, Pu, Mx, My] (Ton-m)
rec=[4 4]; % concrete cover: [coverx covery] (cm)

condition_cracking="Cracked";
ductility=3; % high ducitlity demand on cross-section

% ----------------------------------------------------------------------

[Inertia_xy_modif,b,h,bestArrangement,best_disposicion,cost_elem_col,...
Ac_sec_elem,Ef_sec_col,Mr_col]=isrColumnsSymAsym(pu_cols,height,b,h,rec,...
fy,fc,load_conditions,cols_sym_asym_isr,condition_cracking,ductility);
```

Results:

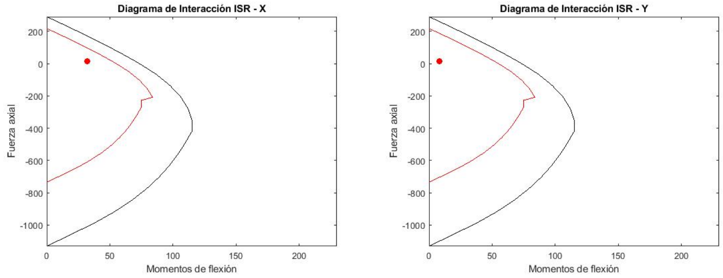


Figure 33: *Optimal reinforced concrete cross-section design of a column. (Left) Optimal ISR Interaction diagram in the X direction, (Right) Optimal ISR Interaction diagram in the Y direction.*

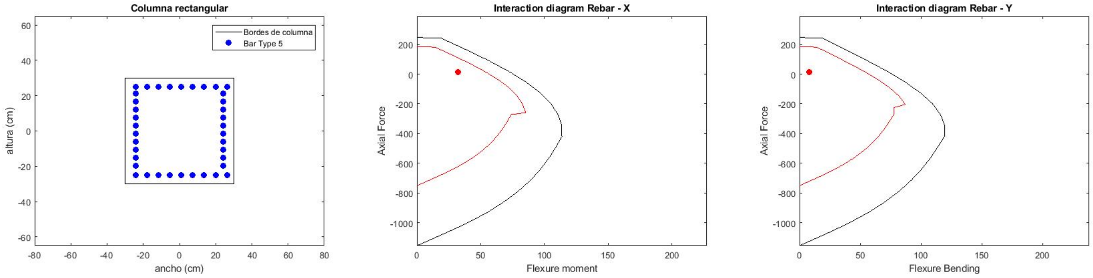


Figure 34: *Optimal reinforced concrete cross-section design of a column. (Left) Optimal Rebar Interaction diagram in the X direction, (Center) reinforced concrete column cross-section, (Right) Optimal Rebar Interaction diagram in the Y direction.*

10.2.2 Symmetric reinforcement

MatLab code:

```
clear all
clc

cols_sym_asym_isr="Symmetric"; % Distribution of reinforcement
if cols_sym_asym_isr=="Symmetric"
    pu_cols=[29.19, 29.06, 28.93, 28.93, 28.93, 28.93, 28.93]; % symmetric rebar
elseif cols_sym_asym_isr=="Asymmetric"
    pu_cols=[32.23, 32.10, 31.96, 31.96, 31.96, 31.96, 31.96;
            36.0, 36.0, 36.0, 36.0, 36.0, 36.0, 36.0]; % asymmetric rebar

elseif  cols_sym_asym_isr=="ISR"
    pu_cols=[28.93];
end

height=400; % column length or height (cm)
```

```
b=60; % width cross-section (cm)
h=60; % height cross-section (cm)
fy=4200; % yield stress of rebars
fc=280; % Kg/cm2
load_conditions=[1 15 32 8]; % [nload, Pu, Mx, My] (Ton-m)
rec=[4 4]; % concrete cover: [coverx covery] (cm)

condition_cracking="Cracked";
ductility=3; % high ducitlity demand on cross-section

% ____________________________________________________________________

[Inertia_xy_modif,b,h,bestArrangement,best_disposicion,cost_elem_col,...
Ac_sec_elem,Ef_sec_col,Mr_col]=isrColumnsSymAsym(pu_cols,height,b,h,rec,...
fy,fc,load_conditions,cols_sym_asym_isr,condition_cracking,ductility);
```

Results:

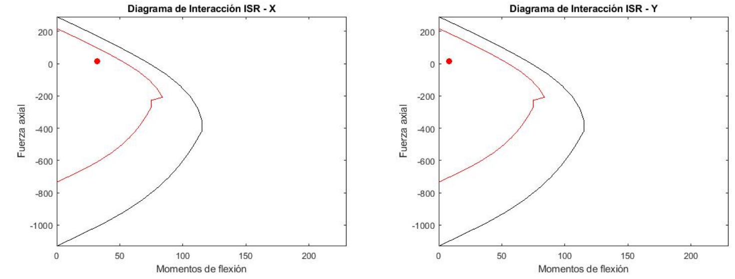


Figure 35: *Optimal reinforced concrete cross-section design of a column. (Left) Optimal ISR Interaction diagram in the X direction, (Right) Optimal ISR Interaction diagram in the Y direction.*

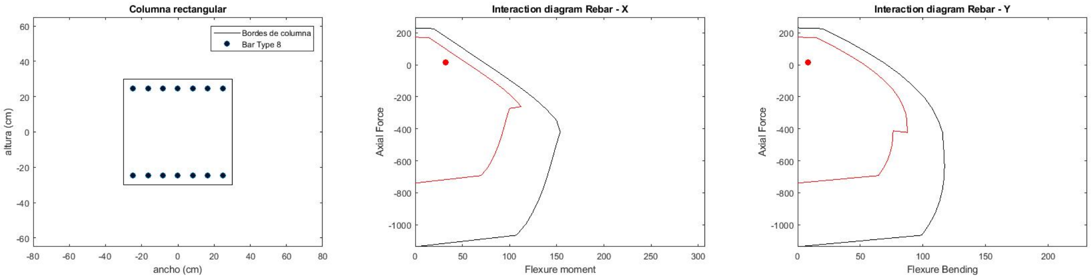


Figure 36: *Optimal reinforced concrete cross-section design of a column. (Left) Optimal Rebar Interaction diagram in the X direction, (Center) reinforced concrete column cross-section, (Right) Optimal Rebar Interaction diagram in the Y direction.*

10.3 Ex3: Design of rectangular isolated footings

Let us consider the reinforcement design of an isolated footing supported by a soil of admissible load equal to $2.5\frac{Kg}{cm^2}$ with a Safety Factor of 1.5, that supports a column of cross-sections $b = 30cm, h = 50cm$ that transmits biaxial loads of $Pu = 20Ton$, $Mu_x = 35Ton \cdot m$, $Mu_y = 24Ton \cdot m$, made of concrete $f'_c = 300\frac{Kg}{cm^2}$, with concrete cover in the vertical and horizontal cross-section direction of $5cm$. It is required a high ductile behaviour of the transversal cross-sections.

MatLab code:

```
clear all
clc

fy=4200; % Yield stress of steel reinforcement (Kg/cm2)
qadm=2.5; % Admissible bearing load of soil
FS=1.5; % Design Safety Factor
qu=qadm*FS; % Design bearing load of soil
pu_steel_footings=26.75; % unit construction cost of reinforcement assembly

he_footing=40; % initial footing height dimensions (cm)
fc=300; %f'c
rec=5; % concrete cover in all directions
bc=30; hc=50;  % transversal dimensions
dimCol=[bc,hc]; % supporting column's dimensions

load_conditions=[1 20 35 24]; % Ton,m
pu=load_conditions(1,2);

% Design of footing dimensions___________________________________________
% Note: in case it is required to design transversal dimensions
[be,le,contact_pressure]=design_dim_footings(pu,qu,dimCol,he_footing,rec);

dim_zap=[be le];

cols_sym_asym_isr="Symmetric";
ductility=3;

[hmodif,m_max_eje,barDispositionFootings,arrangement_bar_footings,...
nbars_footings,AcBar,bestCost_elem,list_ef_footings,list_mr_footings]=...
isrFootings(pu_steel_footings,he_footing,dim_zap(1),dim_zap(2),...
rec,fc,fy,load_conditions,dimCol,cols_sym_asym_isr,ductility)
```

Results:

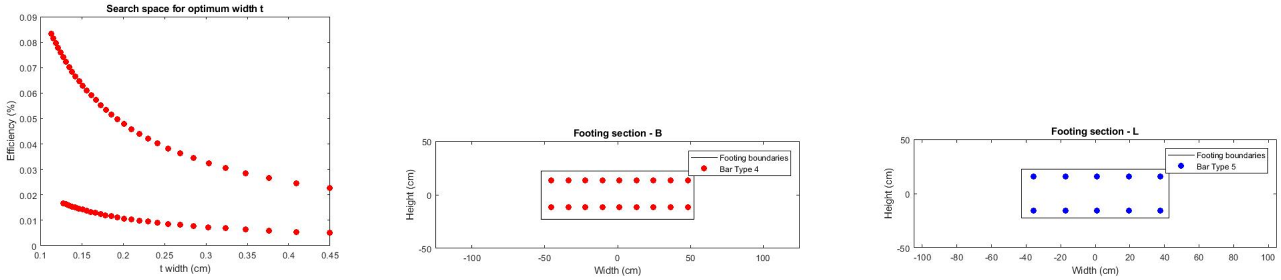


Figure 37: *Optimal reinforced concrete cross-section design of an isolated footing. (Left) Optimal ISR convergence with the SGD method, (Middle) Optimal reinforced transversal cross-section along the L dimension, (Right) Optimal reinforced transversal cross-section along the B dimension.*

10.4 Ex4: Analysis-Design of RC frames and Visual CALRECOD

Let us consider the following structural plane frame of **Fig. 38**. A $f'_c = 300\frac{Kg}{cm^2}$ for all elements (including isolated footings) will be considered. An admissible bearing load capacity of soil of $q_{adm} = 2.5\frac{Kg}{cm^2}$ is considered for the design of the isolated footings, with a Safety Factor $FS = 2.0$.

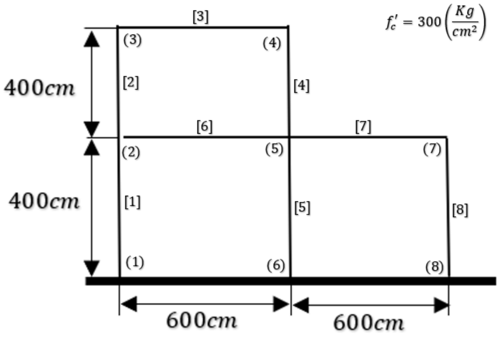


Figure 38: *Topology of the illustrative structural frame.*

The *Inverted Pendulum method* is applied to compute lateral equivalent shear base forces for each floor subject to a ground acceleration of $200\frac{cm}{s^2}$. The initial dimensions as shown in **Fig. 39** and remain constant through the optimization design process with asymmetric reinforcement in columns.

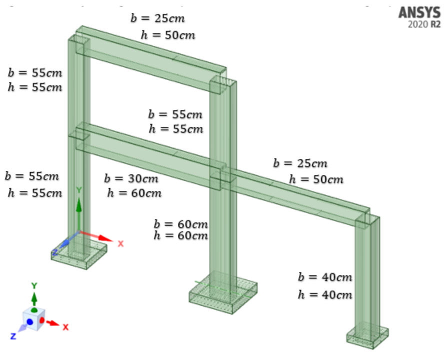


Figure 39: *Dimensions of the structural frame's elements.*

MatLab code: see MatLab .m file $Design_RCPlane_Frames_Ex01.m$ for better reference.

```
clc
clear all

nnodes=8;
nbars=8;

%%%%%%%%%%%---------materials features---------%%%%%%%%%

% f'c for each element:
fpc=[300;
     300;
     300;
     300;
     300;
     300;
     300;
     300];

% Elasticity modulus of each element's material (function of f'c)
Eelem=zeros(nbars,1);
for i=1:nbars
    Eelem(i)=14000*(fpc(i))^0.5;
end

%%%%%%%%%%---------geometrical features---------%%%%%%%%

dimensions=[55 55; % cross-section dimensions of each element
            55 55;
            25 50;
            55 55;
            60 60;
            30 60;
            25 50;
            40 40];
```

```
areaElem=zeros(nbars,1);
inertiaElem=zeros(nbars,1);
for i=1:nbars
    areaElem(i)=dimensions(i,1)*dimensions(i,2);

    inertiaElem(i)=1/12*dimensions(i,1)*dimensions(i,2)^3;
end

%coordinates of each node for each bar
coordxy=[0 -100;
         0 400;
         0 800;
         600 800;
         600 400;
         600 -100;
         1200 400;
         1200 -100];

% coordinates of the beams end' centroids______________________
coordEndBeams=[0 800 0;
               0 400 0;
               600 400 0];

% coordinates of the columns base' centroids______________________
coord_desplante_cols=[0 -100 0;
                      0 400 0;
                      600 400 0;
                      600 -100 0;
                      1200 -100 0];

% coordinates of the footing base' centroids______________________
%_________________________________________________________________

coord_desplante_zapatas=[0 -100 0;
                        600 -100 0;
                        1200 -100 0];

%___________________________________________________________________

%%%---- Initial-final node of each bar --------------------------%%%

ni=[1;2;3;4;5;2;5;7];
nf=[2;3;4;5;6;5;7;8];

lenElem=zeros(nbars,1);
for i=1:nbars
    lenElem(i)=((coordxy(nf(i),1)-coordxy(ni(i),1))^2+(coordxy(nf(i),2)...
        -coordxy(ni(i),2))^2)^0.5;
end
```

```
%  gdl condicion (gdl restringidos) o pre-escritos
bc=[1 0;
    2 0;
    3 0;
    16 0;
    17 0;
    18 0;
    22 0;
    23 0;
    24 0];

% Topology data
Edof=zeros(nbars,7);
for i=1:nbars
    Edof(i,1)=i;
    Edof(i,2)=ni(i)*3-2;
    Edof(i,3)=ni(i)*3-1;
    Edof(i,4)=ni(i)*3;

    Edof(i,5)=nf(i)*3-2;
    Edof(i,6)=nf(i)*3-1;
    Edof(i,7)=nf(i)*3;
end

fglobal=zeros(3*nnodes,1);

type_elem=[1 "Col"; % ID vector to identify beam and column elements
           2 "Col";
           3 "Beam";
           4 "Col";
           5 "Col";
           6 "Beam";
           7 "Beam";
           8 "Col"];

beams_LL=[1 100; % Live Load on beams
          2 100;
          3 100];

% To detect which and how many beam and column elemets there are_______
elem_cols=[];
elem_beams=[];

nbeams=0;
ncols=0;
for j=1:nbars
    if type_elem(j,2)=="Beam"
        nbeams=nbeams+1;
        elem_beams=[elem_beams,j];
    elseif type_elem(j,2)=="Col"
        ncols=ncols+1;
        elem_cols=[elem_cols,j];
    end
end

 np=7; % number of points of analysis for the computation of mechanic
       % elements for each structural element

pu_beams=38.85; % unit construction cost of reinforcement assembly
cols_sym_asym_isr="Asymmetric";
if cols_sym_asym_isr=="Symmetric"
    pu_cols=[29.19, 29.06, 28.93, 28.93, 28.93, 28.93, 28.93]; % symmetric rebar
elseif cols_sym_asym_isr=="Asymmetric"
    pu_cols=[32.23, 32.10, 31.96, 31.96, 31.96, 31.96, 31.96;
          36.0, 36.0, 36.0, 36.0, 36.0, 36.0, 36.0]; % asymmetric rebar

elseif  cols_sym_asym_isr=="ISR"
    pu_cols=[28.93];
end

 pu_steel_footings=26.75;

fcbeams=300;
fccols=300;
fc_footing=300;

qadm=2.5; % Admissible bearing load of soil (Kg/cm2)
FS=2.0; % Design Safety Factor for isolated footings
nodos_apoyo_columna=[1 6 8; %apoyo
                    1 5 8]; %elemento columna
nfootings=length(nodos_apoyo_columna(1,:));
nfloors=2;
floor_elem=[1 1 5 6 7 8; % elements that belong to each floor
            2 2 3 4 0 0]; % (see documentation)


%%%%%------------------ Optimization steel of frames...................

%--------------dinamic análysis (inertial forces).......................

support=[1 "Fixed" "Fixed"; % Type of support at the ends of
       2 "Fixed" "Fixed";   % each element
       3 "Fixed" "Fixed";
       4 "Fixed" "Fixed";
       5 "Fixed" "Fixed";
       6 "Fixed" "Fixed";
       7 "Fixed" "Fixed";
       8 "Fixed" "Fixed"];

% Indicate cracking condition in column sections "Cracked" or "Non-cracked"
condition_cracking="Cracked";
%__________________________________________________________________________
```

```
unit_weight_elem=zeros(nbars,2);
for i=1:nbars
    unit_weight_elem(i,2)=0.0024; % kg/cm3
end

vertical_loads=zeros(nbars,2);
for i=1:nbeams
    vertical_loads(elem_beams(i),2)=1.1*beams_LL(i,2);
end

fprintf('\nRC FRAME OPTIMIZATION DESIGN\n\n');

% To consider the self weight load as a distributed load in beams
qbarra_y=zeros(nbars,2);
for i=1:nbeams
    qbarra_y(elem_beams(i),2)=1.1*areaElem(elem_beams(i))*...
        unit_weight_elem(elem_beams(i),2)+1.1*(beams_LL(i,2));

end

% To consider self weight of columns as punctual vertical loads on its
% supporting nodes
dof_weight_elem_cols=[2 5 14 17 23; % dof
                      1 2 4  5  8]; % cols

for i=1:length(dof_weight_elem_cols(1,:))
    fglobal(dof_weight_elem_cols(1,i))=-1.1*areaElem(dof_weight_elem_cols(2,i))*...
        unit_weight_elem(dof_weight_elem_cols(2,i),2)*lenElem(dof_weight_elem_cols(2,i));

end

%%% To compute lateral equvalent seismic forces..........................
[fmax_floor,modals]=SeismicLoadsNFloor2DFrame(support,unit_weight_elem,...
                type_elem,lenElem,areaElem,Eelem,inertiaElem,vertical_loads,...
                nfloors,floor_elem);

dof_seismic_forces=[4 7];
j=length(dof_seismic_forces);
k=0;
for i=1:length(dof_seismic_forces)
    j=length(dof_seismic_forces)-k;
    fglobal(dof_seismic_forces(i))=1.1*fmax_floor(j);
    k=k+1;
end

%%%-----------------------------------------------------------------
ductility=3; % required ductility on the element's cross-sections

recxy_cols=[4 4];
% To export results
```

```
directionData='C:\Users\';

%%% Structural Optimization Design Processan............................
[total_weight_structure,wsteel_cols_total,pac_prom_cols,sectionRestrictions,...
Mp,final_dimensions,displacements,unit_weight_elem,wsteel_conc_beams_elem,...
wsteel_conc_cols_elem,wsteel_conc_footings_elem,he_footings,dim_zap,...
total_cost_structure,inertiaElem,wsteel_structure]=DesignRCPlaneFrameBCIF...
(nnodes,bc,ni,nf,Edof,coordxy,pu_beams,type_elem,pu_cols,nbars,np,...
lenElem,coord_desplante_cols,fpc,inertiaElem,qadm,FS,nodos_apoyo_columna,...
pu_steel_footings,dimensions,Eelem,fcbeams,fccols,fc_footing,fglobal,...
qbarra_y,areaElem,dof_seismic_forces,floor_elem,cols_sym_asym_isr,...
condition_cracking,ductility,elem_cols,elem_beams,recxy_cols,directionData);

% Additional exportation of results for the use of VISUAL-CALRECOD
nombre_archivo='coord_comienzo_beams.txt';
fileid=fopen([directionData,nombre_archivo],'w+t');
for i=1:nbeams
    fprintf(fileid,'%.2f %.2f %.2f\n',coordEndBeams(i,:));
end
fclose(fileid);

nombre_archivo='coord_desplante_columns.txt';

fileid=fopen([directionData,nombre_archivo],'w+t');
for j=1:ncols
    fprintf(fileid,'%.2f %.2f %.2f\n',coord_desplante_cols(j,:));
end
fclose(fileid);

nombre_archivo='coord_desplante_footings.txt';

fileid_05=fopen([directionData,nombre_archivo],'w+t');
for i=1:nfootings
    fprintf(fileid_05,'%.2f %.2f %.2f\n',coord_desplante_zapatas(i,:));
end
fclose(fileid_05);
```

Results:
Detailed results with the usage of VISUAL-CALRECOD are as following **Fig. 40**:

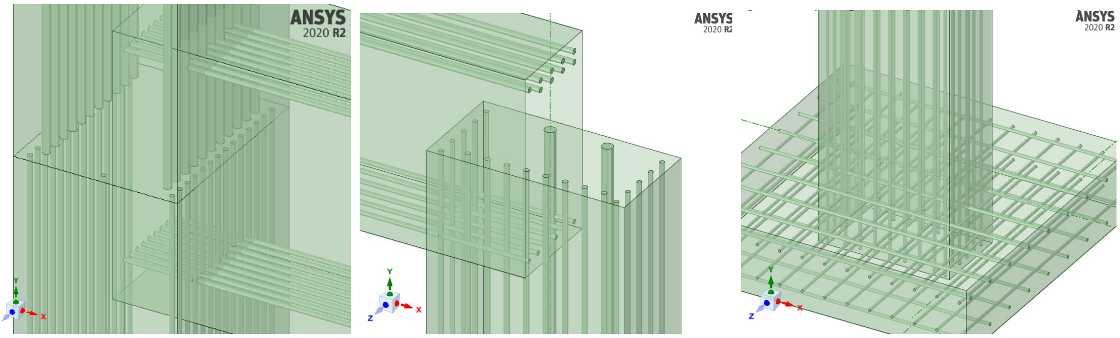


Figure 40: *(Left) Detailed reinforcement in the node (2), (Middle) Detailed reinforcement in node (7), (Right) Detailed reinforcement in node (6).*

10.5 Ex5: Optimal reinforcement area in structural elements

Let us consider the beam cross-section of dimensions $b = 20cm, h = 40cm$, concrete cover along the width dimension of $4cm$ and concrete cover along the height dimension of $3cm$, a $f'_c = 280\frac{Kg}{cm^2}$ will be used. Only one cross-section will be analysed under the load condition of $15Ton \cdot m$ of pure flexure. A requirement of high ductility over the cross-section is considered. Results yield an optima ISR as shown in **Fig. 41**, with an optima ISR width of $1.01cm$ and a corresponding reinforcement area of $12.18cm^2$, with a structural efficiency of $0.989 = 98.9\%$. As it is observed, the optima convergence depicts how the structural efficiency varies as the reinforcement quantity changes, which is very useful. The **function SGD1tBeamsISR** was used.

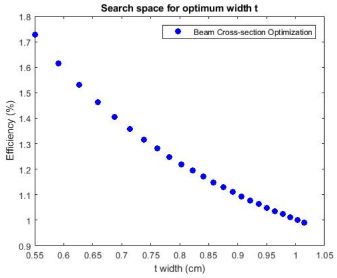


Figure 41: *Convergence of the optimal width t of the ISR in relation with the structural efficiency of the beam cross-section.*

10.6 Ex6: Structural efficiency of a design proposal for a column cross-section

Let us consider the column cross-section of dimensions $b = 40cm, h = 50cm$ reinforced with rebars of $No.\#6$ of diameters $\frac{6}{8}inches$ distributed symmetrically over the cross-section (3 rebars along the width dimension and 4 rebars along the height dimension), width a concrete cover of $4cm$ over all four section boundaries, using concrete of $f'_c = 280\frac{Kg}{cm^2}$. Results yield a structural efficiency of $1.48 = 148\%$ with neutral axis depths of $[11.24cm, 7.26cm]$ for X and Y axis directions, respectively. The reinforced cross-section plot and interaction diagrams are shown in **Fig. 42**, indicating that the design is not efficient, and more reinforcement should be used. The **function RebarDisposition** was used to compute the rebar local coordinates over the cross-section, **function diagramasDisposicion** to compute the interaction diagrams and structural efficiency, and **function diagramsFinalRebarCols** was deployed to plot the design results.

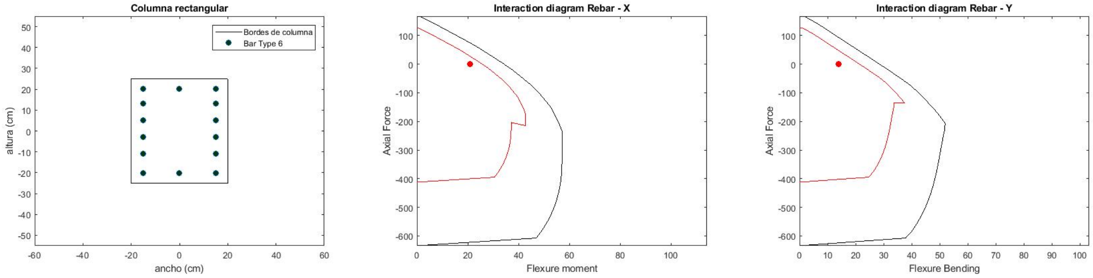


Figure 42: *(Left) Reinforced concrete cross-section design of the column. (Middle) Interaction diagram in the X direction, (Right) Interaction diagram in the Y direction.*

References

[Dahlblom et al., 1986] Dahlblom, O., Peterson, A. and Petersson, H. (1986). CALFEM — a program for computer-aided learning of the finite element method, Engineering Computations, Vol. 3 No. 2, pp. 155-160. DOI: https://doi.org/10.1108/eb023653

[CALFEM, 2004] CALFEM - A finite element toolbox to MATLAB, Version 3.4, Structural Mechanics, Sweden, 2004. Available: https://www.byggmek.lth.se/english/calfem/

[Verduzco, 2021] L. Verduzco Martínez, (2021). The Idealized Smeared Reinforcement (ISR) method for the optimization of concrete sections: a survey of the state-of-the-art and analysis of potential computational approaches, Revista Mexicana de Métodos Numéricos. Vol. 5, (1). DOI: https://www.scipedia.com/public/Verduzco_Martinez_2021a

[ACI 318-19] American Concrete Institute ACI 318-2019, Building Requirements for structural concrete and commentary, 2019

[Chapra & Canale, 2015] Chapra S., Canale R., (2015). Numerical methods for engineering, 7th Edition, McGraw Hill

[Wahde, 2008] Wahde M., (2008). Biologically inspired Optimization Methods. An Introduction, Chalmers University of Technology, Sweden, WIT-PRESS

[Raynolds, 1987] Raynolds, C.W., Flocks, (1987). Herds and Schools: A distributed behavioural model, Computer Graphics, 21,(4)

[Kennedy & Eberhart, 1995] Kennedy, J., (1995). Eberhart, R., Particle Swarm Optimization, IEEE