CAL-RECOD
A Reinforced Concrete Design Toolbox

Luis Fernando Verduzco Martínez
Jaime Moisés Horta Rangel

Copyright ©by the Faculty of Engineering, UAQ, Santiago de Queretaro, Mexico, 2023

No part of this manual may be photocopied or reproduced in any form without the prior consent by the Faculty of Engineering of the Autonomous University of Querétaro.

©Copyright 2022 by the Faculty of Engineering at Autonomous University of Queretaro. All rights reserved.
CAL-RECOD is a trademark of the Faculty of Engineering, Autonomous University of Queretaro.
MatLab is the trademark of The MathWorks, Inc.

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 [1],[2]. 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 second version of CAL-RECOD. The main software consists of MatLab functions (.m-files) both numerical and graphical ones, although as a complementary library called Visual CALRECOD was developed as well, consisting entirely of Dynamo programs (.dyn) in python language for the visualization of the designs as described further in this manual. In the first realese, the Visual CALRECOD consisted of .scscript files (python scripts for ANSYS SpaceClaim), however, due to the greater advantages that could be obtained in Dynamo in relation to SpaceClaim (such as its direct linkage to CAD software from Autodesk, such as AutoCAD, Revit or Robot Structural) it was decided by the authors to focus only on Dynamo scripts and programs for the further development of Visual CALRECOD. 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, April 2023

Contents

1 Introduction **9**

1.1 The Idealized Smeared Reinforcement analogy 9

1.2 Rebar optimization 9

1.3 Design of structural frames 12

1.4 Resume 12

2 Optimization methods and algorithms **13**

2.1 The Steepest Gradient Descent method 13

2.2 The Particle Swarm Optimization method 14

3 Structural Mechanics functions **15**

3.1 Function: casoConcreto 15

3.2 Function: casoConcretoRecRot 17

3.3 Function: casoConcretoCirc 20

3.4 Function: casoConcretoTsec 23

3.5 Function: InertiaBeamCrackedSection 25

3.6 Function: AmpMomSlenderColumns 26

3.7 Function: CrackingColumnsSym 28

3.8 Function: CrackingColumnsAsym 31

3.9 Function: RealPressuresFoot 34

3.10 Function: shearFootings 37

3.11 Function: MomentDistributionFootings 40

3.12 Function: designDimFootings 41

3.13 Function: MomAmpColsGeomNL 43

3.14 Function: shearDesignBeams 45

4 ISR Optimization functions **47**

4.1 ISR analysis for rectangular beams 47

4.1.1 Function: eleMecanicos2tBeams 47

4.1.2 Function: bisectionMr2tBeams 48

4.1.3 Function: Efrec2tbeams 51

4.1.4 Function: SGD1tBeamsISR 52

4.2 ISR analysis for T beams 54

4.2.1 Function: SGD1tTBeamsISR 54

4.2.2 Function: Efrec2tBeamsT 56

4.2.3 Function: bisectionMr2tBeamsT 58

4.2.4 Function: eleMecanicos2tBeamsT 60

4.3 ISR analysis for rectangular columns 62

4.3.1 Function: bisectionMr4t 62

4.3.2 Function: bisectionMrAnalytISR 63

4.3.3 Function: eleMecanicos4t 64

4.3.4 Function: eleMecISRAnalyt 66

4.3.5 Function: widthEfficiencyCols 68

4.3.6 Function: isrColumns 70

4.4 ISR analysis for circular columns 72

4.4.1 Function: isrCircCols 72

4.4.2 Function: diagCircColsISR 74

4.4.3 Function: bisectionMrCircISR1t 76

4.4.4 Function: eleMecanicosISRCirc 77

4.5 ISR analysis for isolated footings 79

4.5.1 Function: EvaluateISR1tFoot . . . 79
4.5.2 Function: SGD1tFootISR . . . 81

5 Element rebar optimization functions **82**
5.1 Rebar analysis for rectangular beams . . . 82
5.1.1 Function: ISR1tRebarBeamsOptimization . . . 82
5.1.2 Function: RebarOptimalDesignBeams . . . 84
5.1.3 Function: EfcriticalRebarbeams . . . 87
5.1.4 Function: bisectionMrRebarBeams . . . 88
5.1.5 Function: eleMecanicosRebarBeams . . . 89
5.1.6 Function: RebarDisposition1tBeams . . . 90
5.1.7 Function: EvaluateCostbeams . . . 91
5.1.8 Function: beamsISR . . . 92
5.1.9 Function: ExportResultsBeam . . . 95
5.2 Rebar analysis for T beams . . . 96
5.2.1 Function: ISR1tRebarTBeamsOptim . . . 96
5.2.2 Function: EfRebarTBeams . . . 98
5.2.3 Function: bisectionMrRebarTBeams . . . 100
5.2.4 Function: eleMecanicosRebarTBeams . . . 102
5.3 Rebar analysis for rectangular columns . . . 104
5.3.1 Function: ExportResultsColumn . . . 104
5.3.2 Function: isrColumnsSymAsym . . . 105
5.3.3 Function: effRecColsLinearSearch . . . 107
5.3.4 Function: effRecColsDoubleDirecLS . . . 109
5.3.5 Function: effColsRot1DirecLS . . . 111
5.3.6 Function: rotReCol2 . . . 113
5.3.7 Symmetrical reinforcement . . . 115
5.3.8 Function: optimalrebarColsSym . . . 115
5.3.9 Function: optimRebarColsSym2Pack . . . 117
5.3.10 Function: superOptimalRebarSym . . . 119
5.3.11 Function: superOptimalRebarSym2Pack . . . 121
5.3.12 Function: sym2typeRebar . . . 123
5.3.13 Function: RebarDisposition . . . 125
5.3.14 Function: RebarDisposition2packSym . . . 126
5.3.15 Function: diagramasDisposicion . . . 127
5.3.16 Function: bisectionMrSymRebarCols . . . 129
5.3.17 Function: eleMecanicosRebarCols . . . 130
5.3.18 Function: diagramRColumnSymRebar . . . 131
5.3.19 Asymmetrical reinforcement . . . 132
5.3.20 Function: dispositionRebarAsymmetric . . . 132
5.3.21 Function: RebarDisposition2packAsym . . . 134
5.3.22 Function: EvaluateAsymmetric . . . 135
5.3.23 Function: EvalAsymDoubleDirection . . . 137
5.3.24 Function: bisectionMrVarAsymm . . . 139
5.3.25 Function: bisectionMrVarAsymRot . . . 141
5.3.26 Function: eleMecanicosVarAsymm . . . 145
5.3.27 Function: eleMecBarAsymRecRot . . . 147
5.3.28 Function: PlastiCenterAxis . . . 149
5.3.29 Function: DiagramsAsymmetricRebar . . . 150
5.3.30 Function: AssembledRebarSymAsym2 . . . 152
5.3.31 Function: AssembledRebarSymAsym2packs . . . 154

5.3.32 Function: asym1typeRebar . . . 156
5.3.33 Function: Asym1Diam . . . 158
5.3.34 Function: asym1typeRebar2pack . . . 160
5.3.35 Function: Asym2pack1Diam . . . 162
5.3.36 Function: Asym2pack4Diam . . . 164
5.3.37 Function: Asym2packSym4Diam . . . 166
5.3.38 Function: asymSym4Diam . . . 168
5.3.39 Function: AsymmSym4Diam . . . 170
5.3.40 Function: Asym4Diam . . . 172
5.3.41 Function: InteractionDiagramAxis . . . 174
5.3.42 Function: multiDiagAxisColRec . . . 176
5.3.43 Function: InteracSurfaceColRec . . . 178
5.3.44 Function: CostDimCurveOptimDesignCols . . . 180
5.3.45 Function: CostDimCurveOptimDesignBeam . . . 182
5.4 Rebar analysis for circular columns . . . 184
5.4.1 Function: ExportResultsColumnCirc . . . 184
5.4.2 Function: effCircColsLS . . . 185
5.4.3 Function: RebarDispositionCirc . . . 187
5.4.4 Function: optimalRebarCirc . . . 188
5.4.5 Function: diagCircColsRebar . . . 190
5.4.6 Function: bisectionMrRebarColsCir . . . 192
5.4.7 Function: eleMecanicosRebarColsCirc . . . 193
5.5 Rebar analysis for isolated footings . . . 195
5.5.1 Function: RebarOptionsFootings . . . 195
5.5.2 Function: dispositionRebarSquareFootings . . . 196
5.5.3 Function: dispositionRebarRectangularFootings . . . 197
5.5.4 Function: EfcriticalFootings . . . 199
5.5.5 Function: EvaluateCostRebarFoot . . . 200
5.5.6 Function: EvaluateCostISRFoot . . . 201
5.5.7 Function: isrFootings . . . 202
5.5.8 Function: ExportResultsIsolFootings . . . 204

6 Design-Analysis of 2D frames **205**
6.0.1 Function: DesignRCPlaneFrameBCI . . . 205
6.0.2 Function: PlaneFrameStaticLinearAnalysis . . . 208

7 Special structures **210**
7.1 Retaining Walls . . . 210
7.1.1 Function: RetainingRCWall . . . 210
7.1.2 Function: DesignRetainingWallPSO . . . 214

8 Graphic functions **217**
8.1 Reinforcement of beam cross-sections . . . 217
8.1.1 Function: beamReinforcedSection . . . 217
8.1.2 Function: MidLeftRightBeamReinforcedSections . . . 218
8.1.3 Function: TbeamReinforcedSection . . . 219
8.2 Interaction diagrams and rebar layout of column cross-sections . . . 220
8.2.1 Function: diagramISR . . . 220
8.2.2 Function: plotdiagramCircISR . . . 221
8.2.3 Function: diagramsFinalRebarCols . . . 222
8.2.4 Function: plotdiagramCircRebar . . . 223

8.2.5 Function: PlotRotRecSecRebarCols . . . 224
8.2.6 Function: diagDoubleDirecAsymRebarCols . . . 225
8.3 Reinforcement of isolated rectangular footing cross-sections . . . 226
8.3.1 Function: ReinforcedSectionsFooting . . . 226
8.4 Retaining walls . . . 227
8.4.1 Function: plotRCWallDesign . . . 227

9 Structural Frame System Functions **228**
9.0.1 Function: WeightStruc . . . 228
9.0.2 Function: CostStruc . . . 230

10 Visual CALREDOC **231**
10.1 Dynamo: Beams . . . 231
10.1.1 Program: RebarRectangBeams . . . 231
10.2 Dynamo: Rectangular columns . . . 235
10.2.1 Program: RebarRectangColumns . . . 235
10.3 Dynamo: Circular columns . . . 238
10.3.1 Program: RebarCircularColumns . . . 238
10.4 Dynamo: Rectangular isolated footings . . . 241
10.4.1 Program: RebarIsolFootings . . . 241
10.5 Dynamo: 2D RC frames . . . 243
10.5.1 Program: RebarPlaneFramesBCIF . . . 243

11 User's Manual Examples **244**
11.1 Ex1: Structural efficiency of a rebar reinforced T-beam cross-section . . . 244
11.2 Ex2: Structural efficiency of a rebar reinforced rectangular column through a 3D analysis with interaction surfaces . . . 246
11.3 Ex3: Comparison between a 3D-rotation analysis and the Breler's formula/Load contour method . . . 250
11.4 Ex4: Structural efficiency of an asymmetrically reinforced column cross-section subject to compression and positive bending moments . . . 254
11.5 Ex5: Interaction diagrams in both directions of an asymmetrically reinforced rectangular column cross-section . . . 257
11.6 Ex6: Amplification of moments for asymmetrically reinforced slender concrete columns . . . 260
11.7 Ex7: Amplification of moments for symmetrically reinforced slender concrete columns . . . 266
11.8 Ex8: Interaction Surfaces of rectangular reinforced concrete column cross-sections . . . 271
11.9 Ex9: Structural efficiency of an symmetrically reinforced column cross-section subject to biaxial bending-compression . . . 274
11.10 Ex10: Optimization design of a rectangular reinforced concrete column . . . 277
11.11 Ex11: Optimization of asymmetrical design of a rectangular concrete column with packages of two rebars - Problem 1 . . . 280
11.12 Ex12: Optimization of asymmetrical design of a rectangular concrete column with packages of two rebars - Problem 2 . . . 283
11.13 Ex13: Required increment in reinforcing area in tension . . . 286
11.14 Ex14: Structural efficiency of a retaining RC wall . . . 289
11.15 Ex15: Optimization design of a retaining RC wall . . . 292
11.16 Ex16: Optimal rebar design of a T-beam cross-section . . . 295
11.17 Ex17: Approximate optimal reinforcing area in tension of a rectangular beam cross-section . . . 298
11.18 Ex18: Comparison of resistance of a T-beam and a rectangular beam with the variation of the neutral axis depth . . . 300
11.19 Ex19: Structural efficiency of a rebar reinforced rectnagular beam cross-section . . . 303
11.20 Ex20: Optimal rebar design of a rectangular-beam element . . . 306

11.21Ex21: Plastic Center of an asymmetrically reinforced rectangular column 309
11.22Ex22: Distribution of soil pressures in a rectangular isolated footing 313
11.23Ex23: Optimal design of rebar for a rectangular isolated footing 316
11.24Ex24: Optimal design of rebar for a circular column 319
11.25Ex25: Modified cross-section inertia of a symmetrically reinforced column 322
11.26Ex26: Cost-Dim curve of rectangular concrete beams 326
11.27Ex27: Cost-Dim curve of rectangular concrete columns 328
11.28Ex28: Optimal design of reinforcement of a concrete plane frame subject to seismic forces 330

12 References **340**

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 Idealized Smeared Reinforcement analogy

The optimization design processes are based on the ISR analogy (Idealized Smeared Reinforcement) [3] using the Steepest Gradient Descent (SGD) Method (for a rapid determination of a required reinforcement area) **Fig. 2** given the concave form of the structural resistance efficiency of structural element's cross-section in function of the ISR's width t or reinforcement area, when only one width t variable applies (see **Fig. 2**).

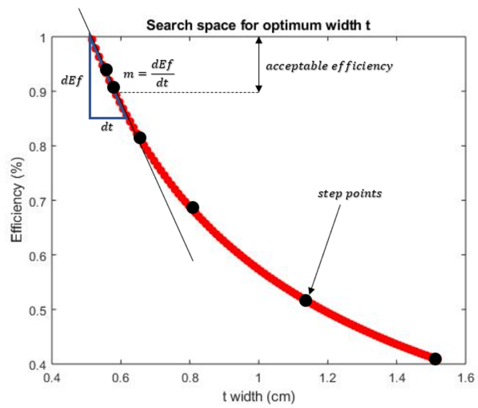


Figure 1: *Concave curve of structural resistance efficiency of a reinforced concrete element's cross-section in function of the ISR's width t or reinforcement area when only one width t variable problem applies.*

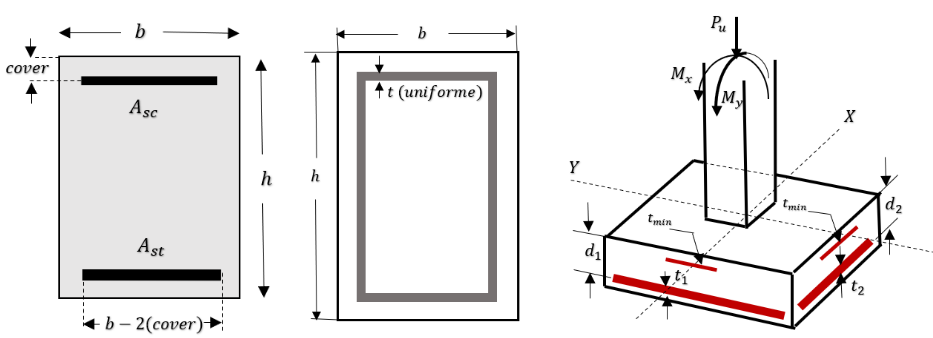


Figure 2: *ISR analogy for each type of structural element: (Left) Beam cross section, (Middle) Column cross-section, (Right) Isolated footing elements.*

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), each cross-section designed independently from one another.

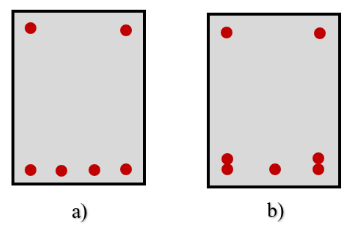


Figure 3: *Reinforcement bar options for a beam cross-section.*

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. 4** when minimum separation restrictions do not suffice for the option of individual rebars.

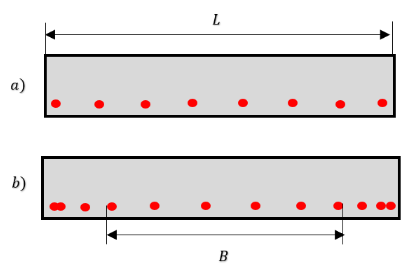


Figure 4: *Reinforcement bar options for a footing cross-section.*

For rectangular isolated footings the longitudinal bars are distributed according to the ACI-318 and NTC-17 design codes as shown in **Fig. 5**, uniformly in the smaller cross-section and non-uniformly in the longer one:

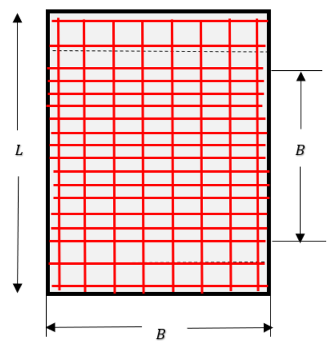


Figure 5: *Longitudinal rebar distribution over a rectangular isolated footing element.*

As for columns, several algorithms and design approaches are integrated. There are functions to design reinforcement in rectangular column cross-sections both in an asymmetrical or symmetrical fashion, either in individual rebars or in packages of two rebars. For all design approaches a Linear-Search method is used to determine the optimal design (with the lowest reinforcement area or volume).

For symmetrical reinforcement there are two options, either using only one rebar diameter or two **Fig. 6**

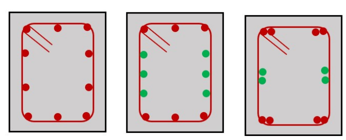


Figure 6: *Possibilities of symmetrical reinforcement designs for rectangular columns, either with one or two rebar diameters, in individual rebars or in packages of two rebars.*

On the other hand, for asymmetrical designs as many as four rebar diameters may be allowed to be placed simultaneously over a reinforce concrete section **Fig. 7**.

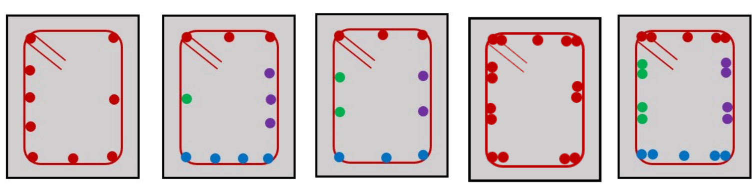


Figure 7: *Possibilities of reinforcement for an asymmetrical rebar configuration, either with one rebar diameter in individual rebars or in packages of two.*

Design and analysis functions for circular concrete columns are also part of this new release of CALRECOD, from which only symmetrical designs with only one rebar diameter are possible to design (**Fig. 8 (left)**). When designing optimally such structures the ISR analogy also applies to determine an initial optimal reinforcement area (see **Fig. 8 (right)**).

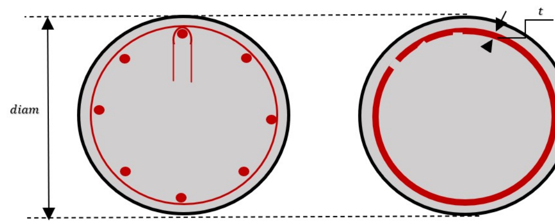


Figure 8: *The left panel of this figure shows the possibility of reinforcement designs that the CALRECOD functions can perform, that is, consisting only of symmetrical designs with one rebar diameter. The right panel shows the ISR for circular concrete sections used to determine initial optimal reinforcement areas or quantities for circular column cross-sections.*

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 and design of 2D frames rather than 3D. The functions hereby presented are able to optimally design the rebar on each structural element composing the structural frame (including isolated footings), either with symmetrical rebar in columns or asymmetrical (according to the user's preferences).

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 [7] and then improved by Eberhart and J.Kennedy [8] 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 functions

3.1 Function: casoConcreto

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

Syntax:

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

System of units: Any.

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 [4] code
- b, h are the cross-section dimensions

Theory:

Is considered that concrete only withstand compression forces, therefore, the zone in tension **Fig. 9** 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 (4).

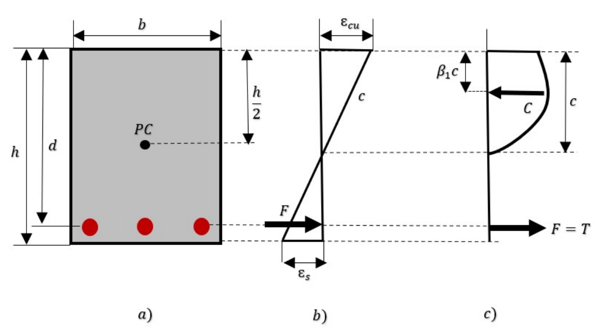


Figure 9: *Concrete compression zone of a rectangular reinforced concrete cross-section subject to bending forces.*

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

The bending resistance (5), on the other hand, assumes that the Plastic Centroid of the cross-section is in the same position as the Geometric Centroid **Fig. 9** (at the depth of $\frac{h}{2}$).

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

3.2 Function: casoConcretoRecRot

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

Syntax:

$$elemConc = casoConcretoRecRot(a, fdpc, b, h, RotCornerSec, rotCP, gamma)$$

System of units: Any.

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 [4] code
- b, h are the cross-section dimensions
- $RotCornerSec$: are the cross-section corner coordinates. Size: 4 x 2, in format: $[x_i, y_i]$
- rotCP: are the Plastic Center depth values of the rotated cross-section
- $gamma$: is the angle of rotation of the cross-section

Theory:

The analysis to determine such contribution in resistance of the compression concrete zone is divided in four cases according to the degree of rotation of the cross-section, as shown in **Fig. 10**. For each case, the integration trapezoidal rule is used to calculate the total area of the concrete zone in compression (3):

$$I = \frac{(b-a)}{2n}[f(x_0) + 2\sum_{i=1}^{n-1} f(x_i) + f(x_n)] \tag{3}$$

Case 1: $0 \leq \gamma < 90$

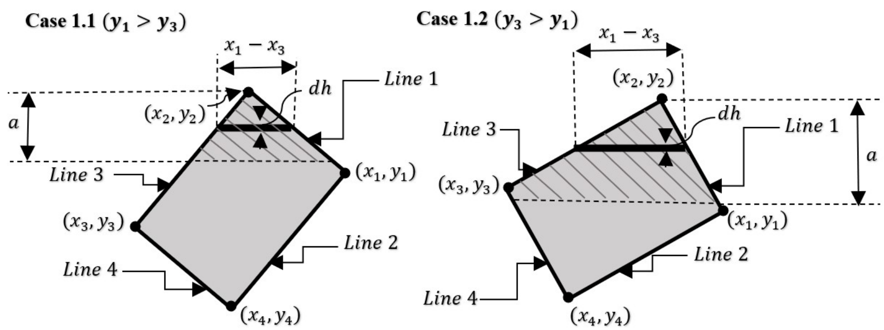


Figure 10: *Case 1: concrete compression zone of a rotated rectangular cross-section subject to bending forces.*

Case 2: $90 \leq \gamma < 180$

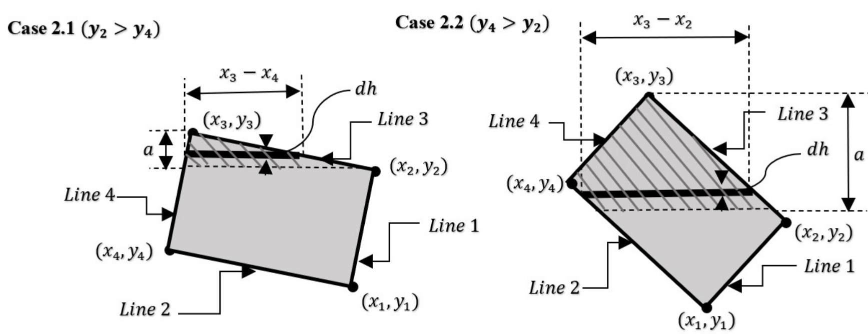


Figure 11: *Case 2: concrete compression zone of a rotated rectangular cross-section subject to bending forces.*

Case 3: $180 \leq \gamma < 270$

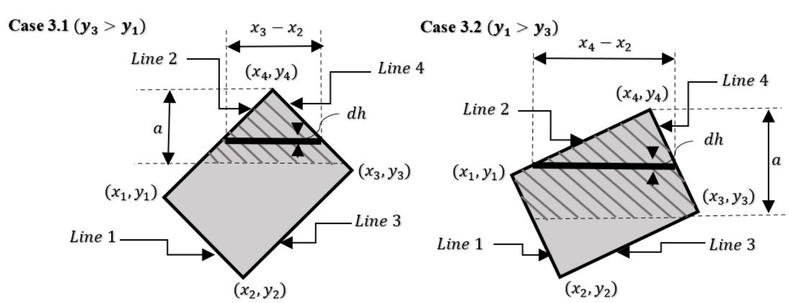


Figure 12: *Case 3: concrete compression zone of a rotated rectangular cross-section subject to bending forces.*

Case 4: $270 \leq \gamma < 360$

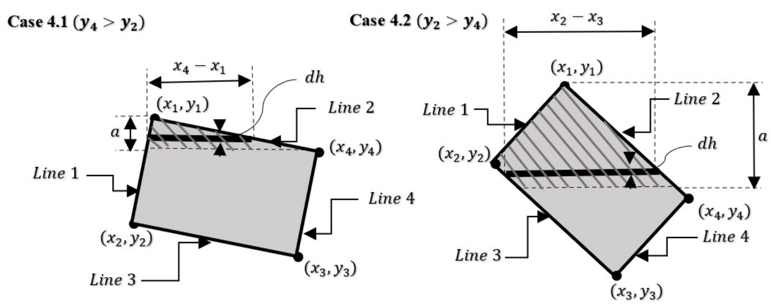


Figure 13: *Case 4: concrete compression zone of a rotated rectangular cross-section subject to bending forces.*

Thus, the resistance of the concrete zone in compression against axial loads is given (4), whereas the resistance against bending moment as (5), where CP is given in the input of the function as $rotCP$:

$$F_c = C = A_{co} \cdot 0.85 \cdot f'_c \tag{4}$$

$$M_c = F_c \cdot (CP - y_c) \tag{5}$$

3.3 Function: casoConcretoCirc

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

Syntax:

$$elemConc = casoConcretoCirc(a, fdpc, diam)$$

System of units: Any.

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 [4] code
- $diam$ is the cross-section diameter

Theory:

Is considered that concrete only withstands compression forces, therefore, the zone in tension **Fig. 14** 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 be cracked and it is assumed that the Plastic Centroid of the cross-section is in the same position as the Geometric Centroid **Fig. 14** (at a depth of $+diam/2$):

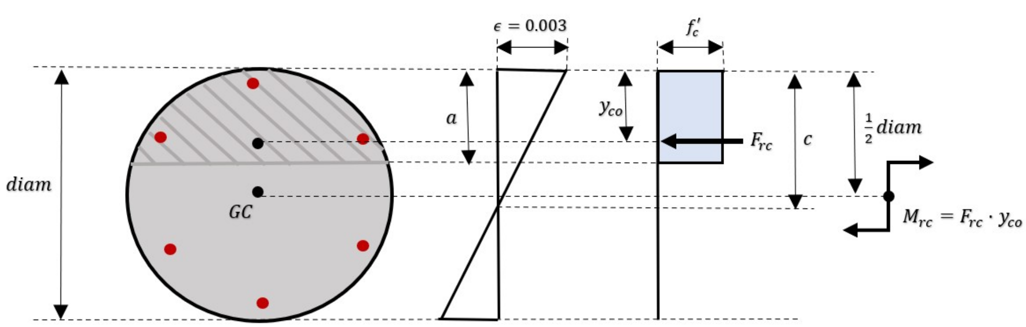


Figure 14: *Concrete compression zone of a circular concrete cross-section subject to bending forces.*

The analysis approach of such resistance contribution of the concrete compression zone can be divided in two cases: when $a < 1/2 \cdot diam$ and when $a \geq 1/2 \cdot diam$ as following:

When $a < \frac{diam}{2}$:

$$F_{rc} = -0.85 f'_c \cdot \int_{-x(a)}^{x(a)} (a - y_{co}(x)) \cdot dx \tag{6}$$

Where the function $y_{co}(x)$ is defined by the equation of a circumference in the plane of reference of **Fig. 15 (Left)** as (7):

$$y_{co}(x) = -\sqrt{(\frac{diam}{2})^2 - x^2} \tag{7}$$

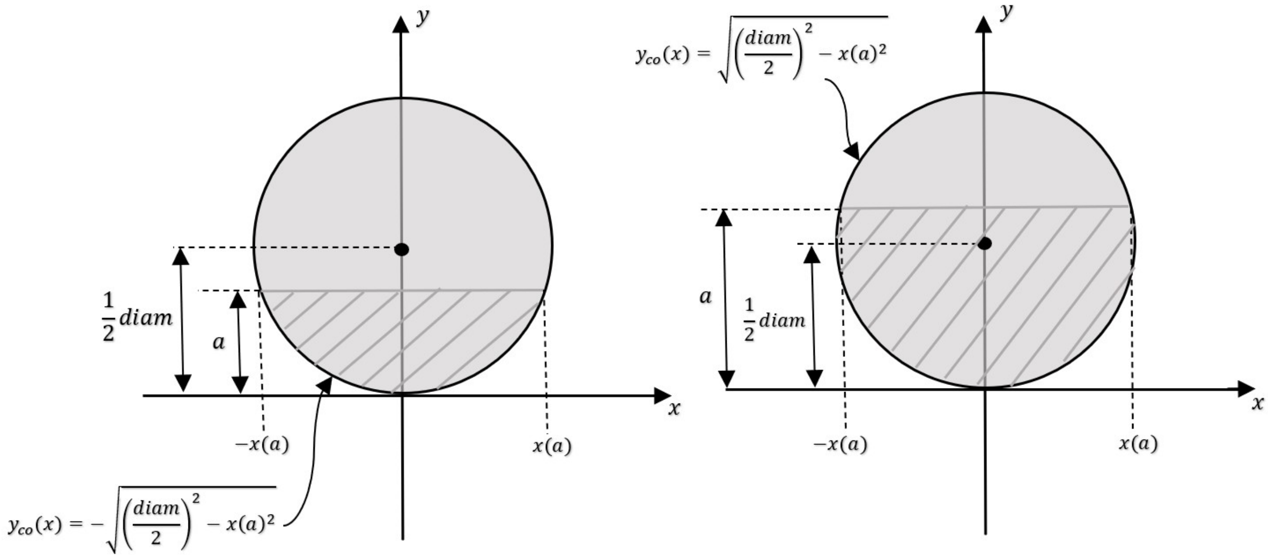


Figure 15: *System of reference of the circular column cross-section for the computation of the resistance contribution of the concrete compression zone. The panel at the left side corresponds to the case when $a < 1/2 \cdot diam$ and panel at the right side to the case when $a \geq 1/2 \cdot diam$.*

The bending resistance, on the other hand is defined by (8):

$$M_{rc} = 0.85 f'_c \cdot \int_{-x(a)}^{x(a)} (a - y_{co}(x)) \cdot (\frac{diam}{2} - \frac{a + y_{co}(x)}{2}) \cdot dx \tag{8}$$

When $a \geq \frac{diam}{2}$ and $a \leq diam$:

$$F_{rc} = -0.85 f'_c \cdot (\frac{\pi \cdot diam^2}{4}) + 0.85 f'_c \cdot \int_{-x(a)}^{x(a)} (y_{co}(x) - a) \cdot dx \tag{9}$$

Where the function $y_{co}(x)$ is defined by the equation of the upper portion of a circumference in the plane of reference of **Fig. 15 (Right)** as (10):

$$y_{co}(x) = \sqrt{(\frac{diam}{2})^2 - x^2} \tag{10}$$

The bending resistance, on the other hand is defined by (11):

$$M_{rc} = 0.85 f'_c \cdot \int_{-x(a)}^{x(a)} (a - y_{co}(x)) \cdot (\frac{diam}{2} - \frac{a + y_{co}(x)}{2}) \cdot dx \tag{11}$$

Finally, when $a > diam$, then whole section is in compression, therefore:

$$F_{rc} = -0.85 f'_c \cdot (\frac{\pi \cdot diam^2}{4}) \tag{12}$$

$$M_{rc} = 0 \tag{13}$$

3.4 Function: casoConcretoTsec

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

Syntax:

$$elemConc = casoConcretoTsec(a, fdpc, bp, be, ha, ht)$$

System of units: Any.

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 [4] code
- be : effective flange width in compression
- bp : web width of the T cross-section
- ha : flange thickness of the T cross-section
- ht : total height of the T beam cross-section

Theory:

Is considered that concrete only withstand compression forces, therefore, the zone in tension **Fig. 16** 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 (14).

$$F_c = C = a \cdot be \cdot 0.85f'_c \tag{14}$$

The bending resistance (15), on the other hand, assumes that the Plastic Centroid of the cross-section is in the same position as the Geometric Centroid (CS) **Fig. 16**.

$$M_c = F_c(CS - CC) \tag{15}$$

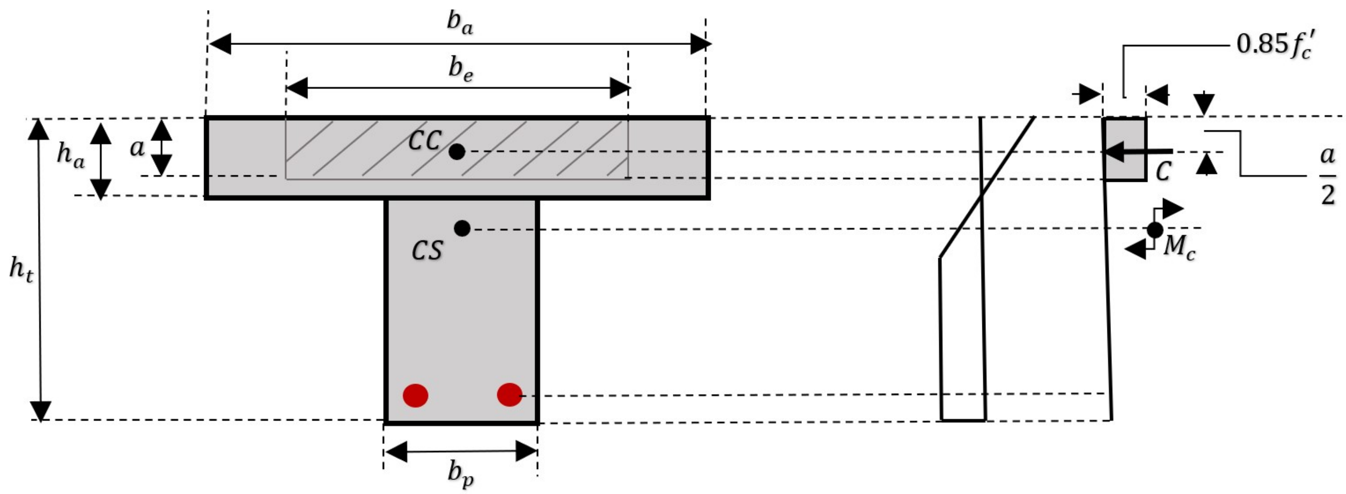


Figure 16: *Concrete compression zone of a T reinforced concrete cross-section subject to bending forces.*

3.5 Function: InertiaBeamCrackedSection

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

Syntax:

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

System of units:
SI - (Kg, cm)
US - (lb, in)

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. 17** 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 (16)

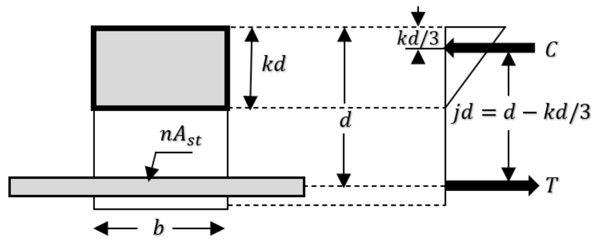


Figure 17: *Transformed section mechanism to consider beam cracked cross-sections.*

$$I_c = \frac{by^3}{12} + \frac{by^2}{4} + nA_s(d-y)^2 \tag{16}$$

3.6 Function: AmpMomSlenderColumns

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

Syntax:

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

System of units:
SI - (Kg, cm).

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(17), 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 (18)

$$M_2 = M_{2b} + F_{as} M_{2s} \tag{17}$$

$$1.5 \geq [F_{as} = \frac{1}{1-\lambda}] \geq 1.0 \tag{18}$$

When $\frac{L_c}{r} \geq \frac{35}{\sqrt{\frac{P_u}{f'_c A_g}}}$ then the amplified factor is computed as (19), where M_2 is also the greater of both moments at the ends of the column, F_{ab} is the amplification factor determined as (20), 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 (21) 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).

$$M_2 = F_{ab} M_2 \tag{19}$$

$$F_{ab} = \frac{C_m}{1 - \frac{P_u}{0.75 P_c}} \tag{20}$$

$$P_c = \frac{\pi^2 EI}{H'^2} \tag{21}$$

3.7 Function: CrackingColumnsSym

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

Syntax:

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

System of units:
SI - (Kg, cm)
US - (lb, in)

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: Modulus of Elasticity of the reinforcing steel

Theory:

For high axial load eccentricity surpassing the limit (22) a cracked cross-section inertia is computed with equation (23) based on the **Fig. 18**, with an equivalent effective transformed area as (24). 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 (26).

$$e_{lim} = \frac{2(\frac{P}{A_t} + f_r)I_g}{Ph} \tag{22}$$

$$\begin{aligned} I_t = Ixx_{ag_{sym}} = \frac{bc^3}{12} + \frac{bc^3}{4} + nt_1(b - 2cover)(h - cover - c)^2 + (n-1)t_1(b - 2cover)(c - cover)^2 + ... \\ \frac{2(n-1)t_2(\frac{h}{2} - cover - k)^3}{12} + 2(n-1)t_2(\frac{h}{2} - cover - k)(\frac{1}{2}(\frac{h}{2} - cover - k))^2 + ... \\ \frac{2nt_2(\frac{h}{2} - cover + k)^3}{12} + 2nt_2(\frac{h}{2} - cover + k)(\frac{1}{2}(\frac{h}{2} - cover + k))^2 \end{aligned} \tag{23}$$

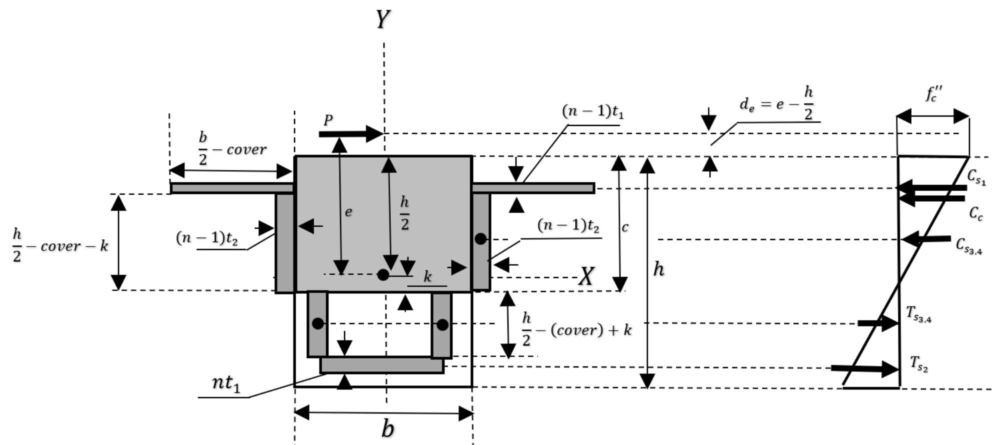


Figure 18: *Transformed cracked cross-section mechanism with symmetric reinforcement..*

$$A_t = A_{ag_{sym}} = b(h-c) + 2(b - 2cover)((n-1) + n)t_1 + 2(\frac{h}{2} - cover - k)(n-1)t_2 + 2(\frac{h}{2} - cover + k)(n-1)t_2 \tag{24}$$

On the other hand, for a non-cracked cross-section for which the axial load eccentricities are lower than e_{lim} then equation (25) applies based on **Fig. 19** with a corresponding transformed cross-section area computed as (26):

$$\begin{aligned} I_t = Ixx_{no-ag_{sym}} = \frac{bh^3}{12} + 2\frac{(b - 2(cover))((n-1)t_1)^3}{12} + ... \\ 2(n-1)t_1(b - 2(cover))(\frac{h}{2} - cover)^2 + \frac{2(n-1)t_2(h - 2(cover))^3}{12} \end{aligned} \tag{25}$$

$$A_t = A_{no-ag_{sym}} = bh + 2(b - 2cover)(n-1)t_1 + 2(h - 2cover)(n-1)t_2 \tag{26}$$

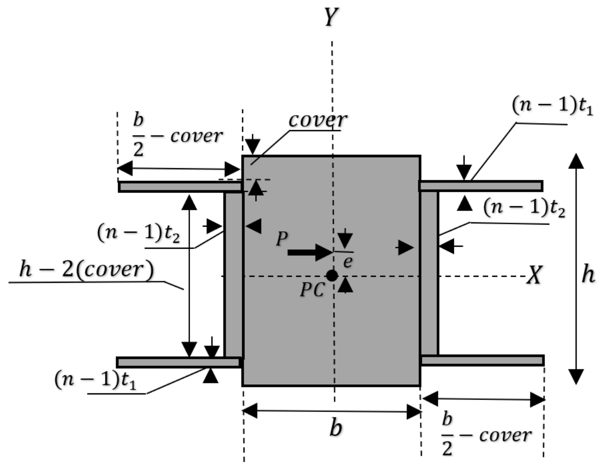


Figure 19: *Transformed non-cracked cross-section mechanism with symmetric reinforcement for small axial load eccentricities* $e \leq e_{lim}$.

3.8 Function: CrackingColumnsAsym

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

Syntax:

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

System of units:
SI - (Kg, cm)
US - (lb, in)

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: Modulus of Elasticity of the reinforcing steel

Theory:

For high axial load eccentricity surpassing the limit (22) a cracked cross-section inertia is computed with equation (27) based on the **Fig. 20**, with an equivalent effective transformed area as (28). 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 (30).

$$\begin{aligned} I_t = Ixx_{ag_{asym}} = \frac{bc^3}{12} + \frac{bc^3}{4} + nt_2(b-2cover)(h-cover-c)^2 + (n-1)t_1(b-2cover)(c-cover)^2 + ... \\ \frac{(n-1)(t_3+t_4)(\frac{h}{2}-cover-k)^3}{12} + (n-1)(t_3+t_4)(\frac{h}{2}-cover-k)(\frac{1}{2}(\frac{h}{2}-cover-k))^2 + ... \\ \frac{n(t_3+t_4)(\frac{h}{2}-cover+k)^3}{12} + n(t_3+t_4)(\frac{h}{2}-cover+k)(\frac{1}{2}(\frac{h}{2}-cover+k))^2 \end{aligned} \tag{27}$$

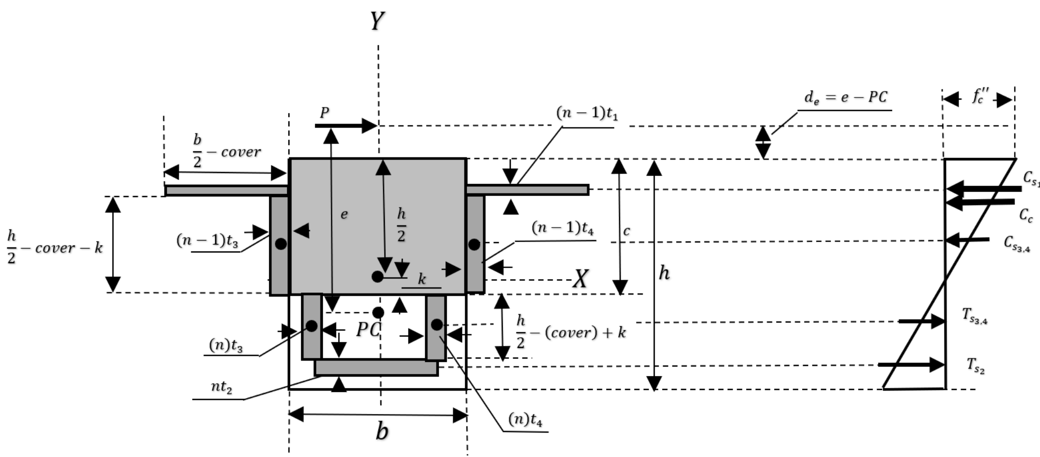


Figure 20: *Transformed cracked cross-section mechanism with asymmetric reinforcement..*

$$\begin{aligned} A_t = A_{ag_{asym}} = b(h-c) + (b-2cover)((n-1))t_1 + (b-2cover)(n)t_2 + ... \\ 2(\frac{h}{2}-cover-k)(n-1)t_3 + 2(\frac{h}{2}-cover+k)(n)t_4 \end{aligned} \tag{28}$$

On the other hand, for a non-cracked cross-section for which the axial load eccentricities are lower than e_{lim} then equation (25) applies based on **Fig. 21** with a corresponding transformed cross-section area computed as (30):

$$\begin{aligned} I_t = Ixx_{no-ag_{asym}} = \frac{bh^3}{12} + bh(\frac{h}{2}-PC)^2 + (n-1)t_1(b-2cover)(cover-CP)^2 + ... \\ (n-1)t_2(b-2cover)(h-cover-CP)^2 + \frac{(n-1)t_3(h-2(cover))^3}{12} + ... \\ ((n-1)t_3)(h-2(cover))(\frac{h}{2}-CP)^2 + \frac{(n-1)t_4(h-2cover)^3}{12} + ... \\ (n-1)t_4(h-2cover)(\frac{h}{2}-PC)^2 \end{aligned} \tag{29}$$

$$A_t = A_{no-ag_{asym}} = bh + (b - 2cover)(n-1)t_1 + (b - 2cover)(n-1)t_2 + ... \\ (h - 2cover)(n-1)t_3 + (h - 2cover)(n-1)t_4 \quad (30)$$

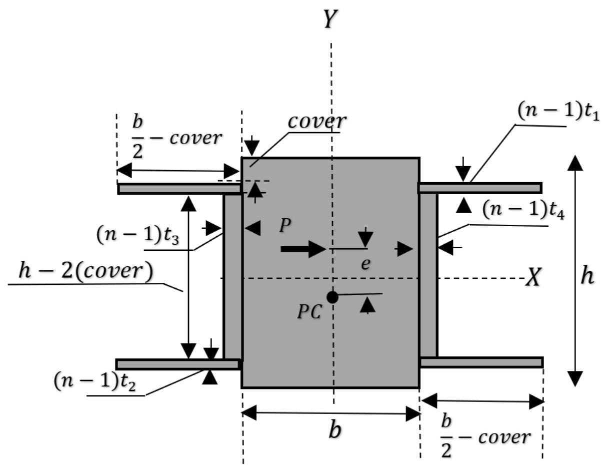


Figure 21: *Transformed non-cracked cross-section mechanism with asymmetric reinforcement for small axial load eccentricities* $e \leq e_{lim}$.

3.9 Function: RealPressuresFoot

Purpose: To compute the distribution soil's contact pressures of a rectangular isolated footing according to the transmitted column loads. Two options of types of isolated footings are available: standard isolated footings and bordering isolated footings.

Syntax:

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

System of units: Any.

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)
- $typeFooting$: type of isolated footing to be analysed. Three options are available:

    1 - Standard isolated rectangular footings

    2 - Bordering isolated rectangular footings

    3.- Corner isolated rectangular footings

- $dimCol$ is a vector containing the column's cross-section dimensions; $[b_c, h_c]$
- $plotPressure$ is an OPTIONAL argument to establish weather or not the soil's contact pressure distribution over the footing is required to be visualized:

    0 - Do not plot

    1 - Do plot

Theory:

For isolated footings undergoing eccentric biaxial loads a linear distribution of contact pressures or stresses is considered, so that the footing's surface remains plane at all times. Thus, the following expression applies for the computation of the stresses at each of the footing's corner:

$$q_{max-min} = \frac{P}{A}(1 \pm \frac{6e_x}{B} \pm \frac{6e_y}{L}) \tag{31}$$

When computing *qprom* only positive pressures are considered (compression against the bearing soil) and negative pressures are considered with a value of 0, as expressed as following:

$$q_{prom} = \frac{\sum_{i=1}^{i=4} q_i}{4} \tag{32}$$

The load's eccentricities are calculated according to the following diagrams of reference and sign convention both for standard and bordering isolated footings **Fig. 22** and **Fig. 23**:

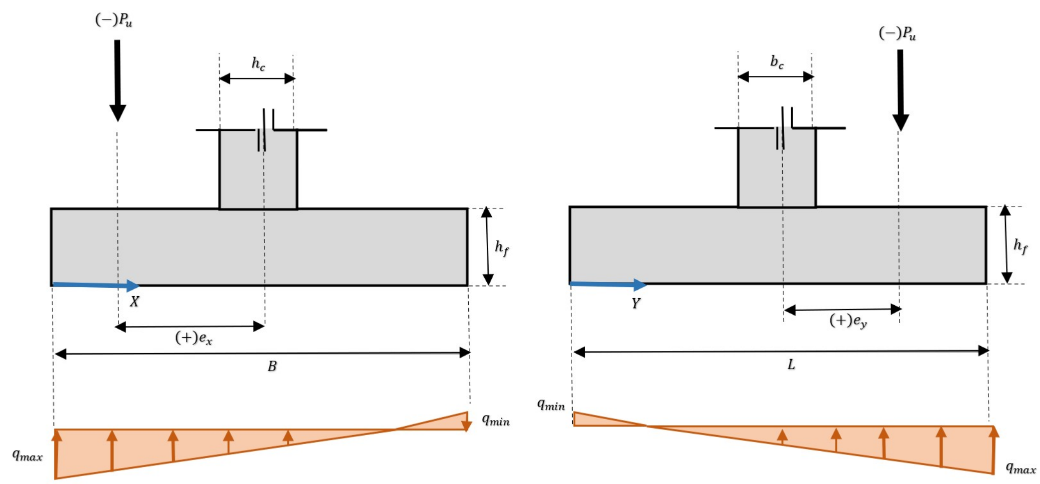


Figure 22: *System of reference for the load's eccentricities of an isolated footing.*

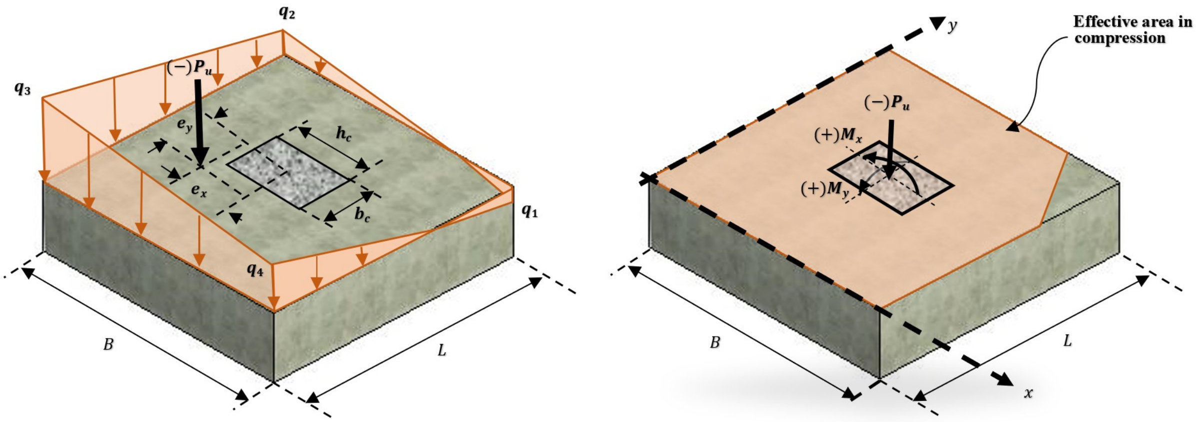


Figure 23: *(Left) 3D visualization of distribution of soil's contact pressures for isolated footings undergoing biaxial bending-compression loads, (Right) System of reference adopted for sign convention of the soil's contact pressures distribution for isolated footings.*

3.10 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:

$$[d, qprom] = shearFootings(be, le, qprom, dimCol, pu, d, fc, typeFooting)$$

System of units: SI - (Kg, cm)

Description:

Output variables:

- d effective modified height dimension based on the critical acting shear demand over the isolated footing

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)
- 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 ($\frac{Kg}{cm^2}$)
- $typeFooting$ is the type of isolated footing to be analysed. Two options are available:

  1 - Standard isolated footings (column at the center)

  2 - Bordering isolated footings (column at the border in one direction and at the center in the other)

  3.- Corner isolated footings

Theory:

In order to efficiently design the height of the footing, the shear actions and demands have to be considered (both punching shear and shear by flexure). For standard isolated footings the punching shear is analysed as (33) and shear by flexure as (34) (see **Fig. 24**). The condition $V_u < V_{CR}$ is complied at any cost. The punching shear acts over the critical perimeter surrounding the supporting column **Fig. 24 (Right)** and when beam-shear is considered is acts as in **Fig. 24 (Left)**. The maximum of the resultant effective height values for both mechanisms d is considered for design.

$$Punching - Shear = \begin{cases} V_{net} = P_u - q_{av}(b_{col} \cdot h_{col}) \\ A_{shear} = d(2(b_{col}+d) + 2(h_{col}+d)) \\ V_u = \frac{V_{net}}{A_{shear}} \\ V_{CR} = 0.85\sqrt{f'_c} \end{cases} \tag{33}$$

$$Beam - Shear = \begin{cases} V_{net} = L(q_{av})\frac{1}{2}(B - h_{col} - d) \\ V_{CR} = \frac{1}{2}0.85\sqrt{f'_c} \\ d = \frac{V_{net}}{BV_{CR}} \end{cases} \tag{34}$$

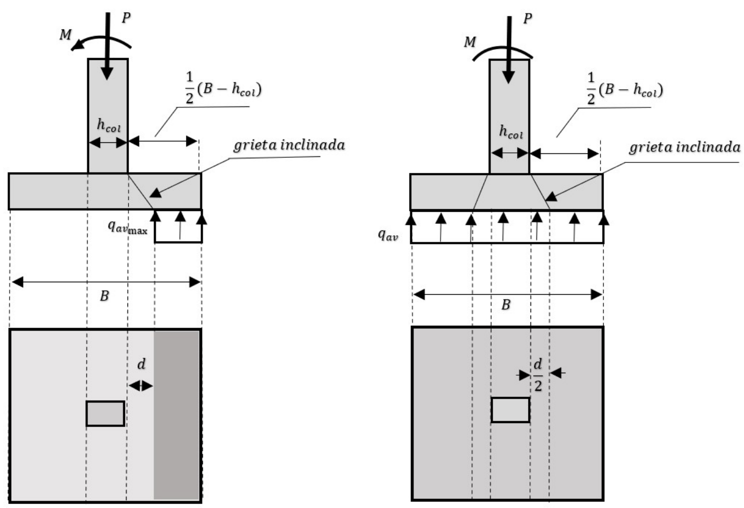


Figure 24: *(Left). Design shear mechanism as a beam, (Right). Design punching shear mechanism.*

Whereas for bordering isolated footings the following expressions (35) and (36) apply for punching shear and shear by flexure, respectively (see **Fig. 25**):

$$Punching - Shear = \begin{cases} V_{net} = P_u - q_{av}(b_{col} \cdot h_{col}) \\ A_{shear} = d((b_{col}+d) + 2(h_{col}+d)) \\ V_u = \frac{V_{net}}{A_{shear}} \\ V_{CR} = 0.85\sqrt{f'_c} \end{cases} \tag{35}$$

$$Beam - Shear = \begin{cases} V_{net} = L(q_{av})(B - h_{col} - d) \\ V_{CR} = \frac{1}{2}0.85\sqrt{f'_c} \\ d = \frac{V_{net}}{BV_{CR}} \end{cases} \tag{36}$$

Similarly as for bordering footings, the shear-by-flexure mechanism also applies for corner footings for each axis direction, however, when computing the punching-shear forces the following expression applies, in which only two shear faces or areas are considered for such shear forces to be distributed over:

$$Punching - Shear = \begin{cases} V_{net} = P_u - q_{av}(b_{col} \cdot h_{col}) \\ A_{shear} = d((b_{col}+d) + (h_{col}+d)) \\ V_u = \frac{V_{net}}{A_{shear}} \\ V_{CR} = 0.85\sqrt{f'_c} \end{cases} \tag{37}$$

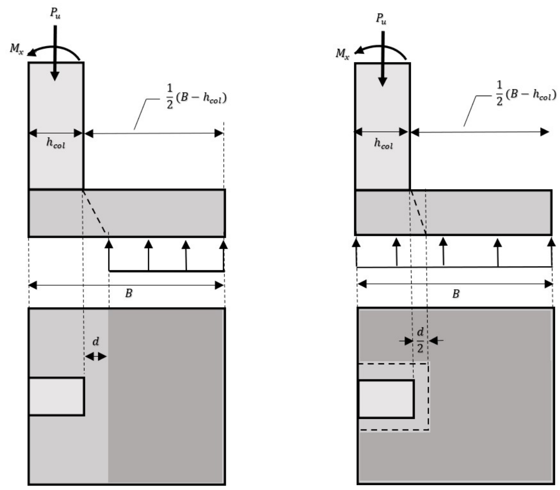


Figure 25: *(Left). Design shear mechanism as a beam for a bordering footing, (Right). Design punching shear mechanism for a bordering footing.*

3.11 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:

$[m] = Moment\_Distribution\_Footings(qmax, dimp, dimfoot)$

System of units: Any.

Description:

Output variables:

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

Input variables:

- $qmax$ is the average soil's contact pressure over the footing
- $dimp$ 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. 26** the average max shear F_{shear} is computed so that a bending moment (38) is calculated considering the effective flexure area as a cantilever beam:

$$M = F_{shear} \cdot xt \tag{38}$$

Where:

$$F_{shear} = q_{max} \cdot (dimfoot \cdot dimp) \tag{39}$$

$$x_t = \frac{1}{2} \cdot dimp \tag{40}$$

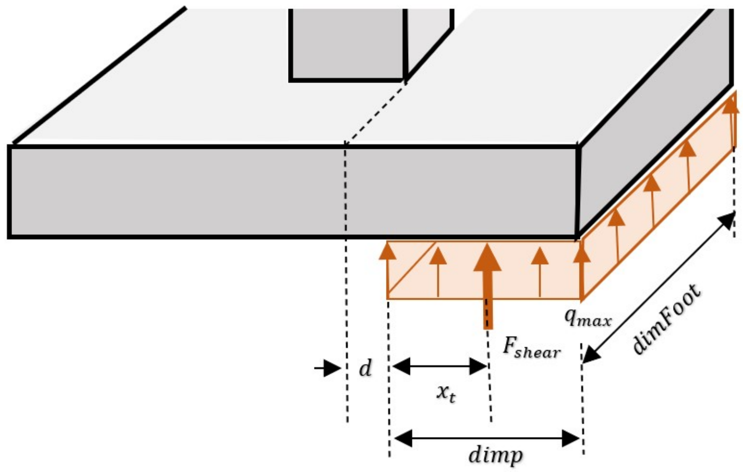


Figure 26: *Block of effective shear force that generates the design bending moment of an isolated footing.*

3.12 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. The cross-section dimensions of the intersecting column are considered so that the footing dimensions may be at least 40 cm wider than such column's cross-section dimensions.

Syntax:

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

System of units: SI - (Kg, cm)

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
- $qadm$ is the max soil bearing capacity
- $dimCol$ column cross-section dimensions $[b, h]$

- $hfooting$ is the height or width of the isolated footing
- rec is the concrete cover (cm)

3.13 Function: MomAmpColsGeomNL

Purpose: To compute the amplified moments for a rectangular cross-section column considering the $P-\Delta$ effects numerically by geometric Non-Linearity.

Syntax:

$$[Delta, Pcr, Mamp] = MomAmpColsGeomNL(fc, k, I, L, V, P, Md, b, h, plotdef)$$

System of units:
SI - (Kg, cm)
US - (lb, in)

Description:

Output variables:

- Δ Lateral displacement or additional load eccentricity that causes the second-order moments
- $Mamp$ is the amplified moment in the direction of question
- Pcr is the critical axial load of Euler for instability

Input variables:

- P Is the axial load over the column's cross-section
- V Is the shear force at the top of the column
- fc is the f'_c used
- b, h are the column cross-section dimensions
- M Is the acting bending moment
- k is the slenderness factor, according to the element's boundary conditions
- I Momentum of inertia of the cross-section in the current axis of reference
- $plotdef$ is the parameter that indicates if the plot of the deformed column is required (1-yes, otherwise-no)
- L is the total length of the column element (without considering the slenderness factor)

Theory:

The function applies geometrical Non-Linearity by computing the geometrical Non-linearity stiffness matrix of the structural element by using the CALFEM library[1] and then iteratively search for convergence considering the effect of the axial load on each iteration. If the solution does not converge in 20 iterations, the process is terminated.

On each iteration the stiffness matrix is computed as $K^e = K^e_0 + K^e_\sigma$ where K^e_0 is computed as $G^T \bar{K}^e G$ with K^e_0 and G defined as (41), (42), respectively, and K^e_σ as (43). The iterative process starts with a very low axial force value Q_x (corresponding to the linear static analysis).

[1]The most recent CALFEM version is available to download as open source at its GitHub repository https://github.com/CALFEM/calfem-matlab

$$K^e = \begin{bmatrix} \frac{EA}{L} & 0 & 0 & -\frac{EA}{L} & 0 & 0 \\ 0 & \frac{12EI}{L^3} & \frac{6EI}{L^2} & 0 & -\frac{12EI}{L^3} & \frac{6EI}{L^2} \\ 0 & \frac{6EI}{L^2} & \frac{4EI}{L} & 0 & -\frac{6EI}{L^2} & \frac{2EI}{L} \\ -\frac{EA}{L} & 0 & 0 & \frac{EA}{L} & 0 & 0 \\ 0 & -\frac{12EI}{L^3} & -\frac{6EI}{L^2} & 0 & \frac{12EI}{L^3} & -\frac{6EI}{L^2} \\ 0 & \frac{6EI}{L^2} & \frac{2EI}{L} & 0 & -\frac{6EI}{L^2} & \frac{4EI}{L} \end{bmatrix} \tag{41}$$

$$G = \begin{bmatrix} \frac{x_2-x_1}{L} & \frac{y_2-y_1}{L} & 0 & 0 & 0 & 0 \\ \frac{y_1-y_2}{L} & \frac{x_2-x_1}{L} & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{x_2-x_1}{L} & \frac{y_2-y_1}{L} & 0 \\ 0 & 0 & 0 & \frac{y_1-y_2}{L} & \frac{x_2-x_1}{L} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{42}$$

$$K^e = Q_x \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{6}{5L} & \frac{1}{10} & 0 & -\frac{6}{5L} & \frac{1}{10} \\ 0 & \frac{1}{10} & \frac{2L}{15} & 0 & -\frac{1}{10} & -\frac{L}{30} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{6}{5L} & -\frac{1}{10} & 0 & \frac{6}{5L} & -\frac{1}{10} \\ 0 & \frac{1}{10} & -\frac{L}{30} & 0 & -\frac{1}{10} & \frac{2L}{15} \end{bmatrix} \tag{43}$$

3.14 Function: shearDesignBeams

Purpose: To design the separation of the transversal reinforcement along the whole length of a rectangular beam element according to the mechanic shear forces distribution).

Syntax:

$$[s1, s2, s3, d1, d2] = shearDesignBeams(span, b, h, rec, rho, fc, fy, shear\_beam)$$

System of units: SI - (Kg, cm)

Description:

Output variables:

- $s1$ is the separation of the transversal reinforcement at the left part/section of the beam element.
- $s2$ is the separation of the transversal reinforcement at the middle part/section of the beam element.
- $s3$ is the separation of the transversal reinforcement at the right part/section of the beam element.
- $d1$ is the length along which the transversal reinforcement is separated by s1 (cm). Left part length.
- $d2$ is the length along which the transversal reinforcement is separated by s2 (cm). Right part length.

Input variables:

- $span$ is the length of the beam element
- b, h are the cross-section dimensions: width and height, respectively
- rec is the concrete cover
- rho is the average longitudinal reinforcing percentage area
- $shear\_beam$ is the array containing the shear forces distribution from left to right
- fc, fy is compressive strength of the concrete and the yield stress of the reinforcing steel.

Theory:

Given the standard distribution of shear stresses along beam elements, as shown in **Fig. 27** the transversal reinforcement separation may vary along such length to optimize the materials. Such separation is calculated as (**??**):

$$s_i = \frac{2 \cdot F_R \cdot a_b \cdot f_y \cdot (h - rec))}{V_u - V_{CR}} \tag{44}$$

where:

$$V_{CR} = F_R \cdot (b \cdot (h - rec)) \cdot (0.2 + 20\rho) \cdot \sqrt{f'_c}$$
$$F_R = 0.75$$

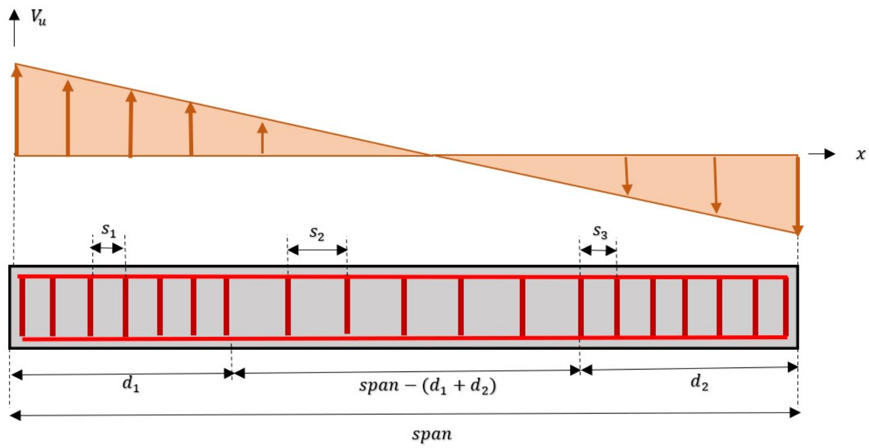


Figure 27: *Transversal reinforcement design of a beam element according to the shear forces distribution along its length.*

4 ISR Optimization functions

4.1 ISR analysis for rectangular 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:

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

System of units: Any.

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)
- $fdpc = 0.85 f'_c$: according to [4]

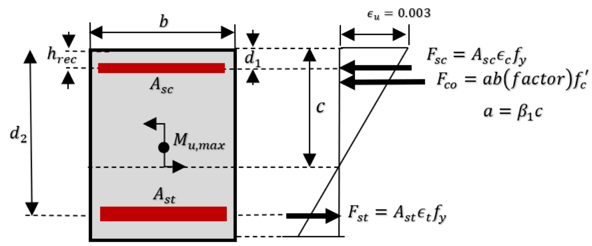
Theory:

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 (45), where $Fs_i = As_i E_y \epsilon_i$ for reinforcement steel.

$$M_R = (\sum Fs_i + \beta_1 ab0.85 f'_c)(\frac{h}{2} - d_i) \quad (45)$$

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 $\sum F = 0$.



Syntax:

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

System of units: Any.

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 [4] 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)
- $fdpc = 0.85f_c'$: according to [4]
- β_1 is determined as following (195) in units Kg, cm or as (196) in units psi

$$0.65 \leq (\beta_1 = 1.05 - \frac{f_c'}{1400}) \leq 0.85 \tag{46}$$

$$0.65 \leq (\beta_1 = 0.85 - 0.05\frac{f_c' - 4000}{1000}) \leq 0.85 \tag{47}$$

- ea is the approximation root error

Theory:

The nominal resistant bending moment is determined as (48) for a given neutral axis depth value c with the function **eleMecanicos2tBeams** (p.47).

$$M_n = \sum F_s + (\beta_1 c) b 0.85 f_c' (\frac{h-a}{2}) \tag{48}$$

The neutral axis depth is restricted by a ductility strain requirement established by code **ACI 318-19** as (49):

$$c \leq \frac{d}{\frac{0.005}{0.003}+1} \tag{49}$$

Now, regarding the Bisection method, it takes advantage of the fact that a function is of different sign at both proximities of a root. In other words if $f(x)$ is real and continuous in an interval x_i to x_u and $f(x_i$ and $f(x_u)$ have opposite signs then there is a root between x_i and x_u (which is the reason why the initial neutral axis depth values must involve the whole cross-section height - which is the possible range of the neutral axis to lie in, that is $[0,h]$). The location of such sign change (root) is identified more precisely by dividing the original interval x_i, x_u into subintervals in an iterative fashion. The bisection method deploys (as its names indicates it) a binary division of intervals until a convergence to the root is reached (the intervals are always divided in half). The pseudo-code is presented next (**Algorithm 5.3**):

Algoritmo 4.1: The bisection method to find roots of a one-variable continuous function

BEGIN

1.- Choose the initial values x_i and x_u to start the iteration such that $f(x_i)f(x_u) < 0$.

2.- Estimate the root by dividing the previous interval in two as:

$x_r = \frac{x_i + x_u}{2}$

3.- Make the following evaluations to determine in which subinterval the root lies:

a) If $f(x_i)f(x_r) > 0$ the root lies in the upper subinterval. Therefore, $x_i = x_r$ for the next evaluation

b) If $f(x_i)f(x_r) < 0$ the root lies in the lower subinterval. Therefore, $x_u = x_r$.

4.- Estimate the approximation error ϵ to stop the process or continue:

If $\epsilon \leq toler$ stop the process, otherwise return to step 2.

END

A good criterion to decide when to terminate the process is to establish a tolerance values under which the error approximation must lie. The computation of such error can be determined as a percent relative error ϵ_a (152):

$$epsilon_a = |\frac{x_r^{new} - x_r^{old}}{x_r^{new}}|100\% \tag{50}$$

In other words, the bisection method traverse through the following function of the cross-section axial force resistance and the neutral axis depicted in **Fig. 28**:

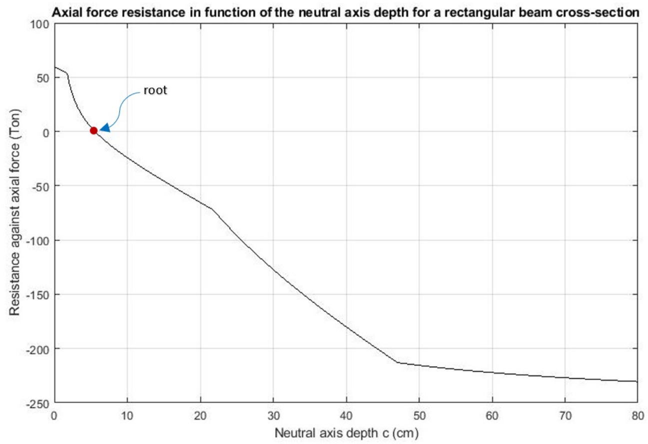


Figure 28: *Axial force resistance of a rectangular beam cross-section in function of the neutral axis depth.*

4.1.3 Function: Efrec2tbeams

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

Syntax:

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

System of units: Any.

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 [4] specifies it as 0.85
- β_1 is determined as following (195) in units Kg, cm

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

- 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.48), the resistance reduction factor ϕ (98) is then applied as (99) according to the condition of *tension controlled* or *compression controlled* for the cross-section:

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

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

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:

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

System of units:
SI - (Kg, cm)
US - lb, in

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 [4] 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 (54) 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.51).

$$Eff = \frac{M_{umax}}{M_R} \tag{54}$$

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 (55), (56) and (57) respectively (in units Kg, cm), controlled by the *duct* input argument, or as specified in the ACI 318 code (58) in units *psi*:

Low ductility:

$$\frac{0.7bd}{b-2b_{rec}}\frac{\sqrt{0.85f_c'}}{f_y} \leq t \leq \frac{0.9(0.85f_c')}{(b-2b_{rec})f_y}\frac{bd(6000\beta_1)}{(f_y+6000)} \tag{55}$$

Medium ductility:

$$\frac{0.7bd}{b-2b_{rec}}\frac{\sqrt{0.85f_c'}}{f_y} \leq t \leq \frac{0.75(0.85f_c')}{(b-2b_{rec})f_y}\frac{bd(6000\beta_1)}{(f_y+6000)} \tag{56}$$

High ductility:

$$\frac{0.7bd}{b-2b_{rec}}\frac{\sqrt{0.85f_c'}}{f_y} \leq t \leq \frac{0.025bd}{b-2b_{rec}} \tag{57}$$

ACI code:

$$3\frac{\sqrt{f_c'}}{f_y}b\cdot d/(b_p-2\cdot cover) \leq t \leq \frac{200}{f_y}b\cdot d/(b_p-2\cdot cover) \tag{58}$$

4.2 ISR analysis for T beams

4.2.1 Function: SGD1tTBeamsISR

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

Syntax:

$[c\_best, bestMr, bestEf, best\_area, tbest] = SGD1tTBeamsISR(bp, ht, ba, ...$
$ha, Lb, duct, cover, fc, load\_conditions, factor\_fc, E, graphConvergencePlot)$

System of units:
SI - (Kg, cm)
US - lb, in

Description:

Output variables:

- $c\_best, bestMr, bestEf$: the neutral axis depth for the optimal design, the resistant bending moment for the optimal design and the final structural efficiency for the optimal design
- $best_area, tbest$: the optimal reinforcement area, the optimal t width of the ISR

Input variables:

- $load\_conditions$: array containing the pure flexure loads, in format: $[nload, M_u]$ - size $nload\ x\ 2$
- $factor\_fc$: is determined by the applicable design code. The ACI 318-19 specifies it as 0.85
- $duct$: is the ductility demand parameter, with possible values of 1,2 or 3, for low ductility, medium ductility or high ductility, respectively
- ba : is the effective flange width of the T-beam cross-section
- ht : is total height of the T-beam cross-section
- bp : is the web width of the T-beam cross-section
- ha : is the flange thickness of the T-beam cross-section
- Lb : is the length of the beam element
- E : is the Elasticity Modulus of reinforcing steel (Kg/cm^2)
- $cover$: is the concrete cover for the reinforcement
- $graphConvergencePlot$: is the parameter that indicates weather or not it is required to plot the optimization convergence with the Steepest Gradient Descent method

Theory:

The function deploys the Steepest Gradient Descent (SGD) method (see **Algorithm 2.1**) to determine the optimal ISR's width t (in tension) for the T beam cross-section, which can then, be transformed to reinforcement area (see **Fig. 29**). The termination criteria is that the structural efficiency is between a range $[Ef_{max}, Ef_{min}]$, where Ef_{max} is usually set to 1.0. Such efficiency is computed by the function $Efrec2tBeamsT$ (56) which is used as the objective function for the optimization process.

There is a parameter called *duct* with which the ductility demand can be set through the establishment of max and min reinforcement area limits (or ISR's t width limits). Three level of demands are possible to set in this function: low ductility (59), medium ductility (60) and high ductility (61) as specified in the NTC-17 Mexican code in units Kg, cm or as specified in the ACI code in units lb, in (62). See **Fig. 29**

$$0.7\frac{\sqrt{0.85f'_c}\cdot(b_p\cdot d)}{f_y\cdot(b_p-2\cdot cover)} \leq t \leq 0.9\frac{0.85f'_c\cdot(b_p\cdot d)}{f_y\cdot(b_p-2\cdot cover)}\cdot\frac{\beta_1\cdot 6000}{f_y+6000} \tag{59}$$

$$0.7\frac{\sqrt{0.85f'_c}\cdot(b_p\cdot d)}{f_y\cdot(b_p-2\cdot cover)} \leq t \leq 0.75\frac{0.85f'_c\cdot(b_p\cdot d)}{f_y\cdot(b_p-2\cdot cover)}\cdot\frac{\beta_1\cdot 6000}{f_y+6000} \tag{60}$$

$$0.7\frac{\sqrt{0.85f'_c}\cdot(b_p\cdot d)}{f_y\cdot(b_p-2\cdot cover)} \leq t \leq \frac{0.025\cdot(b_p\cdot d)}{b_p-2\cdot cover} \tag{61}$$

ACI code:

$$3\frac{\sqrt{f'_c}}{f_y}b\cdot d/(b_p-2\cdot cover) \leq t \leq \frac{200}{f_y}b\cdot d/(b_p-2\cdot cover) \tag{62}$$

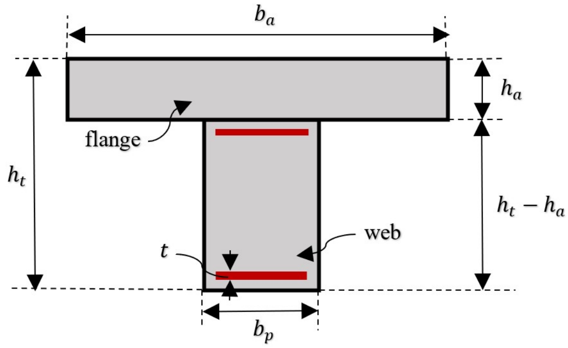


Figure 29: *T beam cross-section geometry. Only one ISR's width t variable is considered in tension, whereas the steel quantity in compression remains constant.*

4.2.2 Function: Efrec2tBeamsT

Purpose: computes the structural efficiency of A T-beam cross-section according to the applied load conditions (pure flexure).

Syntax:

$[maxef, Mrt, c] = Efrec2tBeamsT(load\_conditions, fc, factor\_fc, E, bp, ht, ba, ...$
$ha, Lb, ast, asc, cover, beta1)$

System of units: Any.

Description:

Output variables:

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

Input variables:

- $load\_conditions$: array containing the pure flexure loads, in format: $[nload, M_u]$ - size $nload$ x 2
- $factor\_fc$: is determined by the applicable design code. The ACI 318-19 specifies it as 0.85
- $beta1$: is determined as prescribed by the ACI 318 code (according to the f'_c value)
- ba : is the effective flange width of the T-beam cross-section
- ht : is total height of the T-beam cross-section
- bp : is the web width of the T-beam cross-section
- ha : is the flange thickness of the T-beam cross-section
- Lb : is the length of the beam element
- E : is the Elasticity Modulus of reinforcing steel (Kg/cm^2)
- $cover$: is the concrete cover for the reinforcement
- ast, asc : are the steel reinforcement area quantity in tension and the reinforcement area quantity in compression, respectively

Theory:

The structural efficiency is computed as (63), where Mu_{max} is the max bending load imposed over the cross-section and M_R is the resistant bending moment of the reinforced T beam cross-section computed as (64) (in which Mn is computed by the function $bisectionMr2tBeamsT$ (p. 58)):

$$Eff = \frac{Mu_{max}}{M_R} \tag{63}$$

$$M_R = \phi Mn \tag{64}$$

The resistance reduction factor ϕ is a function of the steel strain in tension ϵ_t (102) which is computed as (103):

$$\phi = \begin{cases} 0.65 + (\epsilon_t - 0.002)\frac{250}{3} & 0.004 \leq \epsilon_t \leq 0.005 \\ 0.9, & \epsilon_t > 0.005 \end{cases} \tag{65}$$

$$\epsilon_t = \frac{0.003 \cdot (d - c)}{c} \tag{66}$$

4.2.3 Function: bisectionMr2tBeamsT

Purpose: to determine the neutral axis depth and the resistant bending moment of a reinforced T-beam cross-section taking on account both the steel in compression and steel in tension with the aid of the root bisection method so that the forces equilibrium condition sum F=0 is complied.

Syntax:

$[raiz] = bisectionMr2tBeamsT(cUno, cDos, fr, E, t1, t2, bp, ht, ba, ha, Lb, cover, ...$
$fdpc, beta1, ea)$

System of units: Any.

Description:

Output variables:

- $raiz$: vector that containg the output $[c, \sum F_i - > 0, MR]$ where c is the final neutral axis depth value, $\sum F_i \leftarrow 0$ is the reached forces equilibrium (which for pure flexure it should be very close to 0) and MR is the corresponding resistant bending moment.

Input variables:

- $cUno, cDos$ are the initial values for the neutral axis depth as the bisection method requires them to begin the iterations. Such values are recommended to be $cUno \rightarrow 0$ and $cDos \rightarrow h_t$
- fr : is the applied axial force over the beam cross-section (which for pure flexure is considered as 0)
- $beta1$: is determined as prescribed by the ACI 318 code (according to the f'_c value)
- ba : is the effective flange width of the T-beam cross-section
- ht : is total height of the T-beam cross-section
- bp : is the web width of the T-beam cross-section
- ha : is the flange thickness of the T-beam cross-section
- Lb : is the length of the beam element
- E : is the Elasticity Modulus of reinforcing steel
- $cover$: is the concrete cover for the reinforcement
- $t1, t2$: are the ISR's width in compression and tension, respectively
- ea : is the termination error for the root method

Theory:

The Bisection method takes advantage of the fact that a function is of different sign at both proximities of a root. In other words if $f(x)$ is real and continuous in an interval x_i to x_u and $f(x_i$ and $f(x_u)$ have opposite signs then there is a root between x_i and x_u (which is the reason why the initial neutral axis depth values must involve the whole cross-section height - which is the possible range of the neutral axis to lie in, that is $[0, ht]$). The location of such sign change (root) is identified more precisely by dividing the original interval x_i, x_u into subintervals in an iterative fashion. The bisection method deploys (as its names indicates it) a binary division of intervals until a convergence to the root is reached (the intervals are always divided in half). The pseudo-code is presented next (**Algorithm 5.3**):

Algoritmo 4.2: The bisection method to find roots of a one-variable continuous function

BEGIN

1.- Choose the initial values x_i and x_u to start the iteration such that $f(x_i)f(x_u) < 0$.

2.- Estimate the root by dividing the previous interval in two as:

$x_r = \frac{x_i + x_u}{2}$

3.- Make the following evaluations to determine in which subinterval the root lies:

a) If $f(x_i)f(x_r) > 0$ the root lies in the upper subinterval. Therefore, $x_i = x_r$ for the next evaluation

b) If $f(x_i)f(x_r) < 0$ the root lies in the lower subinterval. Therefore, $x_u = x_r$.

4.- Estimate the approximation error ϵ to stop the process or continue:

If $\epsilon \leq toler$ stop the process, otherwise return to step 2.

END

A good criterion to decide when to terminate the process is to establish a tolerance values under which the error approximation must lie. The computation of such error can be determined as a percent relative error ϵ_a (152):

$$\epsilon_a = |\frac{x_r^{new} - x_r^{old}}{x_r^{new}}|100\% \tag{67}$$

In other words, the bisection method traverse through the following function of the cross-section axial force resistance and the neutral axis depicted in **Fig. 44**:

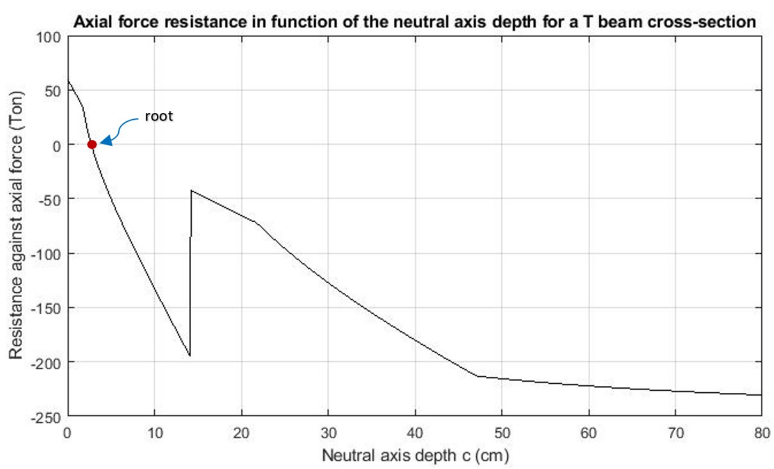


Figure 30: *Axial force resistance of a T-beam cross-section in function of the neutral axis depth.*

4.2.4 Function: eleMecanicos2tBeamsT

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

Syntax:

$$eleMec = eleMecanicos2tBeamsT(c, a, fdpc, bp, ht, ba, ha, cover, Es, t1, t2)$$

System of units: Any.

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
- $cover$ is the concrete cover for the reinforcing steel (cm)
- $fdpc = 0.85f'_c$: according to [4]
- c : is the nuetral axis depth value
- a : is the reduced effective neutral axis depth value $a = beta1 \cdot c$
- ba : is the effective flange width of the T-beam cross-section
- ht : is total height of the T-beam cross-section
- bp : is the web width of the T-beam cross-section
- ha : is the flange thickness of the T-beam cross-section
- $span$: is the length of the beam element

Theory:

When $a \leq ha$, it is considered that the beam cross-section is working as a T-beam cross-section, that is, with an improvement in the bending resistance because the flange is working together with the web. For this case, the effective width of the flange be is computed as (105):

$$be = min \begin{cases} bp + 16 \cdot ha \\ bp + ba \\ bp + \frac{span}{4} \end{cases} \tag{68}$$

and the centroid of the beam is computed as (106) see **Fig. 45** for more reference.

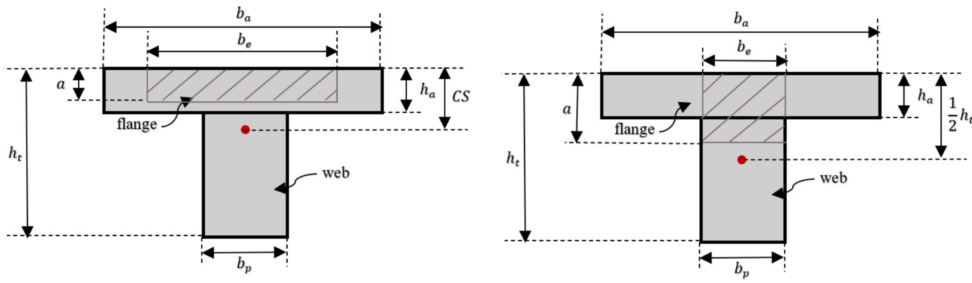


Figure 31: *T-beam cross-section working by pure flexure. The left panel depicts the effective flange width when the cross-section is working as a T-beam cross-section for $a \leq ha$. The right panel shows the effective cross-section working as a rectangular one when $a > ha$.*

$$CS = \frac{ha \cdot be \cdot (a - ha + \frac{ha}{2}) + (a - ha) \cdot bp \cdot (\frac{a-ha}{2})}{ba \cdot ha + (a - ha) \cdot bp} \tag{69}$$

On the other hand, when $a > ha$ it is considered that the beam cross-section is working as a rectangular cross-section with an effective width equal to the web's width $be = bp$ and the centroid located at a depth $CS = \frac{ht}{2}$.

Once it has been determined how the beam is working, then the resistant moment is calculated as (107), where $Fs_i = As_i E_y \epsilon_i$ for reinforcement steel.

$$M_R = (\sum Fs_i + \beta_1 ab0.85f_c')(CS - d_i) \tag{70}$$

4.3 ISR analysis for rectangular columns

4.3.1 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:

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

System of units: Any.

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 (195) in units Kg, cm or as (196) in units psi

$$0.65 \leq (\beta_1 = 1.05 - \frac{f'_c}{1400}) \leq 0.85 \tag{71}$$

$$0.65 \leq (\beta_1 = 0.85 - 0.05\frac{f'_c - 4000}{1000}) \leq 0.85 \tag{72}$$

- 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 [5].

4.3.2 Function: bisectionMrAnalytISR

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. A mathematical approach is used for the ISR's analysis and the bisection root method is used to find such resistance iteratively.

Syntax:

$$[raiz] = bisectionMrAnalytISR(cUno, cDos, fr, E, t, h, b, rec, fdpc, beta1, ea)$$

System of units: Any.

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
- t are the ISR width (considering it constant)
- 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 (195) in units Kg, cm or as (196) in units psi

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

$$0.65 \leq (\beta_1 = 0.85 - 0.05\frac{f'_c - 4000}{1000}) \leq 0.85 \tag{74}$$

- 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 (according to cross-section axis of reference). For more info about this method see [5]. The function $eleMecISRAnalyt$ (p. 66) is used, from which the root is found.

4.3.3 Function: eleMecanicos4t

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

Syntax:

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

System of units: Any.

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 Modulus of Elasticity of the reinforcing steel
- $fdpc$ 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 (195) in units Kg, cm or as (196) in units psi

$$0.65 \leq (\beta_1 = 1.05 - \frac{f'_c}{1400}) \leq 0.85 \tag{75}$$

$$0.65 \leq (\beta_1 = 0.85 - \frac{f'_c - 4000}{1000}) \leq 0.85 \tag{76}$$

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. 32**, so that the resistant moment is calculated as (93), where $Fs_i = As_i E_y \epsilon_i$ for reinforcement steel.

$$M_R = (\sum Fs_i + \beta_1 ab0.85 f'_c)(\frac{h}{2} - d_i) \tag{77}$$

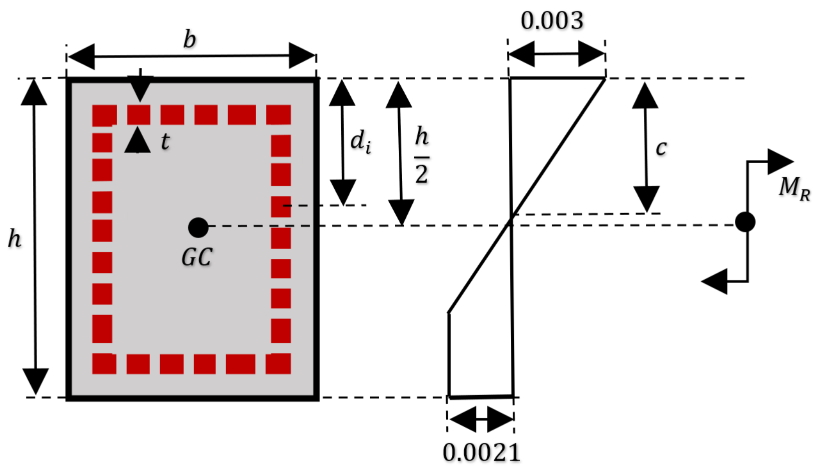


Figure 32: *Flexure-compression mechanism of a column cross-section with a discrete ISR.*

4.3.4 Function: eleMecISRAnalyt

Purpose: To determine the axial load and bending resistance of a reinforced column cross section with the ISR-1t from a given neutral axis depth value. A mathematical approach is deployed for such ISR analysis.

Syntax:

$eleMec = eleMecISRAnalyt(c, a, fdpc, h, b, rec, E, t, dUno, dDos)$

System of units: Any.

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)
- t are the ISR width of the 1t-ISR (width constant)
- rec is the concrete cover
- E Modulus of Elasticity of the reinforcing steel (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

Theory:

The function considers the location of the Plastic Center (PC) the same as the Geometric Center (GC) (which is at a depth $\frac{h}{2}$). The ISR of **Fig. 33** is used, from which the resistant axial force of the reinforcing steel is computed as (78) and the resistant bending moment as (79):

$$F_R = \sum_{i=1}^{i=nblocks} F_i \tag{78}$$

$$M_R = \sum_{i=1}^{i=nblocks} F_i \cdot (\frac{1}{2}h - d_i) \tag{79}$$

where $nblocks$ is the number of geometrical stresses blocks that depends on the neutral axis depth value, as shown in **Fig. 34**, from which as many as eight cases can be defined:

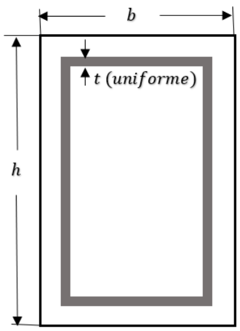


Figure 33: *ISR adopted for analysis of resistance with a mathematical approach .*

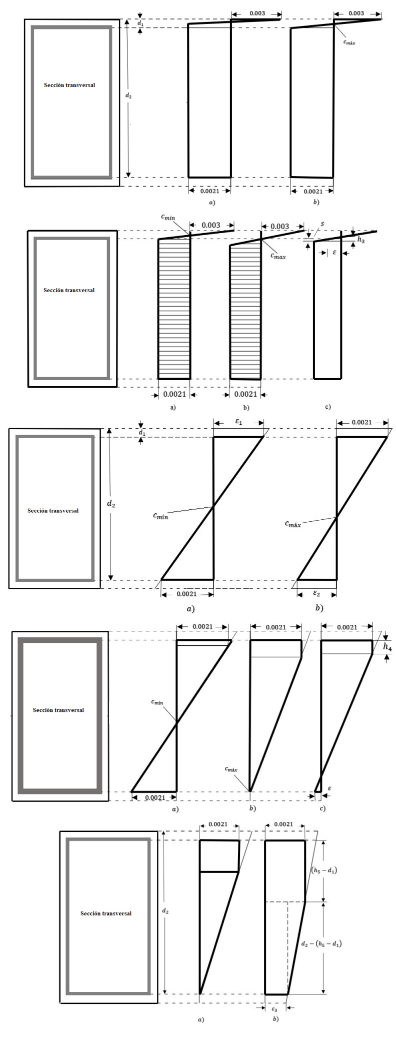


Figure 34: *Geometrical blocks of stresses that may take place for different neutral axis depth values.*

4.3.5 Function: widthEfficiencyCols

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

Syntax:

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

System of units: Any.

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 (195) in units Kg, cm

$$0.65 \leq (\beta_1 = 1.05 - \frac{f'_c}{1400}) \leq 0.85 \tag{80}$$

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

Theory:

The max compression resistance of the column cross-section is determined as (193) 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 (194).

$$P_{oc} = 0.85f'_c(A_c - A_s) + f_y(A_s) \tag{81}$$

$$P_{ot} = f_y(A_s) \tag{82}$$

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. 72**.

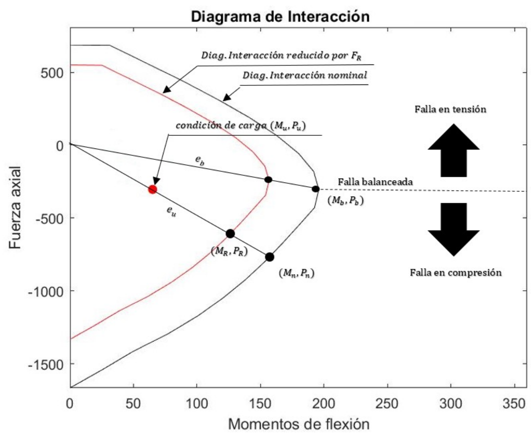


Figure 35: *Interaction diagram of reference.*

4.3.6 Function: isrColumns

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

Syntax:

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

System of units:
SI - (Kg, cm)
US - (lb, in)

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 [4]
- β_1 is determined as following (195) in units Kg, cm

$$0.65 \leq (\beta_1 = 1.05 - \frac{f'_c}{1400}) \leq 0.85 \tag{83}$$

- $load_conditions$ 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 (84) and (87) respectively, controlled by the *duct* input argument.

Low and Medium ductility:

$$\frac{0.01bh}{2(b-2b_{rec})+2(h-2h_{rec})} \leq t \leq \frac{0.06bh}{2(b-2b_{rec})+2(h-2h_{rec})} \tag{84}$$

High ductility:

$$\frac{0.01bh}{2(b-2b_{rec})+2(h-2h_{rec})} \leq t \leq \frac{0.04bh}{2(b-2b_{rec})+2(h-2h_{rec})} \tag{85}$$

4.4 ISR analysis for circular columns

4.4.1 Function: isrCircCols

Purpose: To determine an optimal ISR through the Steepest Gradient Descent method for a column of circular cross-section given certain load conditions.

Syntax:

$[cost\_elem\_col, Ac\_sec\_elem, Ef\_sec\_col, Mr\_col, t\_value, c] = isrCircCols...$
$(pu_cols, height, diam, rec, fy, fc, load_conditions, duct, optimaConvPlot, plotISRResults)$

System of units:
SI - (Kg, cm)
US - (lb, in)

Description:

Output variables:

- *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* is the resisting moment of the optimal reinforced concrete column
- *t_value* is the optimal ISR width t
- c neutral axis depth of optimal design corresponding to the critical load condition

Input variables:

- *diam* is the cross-section diameter
- is the concrete compressive strength $f'c$
- $load_conditions$ are the load conditions applied to the cross-section: size = $[nloads, 3]$ as: $[load, Pu, Mu]$
- *duct* parameter that indicates the ductility demand: (1),(2),(3) for low, medium and high ductility
- *rec* is the concrete cover
- *optimaConvPlot* is the parameter that indicates if the optima convergence plot is required or not. Option are: (1) the plot is required, (2) the plot is not required
- *plotISRResults* is the parameters that indicates if the ISR interaction diagram is required or not. Options are: (1) it is required, (2) it is not required

Theory:

The structural efficiency is calculated with thefunction **diagCircColsISR** (p.74).

The max and min reinforcing steel area is set according to the required ductility; three options are available: low ductility, medium ductility or high ductility (84) and (87) respectively, controlled by the *duct* input argument.

Low and Medium ductility:

$$\frac{0.01 \cdot (\frac{\pi diam^2}{4})}{\pi \cdot diam} \leq t \leq \frac{0.06 \cdot (\frac{\pi diam^2}{4})}{\pi \cdot diam} \tag{86}$$

High ductility:

$$\frac{0.01 \cdot (\frac{\pi diam^2}{4})}{\pi \cdot diam} \leq t \leq \frac{0.04 \cdot (\frac{\pi diam^2}{4})}{\pi \cdot diam)} \tag{87}$$

4.4.2 Function: diagCircColsISR

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

Syntax:

$[Eft, diagramaInteraccion, tablaEficiencias, c] = diagCircColsISR(t, diam, rec, fy, npuntos, ...$
$conditions, fdpc, E, beta)$

System of units: Any.

Description:

Output variables:

- Eft Structural efficiency
- $diagramaInteraccion$ interaction diagram coordinates
- $tablaEficiencias$ is the resume table of results consisting of $nload\_conditions$ rows and eight columns as $[P_u, M_u, P_R, M_R, Eff]$
- c is the neutral axis depth with respect to any cross-section axis according to the most critical load condition

Input variables:

- $diam$ is the cross-section diameter
- $npdiag$ is the number of points to be analysed for the interaction diagram
- t is the ISR width of a 1t-ISR for column cross-sections
- rec is the concrete cover
- E Modulus of elasticity of the reinforcing steel
- $fdpc$ 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 (195) in units Kg, cm

$$0.65 \leq (\beta_1 = 1.05 - \frac{f'_c}{1400}) \leq 0.85 \tag{88}$$

Theory:

The max compression resistance of the column cross-section is determined as (193) 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 (194).

$$P_{oc} = 0.85f'_c(A_c - A_s) + f_y(A_s) \tag{89}$$

$$P_{ot} = f_y(A_s) \tag{90}$$

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. 72**.

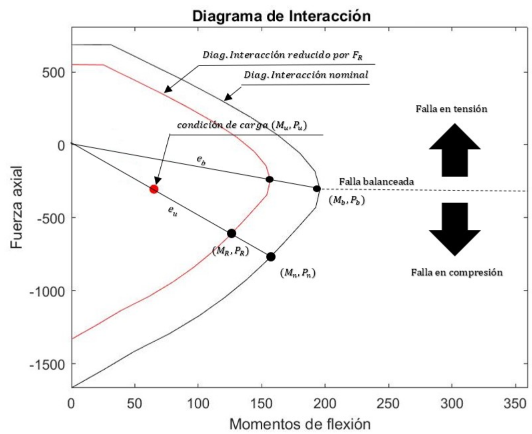


Figure 36: *Interaction diagram of reference.*

4.4.3 Function: bisectionMrCircISR1t

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

Syntax:

$[raiz] = bisectionMrCircISR1t(cUno, cDos, fr, E, t, diam, rec, fdpc, beta1, ea)$

System of units: Any.

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
- t is the circular ISR width
- rec is the concrete cover
- $fdpc$ 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 (195) in units Kg, cm or as (196) in units psi

$$0.65 \leq (\beta_1 = 1.05 - \frac{f'_c}{1400}) \leq 0.85 \tag{91}$$

$$0.65 \leq (\beta_1 = 0.85 - 0.05\frac{f'_c - 4000}{1000}) \leq 0.85 \tag{92}$$

- 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 diameter. For more info about this method see [5].

4.4.4 Function: eleMecanicosISRCirc

Purpose: To determine the axial load and bending resistance of a circular reinforced column cross section with a circular ISR, according to a given neutral axis depth c.

Syntax:

$eleMec = eleMecanicosISRCirc(c, a, fdpc, diam, rec, E_s, t)$

System of units: Any.

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
- a reduced neutral axis value $a = \beta_1 \cdot c$
- $diam$ is the cross-section diameter
- t is the circular ISR width
- rec is the concrete cover
- $fdpc$ is the reduced value of f'_c with the factor 0.85 as prescribed in the **ACI 318-19** code
- E_s is the Modulus of Elasticity of the reinforcing steel

Theory:

The function considers the Geometric Center (GC) (which is at a depth $\frac{diam}{2}$) **Fig. 37**, so that the resistant moment is calculated as (93), where $Fs_i = dAs \cdot E_y \epsilon_i$ for reinforcement steel.

$$M_R = (\sum Fs_i + \beta_1 ab0.85f'_c)(\frac{diam}{2} - d_i) \tag{93}$$

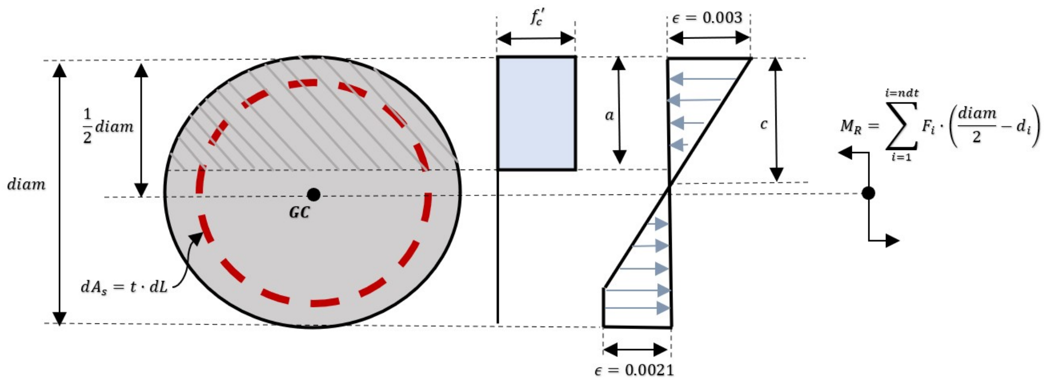


Figure 37: *Flexure-compression mechanism of a circular column cross-section.*

4.5 ISR analysis for isolated footings

4.5.1 Function: EvaluateISR1tFoot

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

Syntax:

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

System of units: Any.

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 (195) in units Kg, cm

$$0.65 \leq (\beta_1 = 1.05 - \frac{f'_c}{1400}) \leq 0.85 \tag{94}$$

- $axis$ 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. ??**)
- 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 $fdcp = 0.85 f'_c$ 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 (95), and the structural efficiency by $\frac{M_u}{M_R}$, where 0.9 represents the resistance reduction factor by code.

$$M_R = 0.9(T(d - \frac{a}{2})) \tag{95}$$

4.5.2 Function: SGD1tFootISR

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

Syntax:

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

System of units:
SI - (Kg, cm)
US - (lb, in)

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. ??**)
- *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**. [6]

5 Element rebar optimization functions

5.1 Rebar analysis for rectangular 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 optimization design process is based on Linear-Search. Only one rebar diameter is allowed to be in tension and also for compression. The algorithm traverses through all available rebar diameters (as specified in the given rebar database table) so that the rebar area (either in tension or compression) may always be equal or greater than the minimum required one specified by the ISR's widths (see **Fig. 38**).

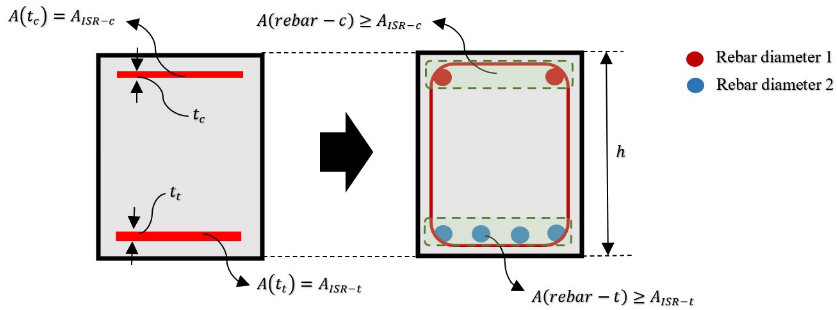


Figure 38: *Transformation of the reinforcing optimum ISR's area of a rectangular beam cross-section to a rebar design of individuals rebars, both in tension and compression.*

When no rebar option is reliable by means of the minimum rebar separation restriction, then, there is the alternative for the algorithm to dispose the rebars in packages of two, as shown in **Fig. 39**:

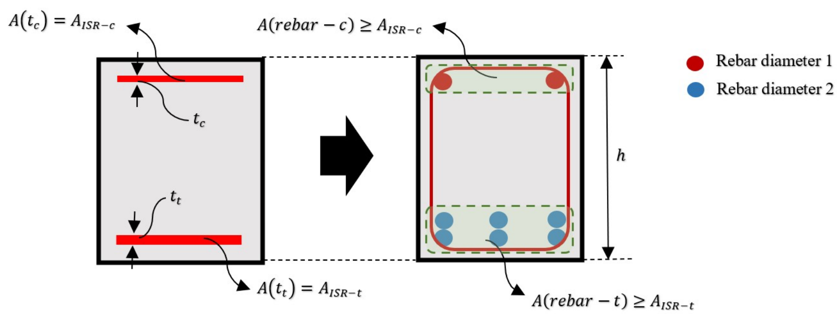


Figure 39: *Transformation of the reinforcing optimum ISR's area of a rectangular beam cross-section to a rebar design in packages of two rebars, either in tension or compression, to comply the minimum rebar separation restriction imposed by code specifications.*

5.1.2 Function: RebarOptimalDesignBeams

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

Syntax:

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

System of units: Any.

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. 40**)
- $disposition\_rebar$ local coordinates of the optimal rebar distribution over the beam cross-section

Input variables:

- $varDisponibles$ Commercial available type of rebars database

Theory:

The rebar layout options are as shown next.

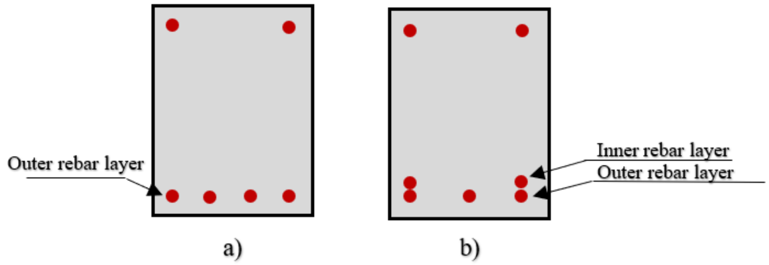


Figure 40: *Reinforcement bar options for a beam cross-section: a) in one pack, b) in two packs.*

The estimation of the minimum rebar separation constraint is determined as the max of (96):

$$sep_{min} = \begin{cases} 1in(2.54cm) \\ \frac{4}{3}d_{ag}, d_{ag} = \frac{2}{3}in \\ diam_{bar} \end{cases} \tag{96}$$

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

System of units: Any.

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 [4] code
- β_1 is determined as following (195) in units Kg, cm

$$0.65 \leq (\beta_1 = 1.05 - \frac{f'_c}{1400}) \leq 0.85 \tag{97}$$

- $disposition\_rebar$ 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.88), the resistance reduction factor ϕ (98) is then applied as (99) according to the condition of *tension controlled* or *compression controlled* for the cross-section:

$$\phi = \begin{cases} 0.65 + (\epsilon_t - 0.002)\frac{250}{3} & [0.004 \leq \epsilon_t < 0.005] \\ 0.9, & [\epsilon_t > 0.005] \end{cases} \tag{98}$$

$$M_R = \phi Mn \tag{99}$$

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

System of units: Any.

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
- $dispositionRebar$ 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 [4] code and other international codes.

The nominal resistant bending moment is determined as (48) 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 (49). For more reference of the bisection method see [5]

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:

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

System of units: Any.

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 [4]

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 (45), 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:

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

System of units: Any.

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:

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

System of units: Any.

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:

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

System of units:
SI - (kg, cm)
US - (lb, in)

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
- fy is the reinforcement steel yielding stress
- *load_conditions* are the load conditions for all three cross-sections: a vector consisting of one row and four columns as $[1 Mu_{left} Mu_{mid} Mu_{right}]$. 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:

  $[rec_left_up, rec_left_low, rec_mid_up, rec_mid_low, rec_right_up, rec_right_low]$
- *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. 41**, where rec_{min} is the minimum height concrete cover and dv_{12} is the diameter of a #12 rebar type.

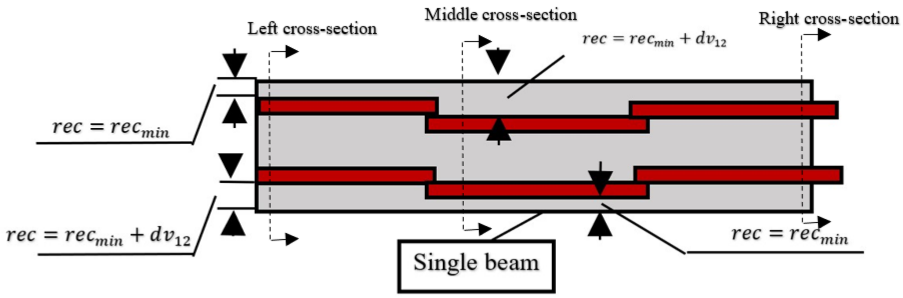


Figure 41: *Free-clash reinforcement criteria for a beam element.*

5.1.9 Function: ExportResultsBeam

Purpose: .

Syntax:

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

System of units: Any.

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 T beams

5.2.1 Function: ISR1tRebarTBeamsOptim

Purpose: to optimally design a rebar distribution over a beam cross-section from a given reinforcement area through a T-beam ISR.

Syntax:

$[sepbarsRestric, cbest, b, h, bestBarDisposition, bestCost, barTypesTen, ...$
$barTypesComp, maxEf, bestMr, area\_var\_t] = ISR1tRebarTBeamsOptim(E, bp, ht, ...$
$ba, ha, fy, fc, cover, conditions, t2, pu\_beams, Lb)$

System of units:
SI - (Kg, cm)
SI - (lb, in)

Description:

Output variables:

- *sepbarsRestric*: is the parameter that indicates if the separation restriction of rebars in tension for the cross-section in question is being complied: (1) indicates that such restriction was not complied, (0) indicates that such restriction was complied
- *cbest* : is the neutral axis depth value for the optimized reinforced T-beam cross-section considering the optimal rebar design
- *bestBarDisposition* is the array containing the local rebar coordinates over the optimally designed T-beam cross-section
- *barTypesTen* is the list of the rebar diameter's indices (according to the rebar data base table) for those rebars in tension. Is a column vector consisting of integers from $[1-7]$ (by default) of size *nrebars*
- *barTypesComp* is the list of the rebar diameter's indices (according to the rebar data base table) for those rebars in compression. Is a column vector consisting of integers from $[1-7]$ (by default) of size *nrebars*
- *maxEf* is the optimal final structural efficiency for the optimal designed beam cross-section considering the optimal rebar design
- *bestMr* is the bending resistance for the optimally designed T-beam cross-section
- *areaRebar* is a vector consisting of the total optimal rebar area in tension and compression. Vector of size 1 x 2 in format $[area - tension, area - compression]$

Input variables:

- *ba* : is the effective flange width of the T-beam cross-section
- *ht* : is total height of the T-beam cross-section
- *bp* : is the web width of the T-beam cross-section
- *ha* : is the flange thickness of the T-beam cross-section

- Lb : is the length of the beam element
- $conditions$: array containing the pure flexure loads, in format: $[nload, M_u]$ - size $nload$ x 2
- fc : is the concrete compressive strength f_c'
- $t2$: is the optimal ISR consisting of a vector of size 1 x 2 in format $[t - tension, t - compresion]$
- pu_beams : is the average unitary cost of rebar assembly for T-beams

Theory:

The optimization design process is based on Linear-Search. Only one rebar diameter is allowed to be in tension and also for compression. The algorithm traverses through all available rebar diameters (as specified in the given rebar database table) so that the rebar area (either in tension or compression) may always be equal or greater than the minimum required one specified by the ISR's widths (see **Fig. 42**).

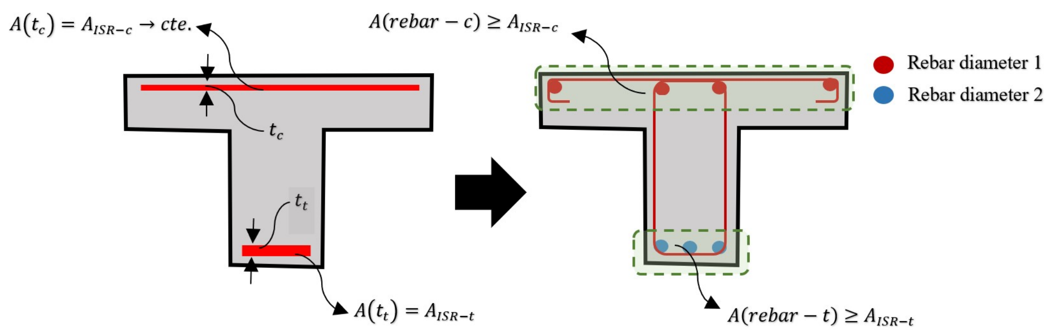


Figure 42: *Transformation of the reinforcing optimum ISR's area of a T beam cross-section to a rebar design of individuals rebars, both in tension and compression.*

When no rebar option is reliable by means of the minimum rebar separation restriction, then, there is the alternative for the algorithm to dispose the rebars in packages of two, as shown in **Fig. 43**:

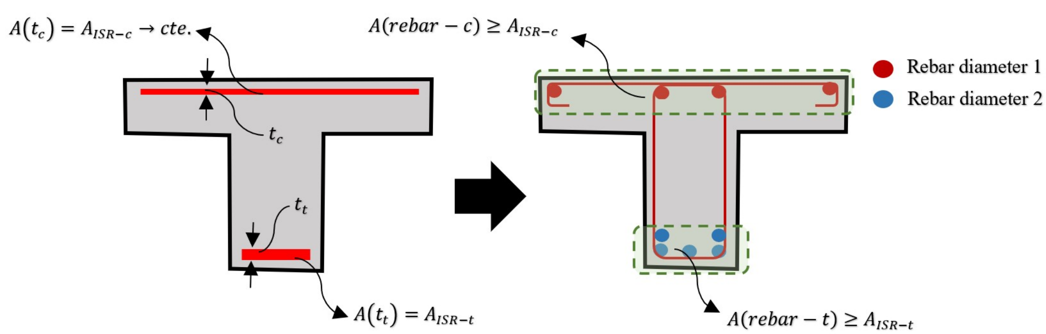


Figure 43: *Transformation of the reinforcing optimum ISR's area of a T beam cross-section to a rebar design in packages of two rebars, either in tension or compression, to comply the minimum rebar separation restriction imposed by code specifications.*

5.2.2 Function: EfRebarTBeams

Purpose: computes the structural efficiency of a rebar reinforced T-beam cross-section according to the applied load conditions (pure flexure).

Syntax:

$[Eff, Mr, c] = EfRebarTBeams(load\_conditions, bp, ht, ba, ha, Lb, fdpc, ...$
$rebarType, rebarAvailable, cover, beta1, rebarDisposition)$

Description:

Output variables:

- Eff, Mr, c :: is the structural efficiency of the reinforced beam cross-section (scale 0-1), the resistant bending moment and the corresponding neutral axis depth, respectively

Input variables:

- $load\_conditions$: array containing the pure flexure loads, in format: $[nload, M_u]$ - size $nload$ x 2
- $fdpc$: is the reduced concrete compressive strength $(0.85 \cdot f'_c)$ as prescribed in the ACI 318 code
- $beta1$: is determined as prescribed by the ACI 318 code (according to the f'_c value)
- ba : is the effective flange width of the T-beam cross-section
- ht : is total height of the T-beam cross-section
- bp : is the web width of the T-beam cross-section
- ha : is the flange thickness of the T-beam cross-section
- Lb : is the length of the beam element
- $cover$: is the concrete cover for the reinforcement
- $rebarAvailable$: is the rebar data base table with each of the eight-of-an-inch rebars available (from the smallest to the biggest diameter)
- $rebarType$: is the vector containing the rebar diameters' indices (according to their place in the rebar database table)
- $rebarDisposition$ is the array containing the rebar local coordinates over the T-beam cross-section

Theory:

The structural efficiency is computed as (100), where Mu_{max} is the max bending load imposed over the cross-section and M_R is the resistant bending moment of the reinforced T beam cross-section computed as (101) (in which Mn is computed by the function $bisectionMrRebarTBeams$ (p. 100)):

$$Eff = \frac{Mu_{max}}{M_R} \tag{100}$$

$$M_R = \phi Mn \tag{101}$$

The resistance reduction factor ϕ is a function of the steel strain in tension ϵ_t (102) which is computed as (103):

$$\phi = \begin{cases} 0.65 + (\epsilon_t - 0.002)\frac{250}{3} & 0.004 \leq \epsilon_t \leq 0.005 \\ 0.9, & \epsilon_t > 0.005 \end{cases} \tag{102}$$

$$\epsilon_t = \frac{0.003 \cdot (d - c)}{c} \tag{103}$$

5.2.3 Function: bisectionMrRebarTBeams

Purpose: to determine the neutral axis depth and the resistant bending moment of a reinforced T-beam cross-section taking on account both the steel rebar in compression and steel rebar in tension with the aid of the root bisection method so that the forces equilibrium condition sum F=0 is complied.

Syntax:

$[raiz] = bisectionMrRebarTBeams(cUno, cDos, fr, E, ha, ba, bp, ht, Lb, cover, fdpc, beta, ...$
$ea, rebarType, dispositionRebar, rebarAvailable)$

System of units: Any.

Description:

Output variables:

- $raiz$: vector that contains the output $[c, \sum F_i - > 0, MR]$ where c is the final neutral axis depth value, $\sum F_i \leftarrow 0$ is the reached forces equilibrium (which for pure flexure it should be very close to 0) and MR is the corresponding resistant bending moment

Input variables:

- $c1, c2$ are the initial values for the neutral axis depth as the bisection method requires them to begin the iterations. Such values are recommended to be $c1 \to 0$ and $c2 \to h_t$
- fr : is the applied axial force over the beam cross-section (which for pure flexure is considered as 0)
- $beta1$: is determined as prescribed by the ACI 318 code (according to the f'_c value)
- ba : is the effective flange width of the T-beam cross-section
- ht : is total height of the T-beam cross-section
- bp : is the web width of the T-beam cross-section
- ha : is the flange thickness of the T-beam cross-section
- Lb : is the length of the beam element
- E : is the Elasticity Modulus of reinforcing steel
- $cover$: is the concrete cover for the reinforcement
- ea : is the termination error for the root method
- $rebarType$: vector that contains the rebar diameters' indices of the optimal rebar design (both in tension and compression) - according to the rebar database table. The vector's size is nbars x 1 containing a number between 1 to $n\#$
- $dispositionRebar$: local coordinates of each rebar over the T-beam cross-section

- $rebarAvailable$: database of the available commercial rebar sizes. An array of size $n\#x2$ containing the the eight-of-an-inch available rebars in the first column and the rebar diameter in the second column, from the smallest to the biggest diameter

Theory:

The Bisection method takes advantage of the fact that a function is of different sign at both proximities of a root. In other words if $f(x)$ is real and continuous in an interval x_i to x_u and $f(x_i$ and $f(x_u)$ have opposite signs then there is a root between x_i and x_u (which is the reason why the initial neutral axis depth values must involve the whole cross-section height - which is the possible range of the neutral axis to lie in, that is $[0, ht]$). The location of such sign change (root) is identified more precisely by dividing the original interval x_i, x_u into subintervals in an iterative fashion. The bisection method deploys (as its names indicates it) a binary division of intervals until a convergence to the root is reached (the intervals are always divided in half). The pseudo-code is presented next (**Algorithm 5.3**):

Algoritmo 5.1: The bisection method to find roots of a one-variable continuous function

BEGIN

1.- Choose the initial values x_i and x_u to start the iteration such that $f(x_i)f(x_u) < 0$.

2.- Estimate the root by dividing the previous interval in two as:

$x_r = \frac{x_i + x_u}{2}$

3.- Make the following evaluations to determine in which subinterval the root lies:

a) If $f(x_i)f(x_r) > 0$ the root lies in the upper subinterval. Therefore, $x_i = x_r$ for the next evaluation

b) If $f(x_i)f(x_r) < 0$ the root lies in the lower subinterval. Therefore, $x_u = x_r$.

4.- Estimate the approximation error ϵ to stop the process or continue:

If $\epsilon \leq toler$ stop the process, otherwise return to step 2.

END

A good criterion to decide when to terminate the process is to establish a tolerance values under which the error approximation must lie. The computation of such error can be determined as a percent relative error ϵ_a (152):

$$\epsilon_a = |\frac{x_r^{new} - x_r^{old}}{x_r^{new}}|100\% \tag{104}$$

In other words, the bisection method traverse through the following function of the cross-section axial force resistance and the neutral axis depicted in **Fig. 44**:

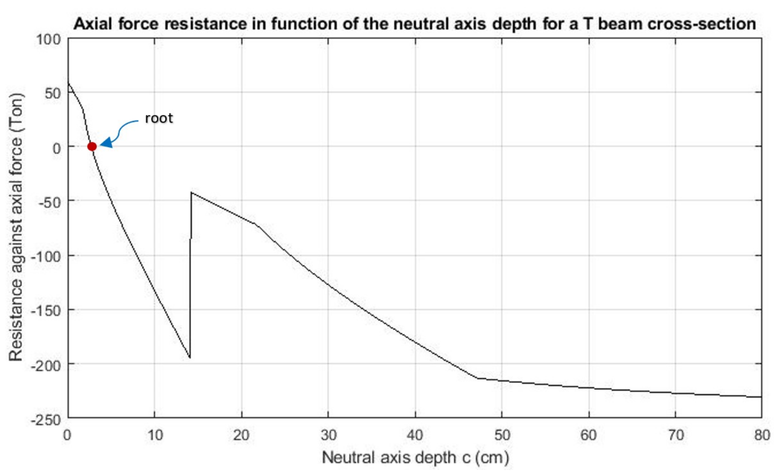


Figure 44: *Axial force resistance of a T-beam cross-section in function of the neutral axis depth.*

5.2.4 Function: eleMecanicosRebarTBeams

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

Syntax:

$eleMec = eleMecanicosRebarTBeams(c, a, fdpc, ha, ba, bp, ht, span, E, rebarType, ...$
$dispositionRebar, rebarAvailable)$

System of units: Any.

Description:

Output variables:

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

Input variables:

- E is the Modulus of Elasticity of the reinforcing steel
- $fdpc = 0.85 f'_c$: according to [4]
- c : is the nuetral axis depth value
- a : is the reduced effective neutral axis depth value $a = beta1 \cdot c$
- ba : is the effective flange width of the T-beam cross-section
- ht : is total height of the T-beam cross-section
- bp : is the web width of the T-beam cross-section
- ha : is the flange thickness of the T-beam cross-section
- $span$: is the length of the beam element
- $rebarType$: vector that contains the rebar diameters' indices of the optimal rebar design (both in tension and compression) - according to the rebar database table. The vector's size is nbars x 1 containing a number between 1 to n-diam (by default)
- $dispositionRebar$: local coordinates of each rebar over the T-beam cross-section
- $rebarAvailable$: database of the available commercial rebar sizes. A column vector containing the the eight-of-an-inch avilable rebars from the smallest

Theory:

When $a \leq ha$, it is considered that the beam cross-section is working as a T-beam cross-section, that is, with an improvement in the bending resistance because the flange is working together with the web. For this case, the effective width of the flange be is computed as (105):

$$be = min \begin{cases} bp + 16 \cdot ha \\ bp + ba \\ bp + \frac{span}{4} \end{cases} \tag{105}$$

and the centroid of the beam is computed as (106) see **Fig. 45** for more reference.

$$CS = \frac{ha \cdot be \cdot (a - ha + \frac{ha}{2}) + (a - ha) \cdot bp \cdot (\frac{a-ha}{2})}{ba \cdot ha + (a - ha) \cdot bp} \tag{106}$$

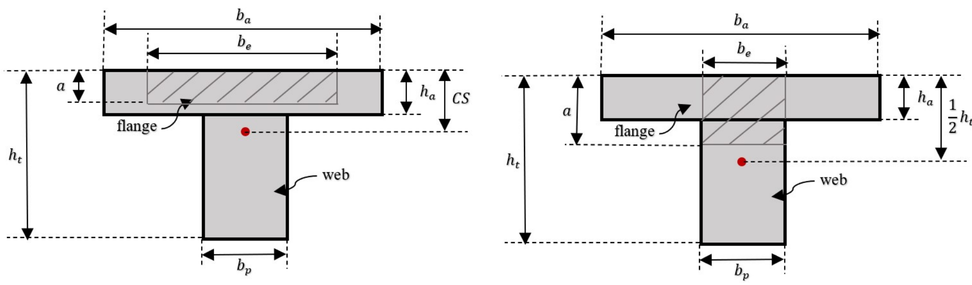


Figure 45: *T-beam cross-section working by pure flexure. The left panel depicts the effective flange width when the cross-section is working as a T-beam cross-section for* $a \leq ha$*. The right panel shows the effective cross-section working as a rectangular one when* $a > ha$*.*

On the other hand, when $a > ha$ it is considered that the beam cross-section is working as a rectangular cross-section with an effective width equal to the web's width $be = bp$ and the centroid located at a depth $CS = \frac{ht}{2}$.

Once it has been determined how the beam is working, then the resistant moment is calculated as (107), where $Fs_i = As_i E_y \epsilon_i$ for reinforcement steel.

$$M_R = (\sum Fs_i + \beta_1 ab0.85 f_c')(CS - d_i) \tag{107}$$

5.3 Rebar analysis for rectangular columns

5.3.1 Function: ExportResultsColumn

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

Syntax:

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

System of units: Any.

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

5.3.2 Function: isrColumnsSymAsym

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

Syntax:

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

System of units:
SI - (Kg, cm)
US - (lb, in)

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: $[cover_x, cover_y]$
- $cols_s ym_a sym_i sr$ is the reinforcement option parameters, options are: $"Symmetric"$ or $"ISR"$
- $condition_c racking$ 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).

5.3.3 Function: effRecColsLinearSearch

Purpose: To compute the structural efficiency of a rectangular reinforced column cross-section. The function deploys linear search to find the corresponding resistance for multiple load combinations according to their load eccentricity, from the interaction diagram's data.

Syntax:

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

System of units: Any.

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* (112) for axial load values in which $\frac{P_u}{P_{oc}} \geq 0.1$ is complied, so that $Ef = [\frac{P_n}{P_{oc}} \leq 1.0]$.

$$\frac{1}{P_n} = \frac{1}{P_{Rx}} + \frac{1}{P_{Ry}} - \frac{1}{P_{ot}} \tag{108}$$

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* (113):

$$Ef = \frac{M_{nx}}{M_{Rx}} + \frac{M_{ny}}{M_{Ry}} \leq 1.0 \tag{109}$$

Analytic geometry is applied for the determination of the resistance corresponding to each load condition P_R, M_{Rx}, M_{Ry} (see **Fig. 71**) as (189) and (190):

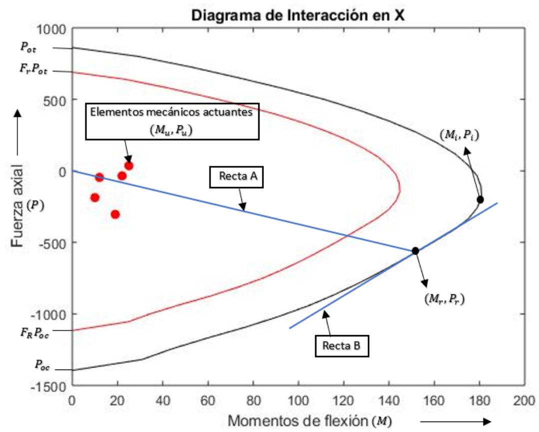


Figure 46: *Interaction diagram in the Cartesian plane as reference for comprehension of the application of analytical geometry for the computation of the structural efficiency of reinforced column cross-sections.*

$$M_r = \frac{P_{i+1} + (\frac{P_i - P_{i+1}}{M_{i+1} - M_i})}{\frac{P_u}{M_u} - \frac{P_{i+1} - P_i}{M_{i+1} - M_i}} \tag{110}$$

$$P_r = \frac{P_u}{M_u} M_r \tag{111}$$

Linear Search is applied to determine the points P_i, M_i and P_{i+1}, M_{i+1}.

5.3.4 Function: effRecColsDoubleDirecLS

Purpose: To compute the structural resistance efficiency of an asymmetrically reinforced rectangular column's cross-section given certain load conditions - with the Breler's formula (for biaxial bending compression) or the with the euclidean distance formula (for uniaxial bending compression). The function accepts both negative and positive bending moments with respect to any axis direction. A linear-search method is used to look for the corresponding resistance of each load condition according to its respective eccentricity.

Syntax:

$[maxef, tableEff, cxy] = effRecColsDoubleDirecLS(diagrama1, ...$
$diagrama2, load\_conditions, pot, poc, c\_vector_bar, c\_vector\_bar2)$

System of units: Any.

Description: .

Output variables:

- $maxef$ is the critical structural efficiency of the column cross-section given different load conditions
- $tableEff$ 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:

- $diagrama1$ is the interaction diagram data for only positive bending moments
- $diagrama2$ is the interaction diagram data for only negative bending moments
- $load\_conditions$ 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_bar1,c_vector_bar2: is the array containing the neutral axis depth values for both cross-section axis directions for all interaction diagram points: size = $[npoints + 2, 2]$. These vectors are obtained from the function $EvalAsymDoubleDirection$

Theory:

The structural efficiency Ef is computed using the Bresler's formula with the *inverse load method* (112) for axial load values in which $\frac{P_u}{P_{oc}} \geq 0.1$ is complied, so that $Ef = [\frac{P_n}{P_{oc}} \leq 1.0]$.

$$\frac{1}{P_n} = \frac{1}{P_{Rx}} + \frac{1}{P_{Ry}} - \frac{1}{P_{ot}} \tag{112}$$

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* (113):

$$Ef = \frac{M_{nx}}{M_{Rx}} + \frac{M_{ny}}{M_{Ry}} \leq 1.0 \tag{113}$$

Analytic geometry is applied for the determination of the resistance corresponding to each load condition P_R, M_{Rx}, M_{Ry} (see **Fig. 71**) as (189) and (190):

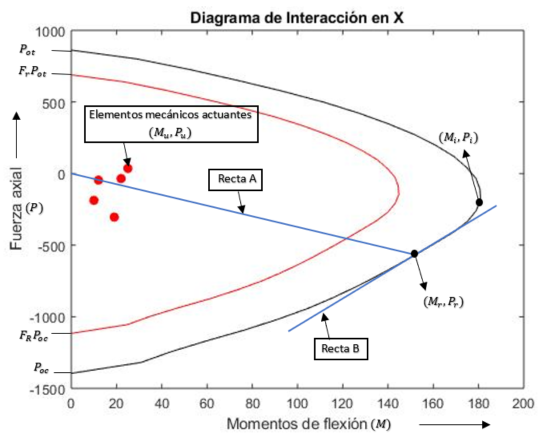


Figure 47: *Interaction diagram in the Cartesian plane as reference for comprehension of the application of analytical geometry for the computation of the structural efficiency of reinforced column cross-sections.*

$$M_r = \frac{P_{i+1} + (\frac{P_i - P_{i+1}}{M_{i+1} - M_i})}{\frac{P_u}{M_u} - \frac{P_{i+1} - P_i}{M_{i+1} - M_i}} \tag{114}$$

$$P_r = \frac{P_u}{M_u} M_r \tag{115}$$

Linear Search is applied to determine the points P_i, M_i and P_{i+1}, M_{i+1}.

5.3.5 Function: effColsRot1DirecLS

Purpose: To compute the structural efficiency of a reinforced column of rectangular cross-section. The function deploys linear search to find the corresponding resistance for a biaxial bending compression load condition according to their load eccentricity, from the interaction diagram's data with respect to the plane in which the biaxial load condition is applied.

Syntax:

$[maxef, tabEff, c] = effColsRot1DirecLS(diagrama, ...$
$load\_conditions, c\_vector\_bar)$

System of units: Any.

Description: .

Output variables:

- $maxef$ is the structural efficiency of the column cross-section for the given load condition
- $tabEff$ is the resume table of results containing the structural efficiency analysis data. Size: 1 x 5 in format:

$$[P_u, M_u, P_R, M_R, Eff]$$

- c is the neutral axis depth value corresponding to the given load condition for the rotated cross-section

Input variables:

- $diagrama$ is the interaction diagram data
- $load\_conditions$ is the array containing the load conditions: size = $1x4$ in format $[1, Pu, Mux, Muy]$
- c_vector_bar is the array containing the neutral axis depth values for all interaction diagram points: size = $npoints + 2$ x 1

Theory:

The structural efficiency Ef is computed by using the euclidean distance formula of the applied load condition and the corresponding resistance from the interaction diagram of the rotated cross-section (see **Fig. 48**):

$$eff = \frac{\sqrt{P_u^2 + M_u^2}}{\sqrt{P_R^2 + M_R^2}} \tag{116}$$

where M_u is the resultant bending moment of its components Mu_x and Mu_y (117):

$$Mu = \sqrt{Mu_x^2 + Mu_y^2} \tag{117}$$

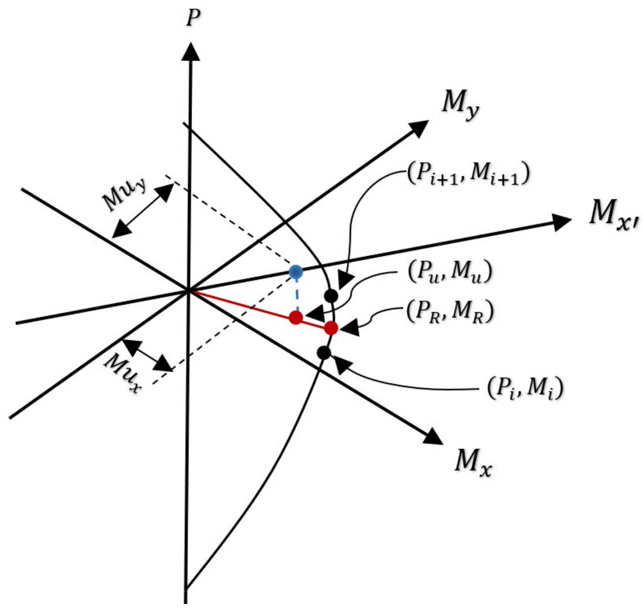


Figure 48: *Interaction diagram for a rotated column cross-section.*

Linear Search is applied to determine the points P_i, M_i and P_{i+1}, M_{i+1} to then apply analytic geometry for the estimation of the point (P_R, M_R).

$$M_R = \frac{P_{i+1} + (\frac{P_i - P_{i+1}}{M_{i+1} - M_i})}{\frac{P_u}{M_u} - \frac{P_{i+1} - P_i}{M_{i+1} - M_i}} \tag{118}$$

$$P_R = \frac{P_u}{M_u} M_R \tag{119}$$

5.3.6 Function: rotReCol2

Purpose: To rotate a rectangular reinforced concrete column cross-section according to a pair of bending moments. The function computes the new local coordinates of the rebar in the rotated system of reference, the cross-section corners and the Plastic Center location.

Syntax:

$[newdispositionRebar, newCoordCorners, newDepthCP] = rotReCol2(Mux, ...$
$Muy, dispositionRebar, b, h, CPaxis)$

System of units: Any.

Description: .

Output variables:

- *newdispositionRebar* are the new local rebar coordinates over the rotated rectangular cross-section
- *newCoordCorners* are the new local coordinates of the rectangular cross-section corners in the rotated system of reference
- *newDpethCP* is the new depth of the reinforced cross-section Plastic Center with respect to the axis X' and Y'

Input variables:

- Mux, Muy is the pair of bending moments in the original non-rotated system of reference
- *dispositionRebar* is the original local rebar coordinates over the non-rotated rectangular cross-section
- b, h : are the cross-section dimensions (width and height, respectively)
- $CPaxis$: are the original depths of the cross-section Plastic Center

Theory:

Once the primary angle of rotation α (120) has been determined, then the real rotation angle (according to the sign of the bending moments) is computed as (121):

$$\alpha = atan(\frac{Mu_x}{Mu_y}) \tag{120}$$

$$\gamma = \begin{cases} (90-\alpha)+180 & when \quad Mu_y \leq 0, Mu_x \geq 0 \quad or \quad Mu_y \leq 0, Mu_x \leq 0 \\ 90-\alpha & when \quad Mu_y \geq 0, Mu_x \leq 0 \quad or \quad Mu_y \geq 0, Mu_x \geq 0 \end{cases} \tag{121}$$

Then, the coordinate rotation for both axis takes place as (122) and (123):

$$x' = x \cdot cos\gamma + y \cdot sin\gamma \tag{122}$$

$$y' = -x \cdot sin\gamma + y \cdot cos\gamma \tag{123}$$

5.3.7 Symmetrical reinforcement

5.3.8 Function: optimalrebarColsSym

Purpose: To determine an optimal symmetrical rebar design with only one rebar diameter in individual rebars.

Syntax:

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

System of units:
SI - (Kg, cm)
US - (lb, in)

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
- $beta1$ is determined as following (195) in units Kg, cm

$$0.65 \leq (\beta_1 = 1.05 - \frac{f'_c}{1400}) \leq 0.85 \tag{124}$$

- 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 (191)

$$sep_{min} = \begin{cases} \frac{3}{2}d_b \\ \frac{4}{3}d_{ag}, d_{ag} = \frac{3}{4}in \\ 4cm \end{cases} \tag{125}$$

The resulting optimal designs can be composed of only one rebar diameter (**Fig. 49**).

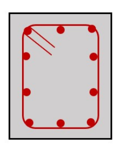


Figure 49: *Rebar design prototype: symmetrical basic rebar design.*

5.3.9 Function: optimRebarColsSym2Pack

Purpose: To determine an optimal symmetrical rebar design with only one rebar diameter either in individual rebars or in packages of two.

Syntax:

$[Mr\_col, h, Inertia\_xy\_modif, bestArea, lowestCost, ovMostEc, nvEc, ...$
$maxEfEc, bestArrangement, best_disposicion, nvxy] = optimRebarColsSym2Pack...$
$(b, h, rec, act, E, npuntos, fdpc, beta1, pu\_col\_sym, RebarAvailable, wac, height, ...$
$load\_conditions, condition\_cracking, plotRebarDesign)$

System of units:
SI - (Kg, cm)
US - (lb, in)

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
- $beta1$ is determined as following (195) in units Kg, cm

$$0.65 \leq (\beta_1 = 1.05 - \frac{f'_c}{1400}) \leq 0.85 \tag{126}$$

- 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. The restriction to accept or not a design is based entirely on the rebar separation constraint, taken as the greater of (191) and also the structural efficiency $Ef < 1.0$.

$$sep_{min} = \begin{cases} \frac{3}{2}d_b \\ \frac{4}{3}d_{ag}, d_{ag} = \frac{3}{4}in \\ 4cm \end{cases} \tag{127}$$

The resulting optimal designs can be composed of only one rebar diameter in packages of two (**Fig. 50**).

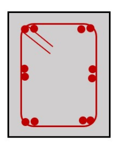


Figure 50: *Rebar design prototype: symmetrical rebar design with only one rebar diameter in packages of two.*

5.3.10 Function: superOptimalRebarSym

Purpose: To determine an optimal rebar design with bars symmetrically distributed over a rectangular cross-section. Two options are available: (1) with only one rebar reameter, (2) with two rebar diameters.

Syntax:

$[Mr\_col, h, Inertia\_xy\_modif, bestArea, bestCost, bestdiagram, bestnv, ...$
$bestEf, bestArrangement, bestDisposition, nv4, bestcxy] = superOptimalRebarSym...$
$(b, h, rec, act, E, npdiag, fdpc, beta1, pu\_col\_sym, load\_conditions, ...$
$condition\_cracking, ductility, plotRebarDesign)$

System of units:
SI - (Kg, cm)
US - (lb, in)

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
- $bestCost$ is the cost of the optimal design option
- $bestnv$ is the total number of rebars over the cross-section corresponding to the optimal design option
- $bestEf$ is the critical structural efficiency corresponding to the optimal design against the most critical of the given load conditions
- $bestArrangement$ is the list of rebar type of each rebar: size $[nbars, 1]$ (a number from 1 to 7 by default)
- $bestDisposition$ is an array containing the local coordinates of position of each rebar over the cross-section corresponding to the optimal rebar design option
- $bestdiagram$: is the interaction diagram data of the optimal rebar design
- $bestcxy$: is the vector containing the neutral axis depth values for each of the cross-section's axis, corresponding to the most critical of the given load conditions

Input variables:

- rec concrete cover of cross-section for both axis direction: $[cover_x, cover_y]$
- act optima ISR reinforcement area

- E Elasticity Modulus of reinforcement steel
- $npdiag$ 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
- $\beta1$ is determined as following (195) or (135) in units (Kg, cm) and (lb, in), respectively

$$0.65 \leq (\beta_1 = 1.05 - \frac{f'_c}{1400}) \leq 0.85 \tag{128}$$

$$0.65 \leq (\beta_1 = 0.85 - \frac{0.05 \cdot (f'_c - 4000)}{1000}) \leq 0.85 \tag{129}$$

- 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
- $ductility$: is the parameter that indicates the level of ductility demand to design the rebar, according to code specifications

Theory:
The optimization process is based on a linear search method. The restriction to accept or not a design is based entirely on the rebar separation constraint, taken as the greater of (191), and the structural efficiency (which must be less than 1.0 by default.)

$$sep_{min} = \begin{cases} \frac{3}{2}d_b \\ \frac{4}{3}d_{ag}, d_{ag} = \frac{3}{4}in \\ 4cm \end{cases} \tag{130}$$

The resulting optimal designs can be composed either of only one rebar diameter (**Fig. 51 (Left)**) or of two rebar diameters (**Fig. 51** (Right)). When analysing those design possibilities with two rebar diameters the function $sym2typeRebar$ (p. 5.3.12) takes places to generate all the permutations from a basic original symmetrical design.

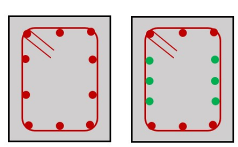


Figure 51: *Possible rebar design prototypes that can be obtained with this function: (Left) symmetrical basic rebar design, (Right) symmetrical rebar design composed of two rebar diameters.*

5.3.11 Function: superOptimalRebarSym2Pack

Purpose: To determine an optimal rebar design with bars symmetrically distributed over a rectangular cross-section in packages of two rebars. Two options are available: (1) with only one rebar reameter, (2) with two rebar diameters.

Syntax:

$[Mr\_col, h, Inertia\_xy\_modif, bestArea, bestCost, bestdiagram, bestnv, ...$
$bestEf, bestArrangement, bestDisposition, nv4, bestcxy] = superOptimalRebarSym2Pack...$
$(b, h, rec, act, E, npdiag, fdpc, beta1, pu\_col\_sym, load\_conditions, ...$
$condition\_cracking, ductility, plotRebarDesign)$

System of units:
SI - (Kg, cm)
US - (lb, in)

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
- $bestCost$ is the cost of the optimal design option
- $bestnv$ is the total number of rebars over the cross-section corresponding to the optimal design option
- $bestEf$ is the critical structural efficiency corresponding to the optimal design against the most critical of the given load conditions
- $bestArrangement$ is the list of rebar type of each rebar: size $[nbars, 1]$ (a number from 1 to 7 by default)
- $bestDisposition$ is an array containing the local coordinates of position of each rebar over the cross-section corresponding to the optimal rebar design option
- $bestdiagram$: is the interaction diagram data of the optimal rebar design
- $bestcxy$: is the vector containing the neutral axis depth values for each of the cross-section's axis, corresponding to the most critical of the given load conditions

Input variables:

- rec concrete cover of cross-section for both axis direction: $[cover_x, cover_y]$
- act optima ISR reinforcement area

- E Elasticity Modulus of reinforcement steel
- $npdiag$ 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 (195) or (135) in units (Kg, cm) and (lb, in), respectively

$$0.65 \leq (\beta_1 = 1.05 - \frac{f'_c}{1400}) \leq 0.85 \tag{131}$$

$$0.65 \leq (\beta_1 = 0.85 - \frac{0.05 \cdot (f'_c - 4000)}{1000}) \leq 0.85 \tag{132}$$

- pu_col_sym 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
- $ductility$: is the parameter that indicates the level of ductility demand to design the rebar, according to code specifications

Theory:
The optimization process is based on a linear search method. The restriction to accept or not a design is based entirely on the rebar separation constraint, taken as the greater of (191), and the structural efficiency (which must be less than 1.0 by default.)

$$sep_{min} = \begin{cases} \frac{3}{2}d_b \\ \frac{4}{3}d_{ag}, d_{ag} = \frac{3}{4}in \\ 4cm \end{cases} \tag{133}$$

The resulting optimal designs can be composed either of only one rebar diameter (**Fig. 52 (Left)**) or of two rebar diameters (**Fig. 52** (Right)). When analysing those design possibilities with two rebar diameters the function *sym2typeRebar* (p. 5.3.12) takes places to generate all the permutations from a basic original symmetrical design.

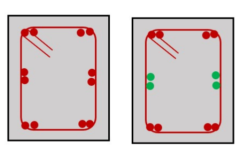


Figure 52: *Possible rebar design prototypes that can be obtained with this function: (Left) symmetrical rebar design in packages of two rebars with only one rebar diameter, (Right) symmetrical rebar design in packages of two rebars composed of two rebar diameters.*

5.3.12 Function: sym2typeRebar

Purpose: To determine an optimal symmetrical rebar design for a rectangular column's cross-section. As many as two rebar diameters are allowed to be placed simultaneously.

Syntax:

$[bestav4, relyEffList, bestArea, bestEf, bestdiagram, bestArrangement, ...$
$bestDisposition, bestMr, bestcxy, bestCost] = sym2typeRebar(ObarDisposition, ...$
$op, arraySym, RebarAvailable, b, h, fy, fdpc, beta1, E, load_conditions, npuntos, ductility, pu\_sym2\_cols)$

System of units: Any.

Description:

Output variables:

- $bestMr$: are the final resistant bending moment for both axis directions of the optimal designed cross-section
- $bestArea$: is the optimal rebar area
- $bestav4$: is the array containing the rebar area at each of the four cross-section boundaries corresponding to the optimal rebar design
- $bestCost$: is the cost of the optimal design option
- $bestEf$: is the critical structural efficiency corresponding to the optimal design option against the most critical of the give load conditions
- $bestArrangement$: is the list of rebar type indices of each rebar: size [nbars,1] (a number from 1 to by default)
- $bestDisposition$: is an array containing the local coordinates of position of each rebar over the cross-section corresponding to the optimal rebar design option
- $bestdiagram$: is the interaction diagram data of the optimal rebar design (considering only positive bending moments)
- $bestcxy$: is a vector containing the neutral axis depth values corresponding to the most critical load condition for each of the two cross-section axis

Input variables:

- b, h are the cross-section dimensions (cm)
- $fdpc$ is the f'_c reduced with the factor 0.85 according to code
- $npdiag$ number of points to be computed for the definition of the interaction diagram
- $\beta1$ is determined as following (195) or (135) in units (Kg, cm) and (lb, in), respectively

$$0.65 \leq (\beta_1 = 1.05 - \frac{f'_c}{1400}) \leq 0.85 \tag{134}$$

$$0.65 \leq (\beta_1 = 0.85 - \frac{0.05 \cdot (f'_c - 4000)}{1000}) \leq 0.85 \tag{135}$$

- rec : concrete cover of cross-section for both axis direction: $[coverX, coverY]$
- $arraySym$: is the original symmetrical rebar arrangement from which the resulting asymmetrical rebar designs take place. The vector contains the number of rebars at each of the four cross-section boundaries in format $[nbars-upper, nbars-lower, nbars-left, nbars-right]$
- $OriginalDisposition$: is the array containing the local rebar coordinates over the cross-section of the original symmetrical rebar design (an array of 2 columns and nbars rows) in format $[x_i, y_i]$
- op : is the rebar diameter index (from the rebar database table - a number between 1 to 7) of which the rebar design is composed
- $ductility$: is a parameter that indicates which ductility demand is required for the reinforcement designs. A number between 1 to 3

Theory:

This design alternative for a symmetrical reinforcement takes place from a basic conventional symmetrical design. As many as two rebar diameters are allowed to be placed simultaneously over the cross-section as shown in **Fig. 53**, for which the max total number of evaluations for each symmetrical option could be determined as (136), where i is the rebar diameter index:

$$N_{eval-max} = roundLower(\frac{b-2cover_x}{sep_{min}+db_i})(1+(i^2-i)) \tag{136}$$

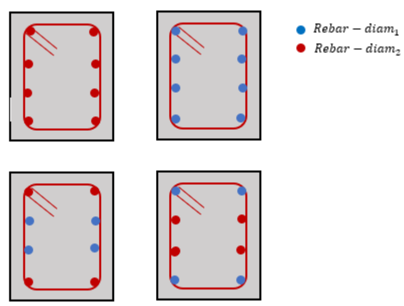


Figure 53: *et of rectangular concrete cross-sections symmetrically reinforced to illustrate the number of symmetrical designs combinations that could be generated with only two rebar diameters.*

5.3.13 Function: RebarDisposition

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

Syntax:

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

System of units: Any.

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: $[cover_x, cover_y]$

5.3.14 Function: RebarDisposition2packSym

Purpose: To compute the local position coordinates of an symmetrical rebar design over a rectangular reinforced concrete section in packages of two rebars.

Syntax:

$[dispositionRebar] = RebarDisposition2packSym(b, ...$
$h, rec, dv, nv, varcos, varsup)$

System of units: Any

Description: .

Output variables:

- $dispositionRebar$ are the local coordinates of the optimal rebar option

Input variables:

- b, h cross-section dimensions
- dv, nv is the rebar diameter of which the rebar design is composed and the total number of rebars of such design
- rec are the concrete cover values for each axis direction of the cross-section $[concrete - cover - x, concrete - cover - y]$
- $varcos, varsup$ are the number of rebars vertically (along the cross-section h dimension) and the number of rebars horizontally (along the cross-section b dimension), respectively

Theory:

The following Cartesian plane of reference shown in **Fig. 56** is considered for the computation of such local coordinates over the rectangular reinforced cross-section:

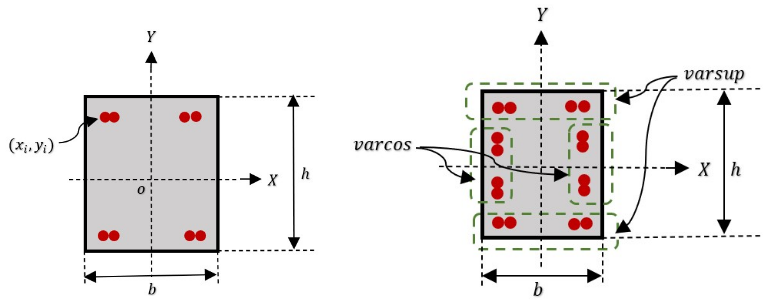


Figure 54: *(Left.) Cartesian system of reference for the computation of local rebar coordinate positions in packages of two rebars over a rectangular reinforced concrete section, (Right.) Depiction of the variables varcos, varsup.*

5.3.15 Function: diagramasDisposicion

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

Syntax:

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

System of units: Any.

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: $[nload, P_u, Mu_x, Mu_y, P_{Rx}, M_{Rx}, P_{Ry}, M_{Ry}, efficiency]$
- 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 $0.85 f'_c$ 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 (195) in units Kg, cm

$$0.65 \leq (\beta_1 = 1.05 - \frac{f'_c}{1400}) \leq 0.85 \tag{137}$$

Theory:

The max compression resistance of the column cross-section is determined as (193) 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 (194).

$$P_{oc} = 0.85f'_c(A_c - A_s) + f_y(A_s) \tag{138}$$

$$P_{ot} = f_y(A_s) \tag{139}$$

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.

Each point in the interaction is then computed from known values of force (y coordinates in the interaction diagram system of reference) by computing for each point i $F_i = -P_{oc} + df$, where df is the force differential $df = \frac{P_{ot}+P_{oc}}{npuntos-1}$. Therefore, for each F_i the bisection root method is deployed to find its corresponding M_i.

5.3.16 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:

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

System of units: Any.

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
- 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 (195) in units Kg, cm or as (196) in units lb, in

$$0.65 \leq (\beta_1 = 1.05 - \frac{f'_c}{1400}) \leq 0.85 \tag{140}$$

$$0.65 \leq (\beta_1 = 0.85 - 0.05\frac{f'_c - 4000}{1000}) \leq 0.85 \tag{141}$$

- 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
- $rebar\_disposition$ are the local coordinates of rebars over the cross-section

Theory:

The root bisection method is employed. For more reference of this method, see [5].

5.3.17 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:

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

System of units: Any.

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 (195) in units Kg, cm

$$0.65 \leq (\beta_1 = 1.05 - \frac{f'_c}{1400}) \leq 0.85 \tag{142}$$

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. 32**, so that the resistant moment is calculated as (93), where $Fs_i = As_i E_y \epsilon_i$ for reinforcement steel.

5.3.18 Function: diagramRColumnSymRebar

Purpose: To compute the interaction diagram of a symmetrical rebar arrangement consisting of only one type of rebar over a rectangular column cross-section.

Syntax:

$[diagrama, cPoints, poc, pot] = diagramRColumnSymRebar(As, b, h, E, npuntos, ...$
$fdpc, nv, beta, ov, av, disposicion_varillado)$

System of units: Any.

Description: .

Output variables:

- $diagrama$ is the array containing the interaction diagram data for both cross-section's axis. Format: $[P, MRx, FR* P, FR * MRx, ec - x, MRy, FR * P, FR * MRy, ecc - y]$
- $cPoints$ are the neutral axis depth values for each axis direction of the cross-section corresponding to each of the interaction diagram points
- poc, pot is the max resistance in compression of the cross-section and the max resistance in tension, respectively

Input variables:

- b, h are the cross-section dimensions (cm)
- $fdpc$ is the f'_c reduced with the factor 0.85 according to code
- $rebarDisposition$ are the local rebar coordinates, in format: [x,y] considering that the origin of the coordinate system of reference is at the Geometrical Center of the cross-section
- av, ov, nv: individual rebar area, the type of rebar in eighth of inches (ov/8 in) and the number of rebars to be placed, respectively
- $npuntos$ number of points to be computed for the definition of the interaction diagram
- $beta$ is determined as specified in code ACI, according to the f'_c used

Theory:

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.

Each point in the interaction is then computed from known values of force (y coordinates in the interaction diagram system of reference) by computing for each point i $F_i = -P_{oc} + df$, where df is the force differential $df = \frac{P_{ot}+P_{oc}}{npuntos-1}$. Therefore, for each F_i the bisection root method is deployed to find its corresponding M_i.

5.3.19 Asymmetrical reinforcement

5.3.20 Function: dispositionRebarAsymmetric

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

Syntax:

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

System of units: Any.

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 (191):
- 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)

Theory:

The following Cartesian plane of reference shown in **Fig. 55** is considered for the computation of such local coordinates over the rectangular reinforced cross-section:

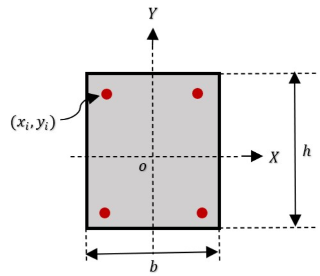


Figure 55: *Cartesian system of reference for the computation of local rebar coordinate positions in individual rebars over a rectangular reinforced concrete section.*

5.3.21 Function: RebarDisposition2packAsym

Purpose: To compute the local position coordinates of an asymmetric rebar design over a rectangular reinforced concrete section in packages of two rebars.

Syntax:

$[dispositionRebar] = RebarDisposition2packAsym(b, ...$
$h, rec, nv, varsup, varsup2, varcos, varcos2, RebarAvailable, op1, op2, op3, op4)$

System of units: Any.

Description: .

Output variables:

- $dispositionRebar$ are the local coordinates of the optimal rebar option

Input variables:

- b, h cross-section dimensions
- rec are the concrete cover values for each axis direction of the cross-section
- $RebarAvailable$ rebar database consisting of an array of size $[nbars\ x\ 3$, in format: $[\#rebar, diam, area]$
- $op1, op2, op3, op4$ resultant rebar diameters' indexes (from the RebarAvailable database rows) for each of the four cross-section boundaries (upper boundary, lower boundary, left side and right side, respectively)

Theory:

The following Cartesian plane of reference shown in **Fig. 56** is considered for the computation of such local coordinates over the rectangular reinforced cross-section:

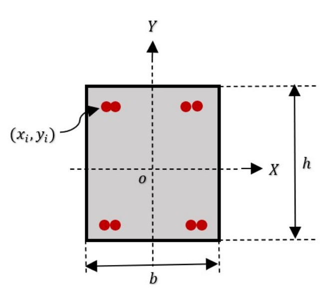


Figure 56: *Cartesian system of reference for the computation of local rebar coordinate positions in packages of two rebars over a rectangular reinforced concrete section.*

5.3.22 Function: EvaluateAsymmetric

Purpose: To compute the interaction diagram of an asymmetrically reinforced cross-section design option, only in relation to the positive bending moment direction.

Syntax:

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

System of units: Any.

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
- $rebarcombo$: is the vector containing the rebar diameters indexes for each of the four cross-section boundaries, in format: $[op1, op2, op3, op4]$ (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 (193) 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 (194).

$$P_{oc} = 0.85 f_c'(A_c - A_s) + f_y(A_s) \tag{143}$$

$$P_{ot} = f_y(A_s) \tag{144}$$

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. 72**.

Each point in the interaction is then computed from known values of force (y coordinates in the interaction diagram system of reference) by computing for each point i $F_i = -P_{oc} + df$, where df is the force differential $df = \frac{P_{ot}+P_{oc}}{npuntos-1}$. Therefore, for each F_i the bisection root method is deployed to find its corresponding M_i.

5.3.23 Function: EvalAsymDoubleDirection

Purpose: To compute the interaction diagrams of an asymmetrically reinforced cross-section design option in both directions (positive and negative) of the bending moments.

Syntax:

$[diagramaInteraccion1, diagramaInteraccion2, poc, pot, cp_axis, cvector1, ...$
$cvector2] = EvalAsymDoubleDirection(npdiag, comborebar, b, h, fy, fdpc, beta, ...$
$E, number\_rebars\_sup, number\_rebars\_inf, number\_rebars\_left,$
$number\_rebars\_right, rebarAvailable, dispositionRebar, concreteCover)$

System of units: Any.

Description: .

Output variables:

- $diagramaInteraccion1$ is the array containing the interaction diagram data for positive bending moments
- $diagramaInteraccion2$ is the array containing the interaction diagram data for negative bending moments
- pot, poc are the max axial load resistances of the reinforced cross-section in tension and compression, respectively
- cp_axis are the Plastic Center depth values for each axis direction of the cross-section: $[CPx, CPy]$ (with respect to the upper outer most concrete fibre)
- $cvector1, cvector2$ are the vectors containing the neutral axis depth values along each interaction diagrams' points (for both cross-section's axis)

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
- $comborebar$: vector of size $1x4$ containing the rebar diameters indexes for each of the four cross-section's boundaries, in format $[op1, op2, op3, op4]$ (upper boundary, lower boundary, left side and right side, respectively)
- $load\_conditions$ are the load conditions: vector of size $[nloads, 4]$ in format $[nload, Pu, Mux, Muy]$

- $npdiag$ number of points to be computed for the definition of the interaction diagram
- $concreteCover$ concrete cover values along each cross-section axis (x,y)

Theory:

The max compression resistance of the column cross-section is determined as (193) 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 (194).

$$P_{oc} = 0.85f'_c(A_c - A_s) + f_y(A_s) \tag{145}$$

$$P_{ot} = f_y(A_s) \tag{146}$$

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. 72**.

Each point in the interaction is then computed from known values of force (y coordinates in the interaction diagram system of reference) by computing for each point i $F_i = -P_{oc} + df$, where df is the force differential $df = \frac{P_{ot}+P_{oc}}{npuntos-1}$. Therefore, for each F_i the bisection root method is deployed to find its corresponding M_i. An illustration of the expected outcome from this function is shown in **Fig. 57**:

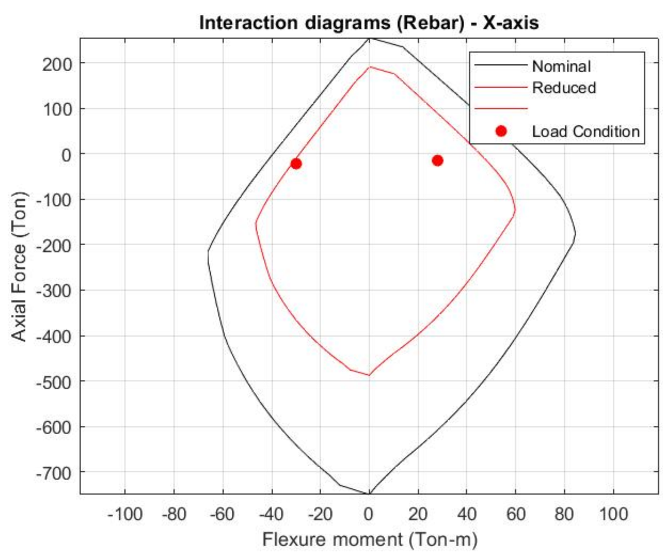


Figure 57: *Example of Interaction diagrams on both directions of bending moments (positive and negative) with respect to a cross-section axis.*

5.3.24 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:

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

System of units: Any.

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
- b, h cross-section dimensions
- $fdpc$ is the f'_c reduced with the factor 0.85 according to code
- β is determined as following (195) in units Kg, cm or as (196) in units psi

$$0.65 \leq (\beta_1 = 1.05 - \frac{f'_c}{1400}) \leq 0.85 \tag{147}$$

$$0.65 \leq (\beta_1 = 0.85 - 0.05\frac{f'_c - 4000}{1000}) \leq 0.85 \tag{148}$$

- 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 $n\#x2$ by default; in format $[\#, diam]$
- $op1, op2, op3, op4$ 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 Bisection method takes advantage of the fact that a function is of different sign at both proximities of a root. In other words if $f(x)$ is real and continuous in an interval x_i to x_u and $f(x_i$ and $f(x_u)$ have opposite signs then there is a root between x_i and x_u (which is the reason why the initial neutral axis depth values must involve the whole cross-section height - which is the possible range of the neutral axis to lie in, that is $[0, ht]$). The location of such sign change (root) is identified more precisely by dividing the original interval x_i, x_u into subintervals in an iterative fashion. The bisection method deploys (as its name indicates it) a binary division of intervals until a convergence to the root is reached (the intervals are always divided in half). The pseudo-code is presented next (**Algorithm 5.3**):

Algoritmo 5.2: The bisection method to find roots of a one-variable continuous function

BEGIN

1.- Choose the initial values x_i and x_u to start the iteration such that $f(x_i)f(x_u) < 0$.

2.- Estimate the root by dividing the previous interval in two as:

$x_r = \frac{x_i+x_u}{2}$

3.- Make the following evaluations to determine in which subinterval the root lies:

a) If $f(x_i)f(x_r) > 0$ the root lies in the upper subinterval. Therefore, $x_i = x_r$ for the next evaluation

b) If $f(x_i)f(x_r) < 0$ the root lies in the lower subinterval. Therefore, $x_u = x_r$.

4.- Estimate the approximation error ϵ to stop the process or continue:

If $\epsilon \leq toler$ stop the process, otherwise return to step 2.

END

A good criterion to decide when to terminate the process is to establish a tolerance values under which the error approximation must lie. The computation of such error can be determined as a percent relative error ϵ_a (152):

$$\epsilon_a = |\frac{x_r^{new} - x_r^{old}}{x_r^{new}}|100\% \tag{149}$$

In other words, the bisection method traverse through the following function of the cross-section axial force resistance and the neutral axis depicted in **Fig. 59**, so that, for each value of fr its corresponding value of neutral axis depth may be found as well as the bending moment:

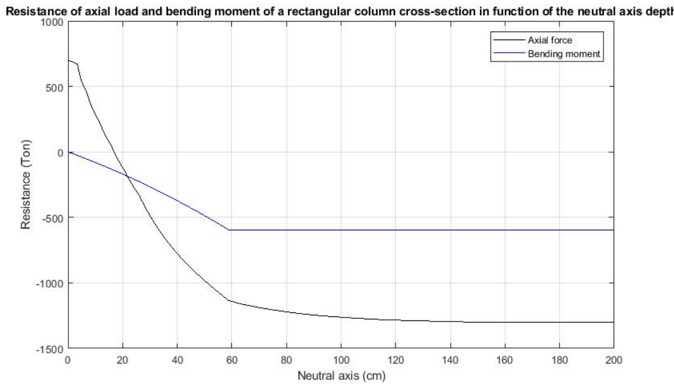


Figure 58: *Axial force resistance of a rectangular column cross-section in function of the neutral axis depth.*

5.3.25 Function: bisectionMrVarAsymRot

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

Syntax:

$[root] = bisectionMrVarAsymRot(cUno, cDos, fr, E, h, b, fdpc, beta1, ...$
$ea, nv, number\_rebars\_sup, number\_rebars\_inf, number\_rebars\_izq, ...$
$number\_rebars\_der, rebarAvailable, op1, op2, op3, op4, ...$
$dispositionRebar, cp, RotCornerSec, gamma)$

System of units: Any.

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. For columns, the value of this variable may be from the max resistance in compression of the cross-section poc to the max resistance in tension pot
- E Elasticity modulus of steel
- b, h cross-section dimensions

- $fdpc$ is the f'_c reduced with the factor 0.85 according to code
- $\beta1$ is determined as following (195) in units Kg, cm or as (196) in units psi

$$0.65 \leq (\beta_1 = 1.05 - \frac{f'_c}{1400}) \leq 0.85 \tag{150}$$

$$0.65 \leq (\beta_1 = 0.85 - 0.05 \cdot \frac{f'_c - 4000}{1000}) \leq 0.85 \tag{151}$$

- 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 rotated cross-section
- $rebarAvailable$ data base of commercial available rebars. An array of size $7x3$ by default; in format $[eight-of-an-inch, diam]$
- $op1, op2, op3, op4$ 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 rotated column cross-section
- $RotCornerSec$: are the coordinates of each of the four corners of the rotated cross-section
- $gamma$: s the angle of rotation of the column cross-section

Theory:

The Bisection method takes advantage of the fact that a function is of different sign at both proximities of a root. In other words if $f(x)$ is real and continuous in an interval x_i to x_u and $f(x_i$ and $f(x_u)$ have opposite signs then there is a root between x_i and x_u (which is the reason why the initial neutral axis depth values must involve the whole cross-section height - which is the possible range of the neutral axis to lie in, that is $[0, ht]$). The location of such sign change (root) is identified more precisely by dividing the original interval x_i, x_u into subintervals in an iterative fashion. The bisection method deploys (as its name indicates it) a binary division of intervals until a convergence to the root is reached (the intervals are always divided in half). The pseudo-code is presented next (**Algorithm 5.3**):

Algoritmo 5.3: The bisection method to find roots of a one-variable continuous function

BEGIN

1.- Choose the initial values x_i and x_u to start the iteration such that $f(x_i)f(x_u) < 0$.

2.- Estimate the root by dividing the previous interval in two as:

$x_r = \frac{x_i + x_u}{2}$

3.- Make the following evaluations to determine in which subinterval the root lies:

a) If $f(x_i)f(x_r) > 0$ the root lies in the upper subinterval. Therefore, $x_i = x_r$ for the next evaluation

b) If $f(x_i)f(x_r) < 0$ the root lies in the lower subinterval. Therefore, $x_u = x_r$.

4.- Estimate the approximation error ϵ to stop the process or continue:

If $\epsilon \leq toler$ stop the process, otherwise return to step 2.

END

A good criterion to decide when to terminate the process is to establish a tolerance values under which the error approximation must lie. The computation of such error can be determined as a percent relative error ϵ_a (152):

$$\epsilon_a = |\frac{x_r^{new} - x_r^{old}}{x_r^{new}}|100\% \tag{152}$$

In other words, the bisection method traverse through the following function of the cross-section axial force resistance and the neutral axis depicted in **Fig. 59**, so that, for each value of fr its corresponding value of neutral axis depth may be found as well as the bending moment:

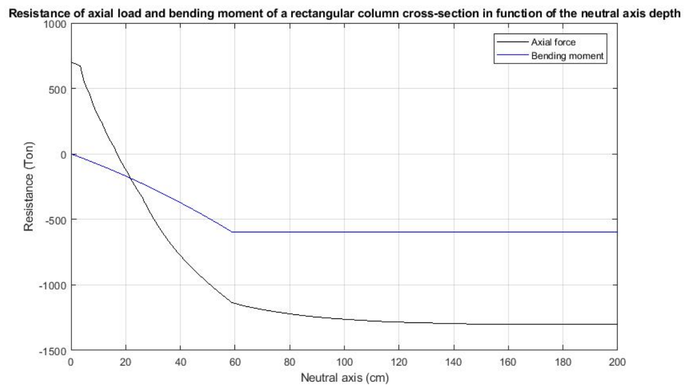


Figure 59: *Axial force resistance of a rectangular column cross-section in function of the neutral axis depth.*

5.3.26 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:

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

System of units: Any.

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 Modulus of Elasticity of reinforcing steel
- 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 (195) in units Kg, cm or as (196) in units psi

$$0.65 \leq (\beta_1 = 1.05 - \frac{f'_c}{1400}) \leq 0.85 \tag{153}$$

$$0.65 \leq (\beta_1 = 0.85 - \frac{f'_c - 4000}{1000}) \leq 0.85 \tag{154}$$

- 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 (160)) aside of the location of the Geometric Center (GC) (which is at a depth $\frac{h}{2}$ for the

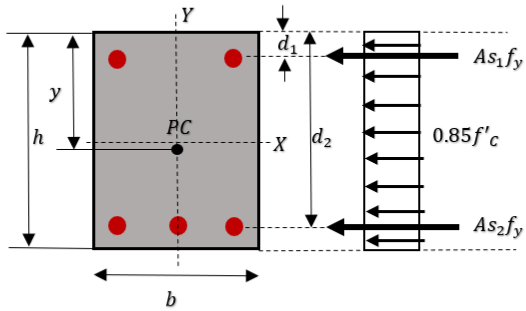


Figure 60: *Design mechanisms under flexure-compression for a symmetrically reinforced cross-section.*

latter), so that the resistant moment is calculated as (156), where $Fs_i = As_iE_y\epsilon_i$ for reinforcement steel. For the computation of the Plastic Center the **Fig. 62** is considered as reference:

$$y = \frac{0.85f_c'\frac{bh^2}{2} + \sum_{i=1}^{i=nbars} A_{s_i}f_ud_i}{0.85f_c'bh + \sum_{i=1}^{i=nbars} A_{s_i}f_y} \tag{155}$$

$$M_n = \sum_{i=1}^{i=nbars} A_{s_i}E_y\epsilon_i(y - d_i) + \beta_iabf_c''(y - \frac{a}{2}) \tag{156}$$

5.3.27 Function: eleMecBarAsymRecRot

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

Syntax:

$[eMecVar] = eleMecBarAsymRecRot(dispositionRebar, nv, number\_rebars\_sup, ...$
$number\_rebars\_inf, number\_rebars\_left, number\_rebars\_right, rebarAvailable, ...$
$op1, op2, op3, op4, b, h, c, fdpc, E, beta, CP, RotCornerSec, gamma)$

System of units: Any.

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 Modulus of Elasticity of reinforcing steel
- 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 (195) in units Kg, cm or as (196) in units psi

$$0.65 \leq (\beta_1 = 1.05 - \frac{f'_c}{1400}) \leq 0.85 \tag{157}$$

$$0.65 \leq (\beta_1 = 0.85 - 0.05 \cdot \frac{f'_c - 4000}{1000}) \leq 0.85 \tag{158}$$

- CP is the Plastic Center location for each of the cross-section axis directions
- $rebarAvailable$: data base of commercial available rebars. An array of size: $n\#$ x 2 by default, in format $[eight - of - an - inch, diam]$
- $RotCornerSec$: coordinates of each of the four rotated cross-section's corners
- $gamma$: angle of rotation for the cross-section

Theory:

Given that the rotated cross-section is reinforced asymmetrically, the function considers the location of the Plastic Center (PC) aside of the location of the Geometric Center (GC) (see **Fig. 61**). Thus, the resistant moment is calculated as (159), where $Fs_i = As_i E_y \epsilon_i$ for the reinforcing steel:

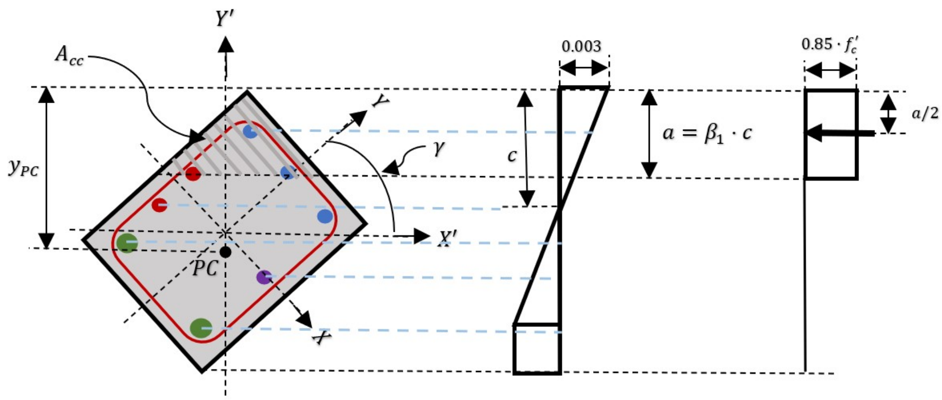


Figure 61: *Design mechanisms under flexure-compression for a rotated asymmetrically reinforced cross-section.*

$$M_n = \sum_{i=1}^{i=nbars} A_{s_i} E_y \epsilon_i (y_{PC} - d_i) + A_{cc} f_c''(y_{PC} - \frac{a}{2}) \tag{159}$$

5.3.28 Function: PlastiCenterAxis

Purpose: To compute the location of the Plastic Center for an asymmetrically reinforced concrete cross-section in the axis of reference.

Syntax:

$$[PC] = PlastiCenterAxis(fy, fdpc, b, h, dispositionRebar, rebarTypes, listrebarAvailable)$$

System of units: Any.

Description: .

Output variables:

- PC is the location of the Plastic Center from the outer most cross-section fibre in the axis of reference.

Input variables:

- $dispositionRebar$ are the local coordinates of rebars over the cross-section
- b, h are the cross-section dimensions (cm)
- fy is the yield stress of the reinforcing steel $(4200\frac{Kg}{cm^2})$
- $fdpc$ is the f'_c reduced with the factor 0.85 according to code
- $RebarAvailable$ is the available rebar database, consisting of an array of size $[nbars, 3]$, where by default $nbars = 7$, in format: $[\#rebar, diam, unit - weight]$

Theory:

The location of the Plastic Center (PC) is computed as (160)), taking as reference the outer most cross-section fibre (see **Fig. 62**):

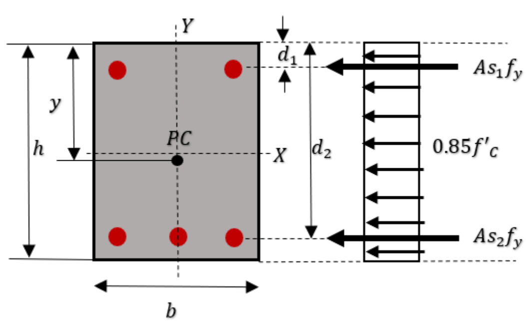


Figure 62: *Design mechanisms under flexure-compression for a symmetrically reinforced cross-section.*

$$y = PC = \frac{0.85f'_c\frac{bh^2}{2} + \sum_{i=1}^{i=nbars} A_{s_i} f_u d_i}{0.85f'_c bh + \sum_{i=1}^{i=nbars} A_{s_i} f_y} \tag{160}$$

5.3.29 Function: DiagramsAsymmetricRebar

Purpose: To compute the interaction diagram of an asymmetrical rebar arrangement over a rectangular column cross-section.

Syntax:

$[diagramaInteraccion, c\_vector, poc, pot] = DiagramsAsymmetricRebar...$
$(npoints, rebarcombo, b, h, fy, fdpc, beta, E, number\_rebars_sup, ...$
$number\_rebars\_inf, number\_rebars\_left, number\_rebars\_right, rebarAvailable, ...$
$dispositionRebar)$

System of units: Any.

Description: .

Output variables:

- *diagramaInteraccion* is the array containing the interaction diagram data for both cross-section's axis. Format: $[P, MRx, FR * P, FR * MRx, ec - x, MRy, FR * P, FR * MRy, ecc - y]$
- *c_vector* are the neutral axis depth values for each axis direction of the cross-section corresponding to each of the interaction diagram points
- poc, pot is the max resistance in compression of the cross-section and the max resistance in tension, respectively

Input variables:

- b, h are the cross-section dimensions (cm)
- $fdpc$ is the f'_c reduced with the factor 0.85 according to code
- *RebarAvailable* is the available rebar database, consisting of an array of size $[nbars, 3]$, where by default $nbars = 7$, 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, in that order
- *dispositionRebar* are the local rebar coordinates, in format: [x,y] considering that the origin of the coordinate system of reference is at the Geometrical Center of the cross-section
- *rebarcombo* Are the combination of types of rebar to be placed over the cross-section (as indices referring to their place in the "RebarAvailable" array). In this case: a vector [op1,op2,op3,op4] of size [1,4] referring to the type of rebar on each of four cross-section boundaries (upper boundary, lower boundary, left side and right side, respectively)
- *npoints* number of points to be computed for the definition of the interaction diagram

Theory:

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.

Each point in the interaction is then computed from known values of force (y coordinates in the interaction diagram system of reference) by computing for each point i $F_i = -P_{oc} + df$, where df is the force differential $df = \frac{P_{ot}+P_{oc}}{npuntos-1}$. Therefore, for each F_i the bisection root method is deployed to find its corresponding M_i.

5.3.30 Function: AssembledRebarSymAsym2

Purpose: To determine an optimal arrangement of rebars, either symmetrical or asymmetrical over a column cross-section, in individual rebars.

Syntax:

$[Mr\_col, h, Inertia\_xy\_modif, bestArea, bestCost, bestdiagram, ...$
$bestdiagram2, bestnv, bestEf, bestArrangement, bestDisposition, nv4, ...$
$bestcxy, bestCP] = AssembledRebarSymAsym2packs(b, h, rec, act, E, npuntos, fdpc, beta1, ...$
$load\_conditions, pu\_asym\_cols, condition\_cracking, ductility)$

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
- $bestCost$: is the cost of the optimal design option
- $bestEf$: is the critical structural efficiency corresponding to the optimal design option. It always should be less than 1.0
- $bestArrangement$: is the list of rebar type indices of each rebar: size [nbars,1] (a number from 1 to by default)
- $bestDisposition$: is an array containing the local coordinates of position of each rebar over the cross-section corresponding to the optimal rebar design option
- $nv4$: number of rebars at each cross-section boundary

Input variables:

- b, h are the cross-section dimensions (cm)
- $fdpc$ is the f'_c reduced with the factor 0.85 according to code
- $npdiag$ number of points to be computed for the definition of the interaction diagram
- $\beta 1$ is determined as following (195) in units Kg, cm

$$0.65 \leq (\beta_1 = 1.05 - \frac{f'_c}{1400}) \leq 0.85 \tag{161}$$

- rec : concrete cover of cross-section for both axis direction: $[coverX, coverY]$
- act : optima ISR reinforcement area

- $sepMin$: min separation of rebars constraint
- E : Modulus of Elasticity of the reinforcement steel: $E = 2.0x10^6(Kg/cm2)$
- pu_sym_cols : is the database of reinforcement assembly and construction unit cost: format by default:
$$pu\_sym\_cols = [PU\#4, PU\#5, PU\#6, PU\#8, PU\#9, PU\#10, PU\#12$$
- $condition\_cracking$: 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
- $load\_conditions$: load conditions for the column cross-section. Array of size: $[nloads, 4]$ in format $[nload, Pu, Mux, Muy]$

Theory:

Five rebar prototypes take part: $Asym - Sym4Diam$ (p. **??**), $Asym - 1Diam$ (p. 158), $Asym - 4Diam$ (p. 172), Sym_2Diam (p. **??**) and $Sym - Basic$ (p. **??**), consisting of asymmetrical arrangements of individual rebars.

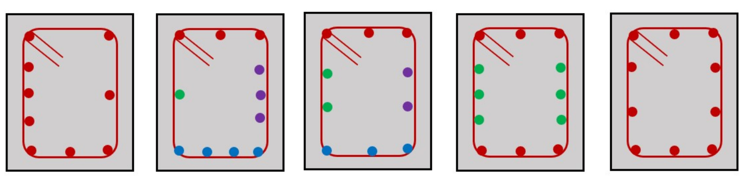


Figure 63: *Rebar prototypes of individual rebars: Asym-1Diam, Asym-4Diam, Asym-4SymDiam, Sym-2Diam, Sym-Basic.*

5.3.31 Function: AssembledRebarSymAsym2packs

Purpose: To determine an optimal arrangement of rebars, either symmetrical or asymmetrical over a column cross-section in packages of two rebars.

Syntax:

$[Inertia\_xy\_modif, h, bestArea, bestEf, bestdiagram, bestdiagram2, ...$
$bestArrangement, bestDisposition, bestMr, bestcxy, bestCP, bestCost] = ...$
$AssembledRebarSymAsym2(b, h, rec, Ac_sec_elem, E, npuntos, fdpc, beta1, ...$
$pu_sym_cols, load_conditions, condition_cracking, ductility, plotRebarDesign)$

System of units:
SI - (Kg, cm)
US - (lb, in)

Description: .

Output variables:

- $bestMr$: 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
- $bestCost$: is the cost of the optimal design option
- $bestEf$: is the critical structural efficiency corresponding to the optimal design option. It always should be less than 1.0
- $bestArrangement$: is the list of rebar type indices of each rebar: size [nbars,1] (a number from 1 to by default)
- $bestDisposition$: 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:

- b, h are the cross-section dimensions
- $fdpc$ is the f'_c reduced with the factor 0.85 according to code
- $npdiag$ number of points to be computed for the definition of the interaction diagram
- $\beta 1$ is determined as following (195) in units Kg, cm or as (196) in units psi

$$0.65 \leq (\beta_1 = 1.05 - \frac{f'_c}{1400}) \leq 0.85 \tag{162}$$

$$0.65 \leq (\beta_1 = 0.85 - 0.05\frac{f'_c - 4000}{1000}) \leq 0.85 \tag{163}$$

- rec : concrete cover of cross-section for both axis direction: $[coverX, coverY]$
- Ac_sec_elem : optima ISR reinforcement area
- $sepMin$: min separation of rebars constraint
- E : Modulus of Elasticity of the reinforcement steel
- pu_sym_cols : is the database of reinforcement assembly and construction unit cost: format by default:

$$pu\_sym\_cols = [PU\#4, PU\#5, PU\#6, PU\#8, PU\#9, PU\#10, PU\#12$$

- $condition\_cracking$: 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:

Five rebar prototypes take part: $Sym-2pack-1Diam$ (p. **??**), $Sym-2pack-2Diam$ (p. **??**), $Asym-2pack-1Diam$ (p. 162), $Asym-2pack-2Diam$ (p. **??**) and $Asym-2pack-4Diam$ (p. 164), consisting of asymmetrical arrangements of individual rebars.

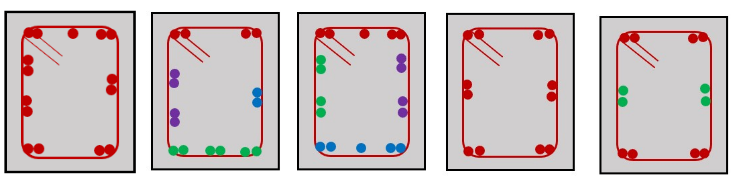


Figure 64: *Rebar prototypes of packages of two rebars: Asym-2pack-1Diam, Asym-2pack-4Diam, Asym-2pack-4SymDiam, Sym-2pack-1Diam and Sym-2pack-2Diam.*

5.3.32 Function: asym1typeRebar

Purpose: To determine an optimal arrangement of rebars asymmetrically distributed over a rectangular column's cross-section. Only one rebar diameter is allowed.

Syntax:

$[bestav4, bestnv4, relyEffList, bestArrangement, bestDisposition, ...$
$bestnv, bestMr, bestEf, bestcxy, bestCP, bestArea, bestdiagram, bestdiagram2, ...$
$bestCost] = asym1typeRebar(fdpc, nvxy, arraySymOriginal, b, h, rec, ...$
$RebarAvailable, op, av, npuntos, symCost, pu_asym_cols, load_conditions, ...$
$ductility, beta1)$

System of units:
SI - (Kg, cm).
US - (lb, in).

Description: .

Output variables:

- $bestMr$: are the final resistant bending moment for both axis directions of the optimal designed cross-section
- $bestArea$: is the optimal rebar area
- $bestCost$: is the cost of the optimal design option
- $bestEf$: is the critical structural efficiency corresponding to the optimal design option. It always should be less than 1.0
- $bestArrangement$: is the list of rebar type indices of each rebar: size [nbars,1] (a number from 1 to by default)
- $bestDisposition$: is an array containing the local coordinates of position of each rebar over the cross-section corresponding to the optimal rebar design option
- $bestdiagram$: is the interaction diagram data of the optimal rebar design (considering only positive bending moments)
- $bestdiagram2$: is the interaction diagram data of the optimal rebar design (considering only negative bending moments)
- $bestcxy$: is a vector containing the neutral axis depth values corresponding to the most critical load condition for each of the two cross-section axis
- $bestCP$: is a vector containing the Plastic Center depth values for each of the two cross-section axis (considering the asymmetry of the reinforcement)

Input variables:

- b, h are the cross-section dimensions
- $fdpc$ is the f'_c reduced with the factor 0.85 according to code

- *npdiag* number of points to be computed for the definition of the interaction diagram
- $\beta 1$ is determined as following (195) in units Kg, cm or as (196) in units *psi*

$$0.65 \leq (\beta_1 = 1.05 - \frac{f'_c}{1400}) \leq 0.85 \tag{164}$$

$$0.65 \leq (\beta_1 = 0.85 - 0.05\frac{f'_c - 4000}{1000}) \leq 0.85 \tag{165}$$

- *rec* : concrete cover of cross-section for both axis direction: $[coverX, coverY]$
- *pu_asym_cols* : is the average construction unit cost for this rebar prototype (corresponding the the $Asym-1Diam$ (p. 158)
- *arraySymOriginal* : is the original symmetrical rebar arrangement from which the resulting asymmetrical rebar designs take place. The vector contains the number of rebars at each of the four cross-section boundaries in format $[nbars-upper, nbars-lower, nbars-left, nbars-right]$
- *nvxy* is a vector containing the number of rebars in the upper or lower boundary of the cross-section and in the left or right boundary of the cross-section, in format: $[nbars-upper, nbars-left]$
- *op* : is the rebar diameter index (from the rebar database table - a number between 1 to 7) of which the rebar design is composed
- *ductility* : is a parameter that indicates which ductility demand is required for the reinforcement designs. A number between 1 to 3 (see Documentation)

Theory:

This design alternative for an asymmetrical reinforcement takes place from a basic conventional symmetrical design. Only one rebar diameter is allowed and is distributed asymmetrically over a cross-section as shown in **Fig. 65**, for which the max total number of evaluations for each symmetrical option could be determined as (166). Where nb_b and nb_h is the number of rebars at the upper/lower cross-section boundaries and at right/left cross-section boundaries, respectively. Note that $nb_b \geq 2$ must be always complied according to design codes.

$$N_{ev-max}^{non-conv} = (nb_b - 1)^2(nb_h + 1)^2 - (nb_b - 1)(nb_h + 1) \tag{166}$$

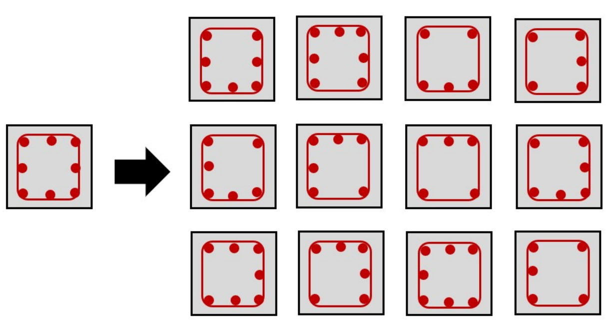


Figure 65: *Potential combinations of asymmetrical rebar arrangements from a symmetrical rebar design - reinforcement prototype* ***Asym1Diam****.*

5.3.33 Function: Asym1Diam

Purpose: To determine an optimal arrangement of rebars asymmetrically distributed over a rectangular column cross-section. Every single feasible symmetrical reinforcement option is assessed, from which then the asymmetrical designs take place by taking off certain number of rebars on each cross-section boundary. Only one rebar diameter is allowed and the function *asym1typeRebar* (p. 156) is used to compute such permutations.

Syntax:

$[Mr\_col, h, Inertia\_xy\_modif, bestArea, bestCost, bestdiagram, ...$
$bestdiagram2, bestnv, bestEf, bestArrangement, bestDisposition, nv4, ...$
$bestcxy, bestCP] = Asym1Diam(b, h, rec, act, npdiag, fdpc, beta1, ...$
$load\_conditions, pu\_asym\_cols, condition\_cracking, ductility)$

System of units:
SI - (Kg, cm).
US - (lb, in).

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 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
- $bestCost$: is the cost of the optimal design option
- $nv4$: is the number of rebars at each boundary of the column cross-section corresponding to the optimal design option
- $bestEf$: is the critical structural efficiency corresponding to the optimal design option against the given load conditions
- $bestArrangement$: is the list of rebar type indices of each rebar: size [nbars,1] (a number from 1 to by default)
- $bestDisposition$: is an array containing the local coordinates of position of each rebar over the cross-section corresponding to the optimal rebar design option
- $bestdiagram$: is the interaction diagram data of the optimal rebar design (considering only positive bending moments)
- $bestdiagram2$: is the interaction diagram data of the optimal rebar design (considering only negative bending moments)
- $bestcxy$: is a vector containing the neutral axis depth values corresponding to the most critical load condition for each of the two cross-section axis

- $bestCP$: is a vector containing the Plastic Center depth values for each of the two cross-section axis (considering the asymmetry of the reinforcement)

Input variables:

- b, h are the cross-section dimensions
- act: optimum ISR reinforcement area (from the function "$isrColumns$" (p. 70)
- $fdpc$ is the f'_c reduced with the factor 0.85 according to code
- $npdiag$ number of points to be computed for the definition of the interaction diagram
- $\beta 1$ is determined as following (195) in units Kg, cm or as (196) in units psi

$$0.65 \leq (\beta_1 = 1.05 - \frac{f'_c}{1400}) \leq 0.85 \tag{167}$$

$$0.65 \leq (\beta_1 = 0.85 - 0.05\frac{f'_c - 4000}{1000}) \leq 0.85 \tag{168}$$

- rec : concrete cover of cross-section for both axis direction: $[coverX, coverY]$
- pu_asym_cols : is the average construction unit cost for this rebar prototype (considering the Complexity Factor of Assembly for this rebar prototype multiplied by the unit construction costs of a conventional symmetrical rebar design)
- $ductility$: is a parameter that indicates which ductility demand is required for the reinforcement designs. A number between 1 to 3 (see Documentation)

Theory:

For every potential symmetrical rebar design a permutation process is performed to search for the optimal asymmetrical rebar option with only one rebar diameter that may comply with the design constraints imposed. In total, a max number of evaluations of (169) in the worst-case scenario could take place:

$$N^{as-basic}_{ev-max} = \sum_{i=1}^{i=N_{bar-types}} [N^{basic}_{ev_i} \cdot (\sum_{j}^{N_i^{sym}} (nb_{bj} - 1)^2 (nb_{hj} + 1)^2 - (nb_{bj} - 1)(nb_{hj} + 1))] \tag{169}$$

5.3.34 Function: asym1typeRebar2pack

Purpose: To determine an optimal arrangement of rebars asymmetrically distributed over a rectangular column's cross-section in packages of two rebars. Only one rebar diameter is allowed for each design option.

Syntax:

$[bestav4, bestnv4, relyEffList, bestArrangement, bestDisposition, ...$
$bestnv, bestMr, bestEf, bestcxy, bestCP, bestArea, bestdiagram, bestdiagram2, ...$
$bestCost] = asym1typeRebar2pack(fdpc, nvxy, arraySymOriginal, b, h, rec, ...$
$RebarAvailable, op, av, npdiag, symCost, pu_asym_cols, load_conditions, ductility, beta1)$

System of units:
SI - (Kg, cm).
US - (lb, in).

Description: .

Output variables:

- $bestMr$: are the final resistant bending moment for both axis directions of the optimal designed cross-section
- $bestArea$: is the optimal rebar area
- $bestCost$: is the cost of the optimal design option
- $bestEf$: is the critical structural efficiency corresponding to the optimal design option against the most critical of the load conditions
- $bestnv$ is total number of rebars over the cross-section corresponding to the optimal design option
- $bestArrangement$: is the list of rebar type indices of each rebar: size [nbars,1] (a number from 1 to by default)
- $bestDisposition$: is an array containing the local coordinates of position of each rebar over the cross-section corresponding to the optimal rebar design option
- $bestdiagram$: is the interaction diagram data of the optimal rebar design (considering only positive bending moments)
- $bestdiagram2$: is the interaction diagram data of the optimal rebar design (considering only negative bending moments)
- $bestcxy$: is a vector containing the neutral axis depth values corresponding to the most critical load condition for each of the two cross-section axis
- $bestCP$: is a vector containing the Plastic Center depth values for each of the two cross-section axis (considering the asymmetry of the reinforcement)

Input variables:

- b, h are the cross-section dimensions
- $fdpc$ is the f'_c reduced with the factor 0.85 according to code

- *npdiag* number of points to be computed for the definition of the interaction diagram
- $\beta 1$ is determined as following (195) in units Kg, cm or as (196) in units psi

$$0.65 \leq (\beta_1 = 1.05 - \frac{f'_c}{1400}) \leq 0.85 \tag{170}$$

$$0.65 \leq (\beta_1 = 0.85 - 0.05\frac{f'_c - 4000}{1000}) \leq 0.85 \tag{171}$$

- *rec* : concrete cover of cross-section for both axis direction: $[coverX, coverY]$
- *load_conditions*: is the array containing the load conditions for the column. Array size: $[nloads, 4]$ in format $[nload, Pu, Mux, Muy]$
- *pu_asym_cols* : is the average construction unit cost for this rebar prototype (corresponding to the $Asym-2pack-1Diam$ (p. 162)
- *arraySymOriginal* : is the original symmetrical rebar arrangement with packages of two rebars from which the resulting asymmetrical rebar designs take place. The vector contains the number of rebars at each of the four cross-section boundaries in format $[nbars-upper, nbars-lower, nbars-left, nbars-right]$
- *nvxy* is a vector containing the number of rebars in the upper or lower boundary of the cross-section and in the left or right boundary of the cross-section, in format: $[nbars-upper, nbars-left]$
- *op* : is the rebar diameter index (from the rebar database table - a number between 1 to $n\#$) of which the rebar design is composed
- *ductility* : is a parameter that indicates which ductility demand is required for the reinforcement designs. A number between 1 to 3 (see Documentation) - only for units in kg, cm, otherwise just put any other number

Theory:

Similar as the function *asym1typeRebar* (p. 156) this design alternative (rebar prototype) for an asymmetrical reinforcement takes place from a symmetrical design in which a homogeneous rebar diameter is distributed asymmetrically over a cross-section, but for this case, such symmetrical design is.

5.3.35 Function: Asym2pack1Diam

Purpose: To determine an optimal arrangement of rebars asymmetrically distributed over a rectangular column cross-section in packages of two rebars, through linear search. Only one rebar diameter is allowed and the function *asym1typeRebar2pack* (p. 160) is used to compute such permutations.

Syntax:

$[Mr\_col, h, Inertia\_xy\_modif, bestArea, bestCost, bestdiagram, ...$
$bestdiagram2, bestnv, bestEf, bestArrangement, bestDisposition, nv4, ...$
$bestcxy, bestCP] = Asym2pack1Diam(b, h, rec, act, npuntos, fdpc, beta1, ...$
$load\_conditions, pu\_asym\_cols, condition\_cracking, ductility)$

System of units:
SI - (Kg, cm).
US - (lb, in).

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 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
- $bestCost$: is the cost of the optimal design option
- $nv4$: is the number of rebars at each boundary of the column cross-section corresponding to the optimal design option
- $bestEf$: is the critical structural efficiency corresponding to the optimal design option against the given load conditions
- $bestArrangement$: is the list of rebar type indices of each rebar: size [nbars,1] (a number from 1 to by default)
- $bestDisposition$: is an array containing the local coordinates of position of each rebar over the cross-section corresponding to the optimal rebar design option
- $bestdiagram$: is the interaction diagram data of the optimal rebar design (considering only positive bending moments)
- $bestdiagram2$: is the interaction diagram data of the optimal rebar design (considering only negative bending moments)
- $bestcxy$: is a vector containing the neutral axis depth values corresponding to the most critical load condition for each of the two cross-section axis
- $bestCP$: is a vector containing the Plastic Center depth values for each of the two cross-section axis (considering the asymmetry of the reinforcement)

Input variables:

- b, h are the cross-section dimensions
- act: optimum ISR reinforcement area (from the function "$isrColumns$" (p. 70)
- $fdpc$ is the f'_c reduced with the factor 0.85 according to code
- $npdiag$ number of points to be computed for the definition of the interaction diagram
- $\beta1$ is determined as following (195) in units Kg, cm or as (196) in units psi

$$0.65 \leq (\beta_1 = 1.05 - \frac{f'_c}{1400}) \leq 0.85 \tag{172}$$

$$0.65 \leq (\beta_1 = 0.85 - 0.05\frac{f'_c - 4000}{1000}) \leq 0.85 \tag{173}$$

- rec : concrete cover of cross-section for both axis direction: $[coverX, coverY]$
- pu_asym_cols : is the average construction unit cost for this rebar prototype
- $ductility$: is a parameter that indicates which ductility demand is required for the reinforcement designs. A number between 1 to 3 (see Documentation)

Theory:

For every potential symmetrical rebar design in packages of two rebars a permutation process is performed to search for the optimal asymmetrical rebar option in packages of two rebars that may comply with the design constraints imposed. The function $asym1typeRebar2pack$ (p. 160).

5.3.36 Function: Asym2pack4Diam

Purpose: To determine an optimal arrangement of rebars asymmetrically distributed over a rectangular column cross-section in packages of two rebars, through linear search. A max of four rebar diameters are allowed to be placed simultaneously over the cross-section. The functions $asym1typeRebar2pack$ (p. 160) and $asymSym4Diam$ (p. 168) are used to compute such permutations.

Syntax:

$[Mr\_col, h, Inertia\_xy\_modif, bestArea, bestCost, bestdiagram, ...$
$bestdiagram2, bestnv, bestEf, bestArrangement, bestDisposition, nv4, ...$
$bestcxy, bestCP] = Asym2pack4Diam(b, h, rec, act, npuntos, fdpc, beta1, ...$
$load\_conditions, pu\_asym\_cols, condition\_cracking, ductility)$

System of units:
SI - (Kg, cm).
US - (lb, in).

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 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
- $bestCost$: is the cost of the optimal design option
- $nv4$: is the number of rebars at each boundary of the column cross-section corresponding to the optimal design option
- $bestEf$: is the critical structural efficiency corresponding to the optimal design option against the given load conditions
- $bestArrangement$: is the list of rebar type indices of each rebar: size [nbars,1] (a number from 1 to by default)
- $bestDisposition$: is an array containing the local coordinates of position of each rebar over the cross-section corresponding to the optimal rebar design option
- $bestdiagram$: is the interaction diagram data of the optimal rebar design (considering only positive bending moments)
- $bestdiagram2$: is the interaction diagram data of the optimal rebar design (considering only negative bending moments)
- $bestcxy$: is a vector containing the neutral axis depth values corresponding to the most critical load condition for each of the two cross-section axis

- *bestCP*: is a vector containing the Plastic Center depth values for each of the two cross-section axis (considering the asymmetry of the reinforcement)

Input variables:

- b, h are the cross-section dimensions
- *act*: optimum ISR reinforcement area (from the function "*isrColumns*" (p. 70)
- $fdpc$ is the f'_c reduced with the factor 0.85 according to code
- *npdiag* number of points to be computed for the definition of the interaction diagram
- $\beta 1$ is determined as following (195) in units Kg, cm or as (196) in units *psi*

$$0.65 \leq (\beta_1 = 1.05 - \frac{f'_c}{1400}) \leq 0.85 \tag{174}$$

$$0.65 \leq (\beta_1 = 0.85 - 0.05\frac{f'_c - 4000}{1000}) \leq 0.85 \tag{175}$$

- *rec* : concrete cover of cross-section for both axis direction: $[coverX, coverY]$
- *pu_asym_cols* : is the average construction unit cost for this rebar prototype
- *ductility* : is a parameter that indicates which ductility demand is required for the reinforcement designs. A number between 1 to 3 (see Documentation)

Theory:

For every potential symmetrical rebar design in packages of two rebars a permutation process is performed to search for the optimal asymmetrical rebar option in packages of two rebars that may comply with the design constraints imposed. The function *asym1typeRebar2pack* (p. 160).

5.3.37 Function: Asym2packSym4Diam

Purpose: To determine an optimal arrangement of rebars asymmetrically distributed over a rectangular column cross-section in packages of two rebars, through linear search. A max of four rebar diameters are allowed to be placed simultaneously over the cross-section and their distribution is symmetrical in quantity. The function *asymSym4Diam* (p. 168) is used to compute such permutations.

Syntax:

$[Mr\_col, h, Inertia\_xy\_modif, bestArea, bestCost, bestdiagram, ...$
$bestdiagram2, bestnv, bestEf, bestArrangement, bestDisposition, nv4, ...$
$bestcxy, bestCP] = Asym2packSym4Diam(b, h, rec, act, E, npuntos, fdpc, beta1, ...$
$load\_conditions, pu\_asym\_cols, condition\_cracking, ductility)$

System of units:
SI - (Kg, cm).
US - (lb, in).

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 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
- $bestCost$: is the cost of the optimal design option
- $nv4$: is the number of rebars at each boundary of the column cross-section corresponding to the optimal design option
- $bestEf$: is the critical structural efficiency corresponding to the optimal design option against the most critical given load condition
- $bestArrangement$: is the list of rebar type indices of each rebar: size [nbars,1] (a number from 1 to by default)
- $bestDisposition$: is an array containing the local coordinates of position of each rebar over the cross-section corresponding to the optimal rebar design option
- $bestdiagram$: is the interaction diagram data of the optimal rebar design (considering only positive bending moments)
- $bestdiagram2$: is the interaction diagram data of the optimal rebar design (considering only negative bending moments)
- $bestcxy$: is a vector containing the neutral axis depth values corresponding to the most critical load condition for each of the two cross-section axis

- $bestCP$: is a vector containing the Plastic Center depth values for each of the two cross-section axis (considering the asymmetry of the reinforcement)

Input variables:

- b, h are the cross-section dimensions
- act: optimum ISR reinforcement area (from the function "$isrColumns$" (p. 70)
- $fdpc$ is the f'_c reduced with the factor 0.85 according to code
- $npdiag$ number of points to be computed for the definition of the interaction diagram
- $\beta 1$ is determined as following (195) in units Kg, cm or as (196) in units psi

$$0.65 \leq (\beta_1 = 1.05 - \frac{f'_c}{1400}) \leq 0.85 \tag{176}$$

$$0.65 \leq (\beta_1 = 0.85 - 0.05\frac{f'_c - 4000}{1000}) \leq 0.85 \tag{177}$$

- rec : concrete cover of cross-section for both axis direction: $[coverX, coverY]$
- pu_asym_cols : is the average construction unit cost for this rebar prototype
- $ductility$: is a parameter that indicates which ductility demand is required for the reinforcement designs. A number between 1 to 3 (see Documentation)

Theory:

For every potential symmetrical rebar design in packages of two rebars a permutation process is performed to search for the optimal asymmetrical rebar option in packages of two rebars that may comply with the design constraints imposed. The function $asymSym4Diam$ (p. 168) to perform such permutations.

5.3.38 Function: asymSym4Diam

Purpose: To determine an optimal asymmetrical arrangement of rebars distributed over a rectangular column's cross-section. The distributions of rebars are symmetrical in quantity of rebars but asymmetrical in rebar diameters. As many as four rebar diameters are allowed to be placed simultaneously.

Syntax:

$[bestav4, relyEffList, bestArea, bestEf, bestdiagram, bestdiagram2, ...$
$bestArrangement, bestDisposition, bestMr, bestcxy, bestCP, bestCost] = ...$
$asymSym4Diam(OriginalDisposition, op, arrayOriginal, RebarAvailable, rec, ...$
$b, h, fy, fdpc, beta1, E, pu\_asym\_cols, load\_conditions, npuntos, ductility)$

System of units:
SI - (Kg, cm).
US - (lb, in).

Description: .

Output variables:

- $bestMr$: are the final resistant bending moment for both axis directions of the optimal designed cross-section
- $bestArea$: is the optimal rebar area
- $bestav4$: is the array containing the rebar area at each of the four cross-section boundaries corresponding to the optimal rebar design
- $bestCost$: is the cost of the optimal design option
- $bestEf$: is the critical structural efficiency corresponding to the optimal design option against the most critical of the give load conditions
- $bestArrangement$: is the list of rebar type indices of each rebar: size [nbars,1] (a number from 1 to by default)
- $bestDisposition$: is an array containing the local coordinates of position of each rebar over the cross-section corresponding to the optimal rebar design option
- $bestdiagram$: is the interaction diagram data of the optimal rebar design (considering only positive bending moments)
- $bestdiagram2$: is the interaction diagram data of the optimal rebar design (considering only negative bending moments)
- $bestcxy$: is a vector containing the neutral axis depth values corresponding to the most critical load condition for each of the two cross-section axis
- $bestCP$: is a vector containing the Plastic Center depth values for each of the two cross-section axis (considering the asymmetry of the reinforcement)

Input variables:

- b, h are the cross-section dimensions (cm)

- $fdpc$ is the f'_c reduced with the factor 0.85 according to code
- $npdiag$ number of points to be computed for the definition of the interaction diagram
- $\beta 1$ is determined as following (195) in units Kg, cm or as (196) in units psi

$$0.65 \leq (\beta_1 = 1.05 - \frac{f'_c}{1400}) \leq 0.85 \quad (178)$$

$$0.65 \leq (\beta_1 = 0.85 - 0.05\frac{f'_c - 4000}{1000}) \leq 0.85 \quad (179)$$

- rec : concrete cover of cross-section for both axis direction: $[coverX, coverY]$
- pu_asym_cols : is the average construction unit cost for this rebar prototype
- $arrayOriginal$: is the original symmetrical rebar arrangement from which the resulting asymmetrical rebar designs take place. The vector contains the number of rebars at each of the four cross-section boundaries in format $[nbars - upper, nbars - lower, nbars - left, nbars - right]$
- $OriginalDisposition$: is the array containing the local rebar coordinates over the cross-section of the original symmetrical rebar design (an array of 2 columns and nbars rows) in format $[x_i, y_i]$
- op : is the rebar diameter index (from the rebar database table - a number between 1 to $n\#$) of which the rebar design is composed
- $ductility$: is a parameter that indicates which ductility demand is required for the reinforcement designs. A number between 1 to 3 (which is considered only when working in Kg, cm)

Theory:

This design alternative for an asymmetrical reinforcement takes place from a basic conventional symmetrical design. As many as four rebar diameters are allowed and are distributed symmetrically in number over the cross-section as shown in **Fig. 66**, for which the max total number of evaluations for each symmetrical option could be determined as (180), where i is the rebar diameter index:

$$N_{eval-max} = roundLower(\frac{b - 2cover_x}{sep_{min} + db_i}) \cdot (i^4 - i^2) \quad (180)$$

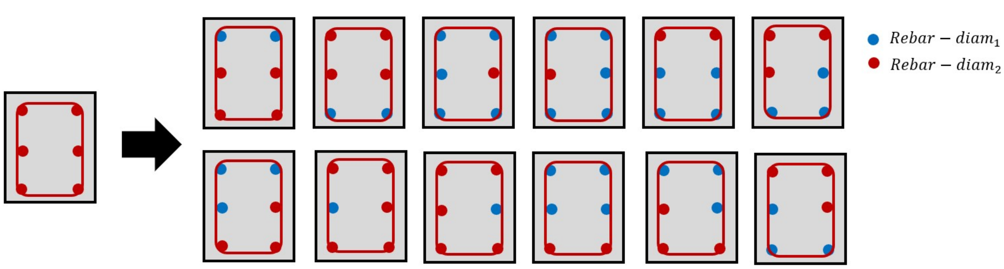


Figure 66: *Potential combinations of asymmetrical rebar arrangements from a symmetrical rebar design for the reinforcement prototype* ***Asym-Sym4Diam*** *by considering* $i = 2$.

5.3.39 Function: AsymmSym4Diam

Purpose: To determine an optimal arrangement of rebars asymmetrically distributed over a rectangular column cross-section, through linear search. As many as four rebar diameters are allowed to be placed simultaneously in an asymmetrical fashion, although symmetrically in number for each cross-section's boundary. The function *asymSym4Diam* (p. 168) is used to compute such permutations.

Syntax:

$[Mr\_col, h, Inertia\_xy\_modif, bestArea, bestCost, bestdiagram, ...$
$bestdiagram2, bestnv, bestEf, bestArrangement, bestDisposition, nv4, ...$
$bestcxy, bestCP] = AsymmSym4Diam(b, h, rec, act, E, npuntos, fdpc, beta1, ...$
$load\_conditions, pu\_asym\_cols, condition\_cracking, ductility)$

System of units:
SI - (Kg, cm).
US - (lb, in).

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 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
- $bestCost$: is the cost of the optimal design option
- $nv4$: is the number of rebars at each boundary of the column cross-section corresponding to the optimal design option
- $bestEf$: is the critical structural efficiency corresponding to the optimal design option against the given load conditions
- $bestArrangement$: is the list of rebar type indices of each rebar: size [nbars,1] (a number from 1 to by default)
- $bestDisposition$: is an array containing the local coordinates of position of each rebar over the cross-section corresponding to the optimal rebar design option
- $bestdiagram$: is the interaction diagram data of the optimal rebar design (considering only positive bending moments)
- $bestdiagram2$: is the interaction diagram data of the optimal rebar design (considering only negative bending moments)
- $bestcxy$: is a vector containing the neutral axis depth values corresponding to the most critical load condition for each of the two cross-section axis

- $bestCP$: is a vector containing the Plastic Center depth values for each of the two cross-section axis (considering the asymmetry of the reinforcement)

Input variables:

- b, h are the cross-section dimensions
- act: optimum ISR reinforcement area (from the function "$isrColumns$" (p. 70)
- $fdpc$ is the f'_c reduced with the factor 0.85 according to code
- $npdiag$ number of points to be computed for the definition of the interaction diagram
- $\beta 1$ is determined as following (195) in units Kg, cm or as (196) in units psi

$$0.65 \leq (\beta_1 = 1.05 - \frac{f'_c}{1400}) \leq 0.85 \tag{181}$$

$$0.65 \leq (\beta_1 = 0.85 - 0.05\frac{f'_c - 4000}{1000}) \leq 0.85 \tag{182}$$

- rec : concrete cover of cross-section for both axis direction: $[coverX, coverY]$
- pu_asym_cols : is the average construction unit cost for this rebar prototype
- $ductility$: is a parameter that indicates which ductility demand is required for the reinforcement designs. A number between 1 to 3 (see Documentation)

Theory:

For every potential symmetrical rebar design a permutation process is performed to search for the optimal asymmetrical rebar option that may comply with the design constraints imposed. The function $asymSym4Diam$ (p. 168) to perform the permutations of rebar diameters from each original symmetrical design.

5.3.40 Function: Asym4Diam

Purpose: To determine an optimal arrangement of rebars asymmetrically distributed over a rectangular column cross-section, through linear search. A max of four rebar diameters are allowed to be placed simultaneously over the cross-section, both in number and in rebar diameter. The functions *asym1typeRebar* (p. 156) and *asymSym4Diam* (p. 168) are used to compute such permutations of rebar diameters and number of rebar on each of the four cross-section's boundaries.

Syntax:

$[Mr\_col, h, Inertia\_xy\_modif, bestArea, bestCost, bestdiagram, ...$
$bestdiagram2, bestnv, bestEf, bestArrangement, bestDisposition, nv4, ...$
$bestcxy, bestCP] = Asym2pack4Diam(b, h, rec, act, npuntos, fdpc, beta1, ...$
$load\_conditions, pu\_asym\_cols, condition\_cracking, ductility)$

System of units:
SI - (Kg, cm).
US - (lb, in).

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 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
- $bestCost$: is the cost of the optimal design option
- $nv4$: is the number of rebars at each boundary of the column cross-section corresponding to the optimal design option
- $bestEf$: is the critical structural efficiency corresponding to the optimal design option against the given load conditions
- $bestArrangement$: is the list of rebar type indices of each rebar: size [nbars,1] (a number from 1 to by default)
- $bestDisposition$: is an array containing the local coordinates of position of each rebar over the cross-section corresponding to the optimal rebar design option
- $bestdiagram$: is the interaction diagram data of the optimal rebar design (considering only positive bending moments)
- $bestdiagram2$: is the interaction diagram data of the optimal rebar design (considering only negative bending moments)
- $bestcxy$: is a vector containing the neutral axis depth values corresponding to the most critical load condition for each of the two cross-section axis

- $bestCP$: is a vector containing the Plastic Center depth values for each of the two cross-section axis (considering the asymmetry of the reinforcement)

Input variables:

- b, h are the cross-section dimensions
- act: optimum ISR reinforcement area (from the function "$isrColumns$" (p. 70)
- $fdpc$ is the f'_c reduced with the factor 0.85 according to code
- $npdiag$ number of points to be computed for the definition of the interaction diagram
- $\beta 1$ is determined as following (195) in units Kg, cm or as (196) in units Kg, cm

$$0.65 \leq (\beta_1 = 1.05 - \frac{f'_c}{1400}) \leq 0.85 \tag{183}$$

$$0.65 \leq (\beta_1 = 0.85 - 0.05\frac{f'_c - 4000}{1000}) \leq 0.85 \tag{184}$$

- rec : concrete cover of cross-section for both axis direction: $[coverX, coverY]$
- pu_asym_cols : is the average construction unit cost for this rebar prototype
- $ductility$: is a parameter that indicates which ductility demand is required for the reinforcement designs. A number between 1 to 3 (see Documentation)

Theory:

For every potential symmetrical rebar design in packages of two rebars a permutation process is performed to search for the optimal asymmetrical rebar option in packages of two rebars that may comply with the design constraints imposed. The function $asym1typeRebar2pack$ (p. 160).

5.3.41 Function: InteractionDiagramAxis

Purpose: To compute the structural efficiency the interaction diagram with respect to a rotated axis of a rectangular column cross-section asymmetrical reinforced.

Syntax:
$[diagramIntAxis1, pot, poc, cvectorX, newdispositionRebar, ...$
$newCoordCorners, newCP, gamma] = InteractionDiagramAxis...$
$(npdiag, comborebar, b, h, fy, fdpc, beta1, E, number\_rebars\_sup, ...$
$number\_rebars\_inf, number\_rebars\_left, number\_rebars\_right, ...$
$rebarAvailable, dispositionRebar, Mux, Muy)$

System of units: Any.

Description: .

Output:

- $diagramIntAxis1$ is the array containing the interaction diagram data whose rebars in tension are at the bottom of the cross-section. See documentation
- pot, poc : are tha max axial load resistance in tension and compression, respectively
- $newCP$: are the Plastic Center depths over the rotated cross-section (with respect to the upper most concrete corner) in both perpendicular axis (X' and Y') directions
- $cvectorX$: is the neutral axis depth values corresponding to each of the points of the interaction diagram of the axis X' in quest (fro the upper cross-section corner downwards)
- $newdispositionRebar$: is the array containing the local coordinates of the rebars distributed over the rotated rectangular cross-section
- $gamma$: is the angle of rotation for the cross-section

Input variables:

- $npdiag$ is the number of points for the computation of the interaction diagram with respect to the bending compression load's axis
- b, h are the cross-section dimensions (width and height, respectively)
- $comborebar$: vector of size $1x4$ containing the four rebar diameters' indices over each of the four cross-section's boundaries in order: *upper boundary, lower boundary, left side and right side*
- fy : is the yield stress for the reinforcing steel ($4200Kg/cm^2$ by default)
- $fdpc$: is the reduced concrete's compressive strength by the application of the reduction resistance factor of 0.85 (according to the ACI 318 code)
- E : is the Modulus of Elasticity of the reinforcing steel

- $number_rebars_sup, number_rebars_inf, number_rebars_left, number_rebars_right$: number of rebars to placed on each of the cross-section boundaries
- $rebarAvailable$: is the rebar database array, containing the commercially available eight-of-an-inch rebars. Size 7 x 2 (by default) in format $[eight-of-an-inch, diameter]$
- $dispositionRebar$: is the array containing the rebar local coordinates over the non-rotated cross-section (in the original system of reference)

Theory:

The function first takes the rectangular cross-section in its original coordinate system to determine the Plastic Center location. Then, the rotation of such cross-section takes place with the given bending moment actions Mu_x, Mu_y as (185). In this rotation process, the original rebar coordinates, the cross-section's corners' coordinates and the Plastic Center location are transformed and mapped in the new rotated system of reference, as depicted in **Fig. 67**. Finally, the computation of the interaction diagram in this new rotated system of reference with respect to the X' axis takes place.

$$\alpha = atan(\frac{Mu_x}{Mu_y}) \tag{185}$$

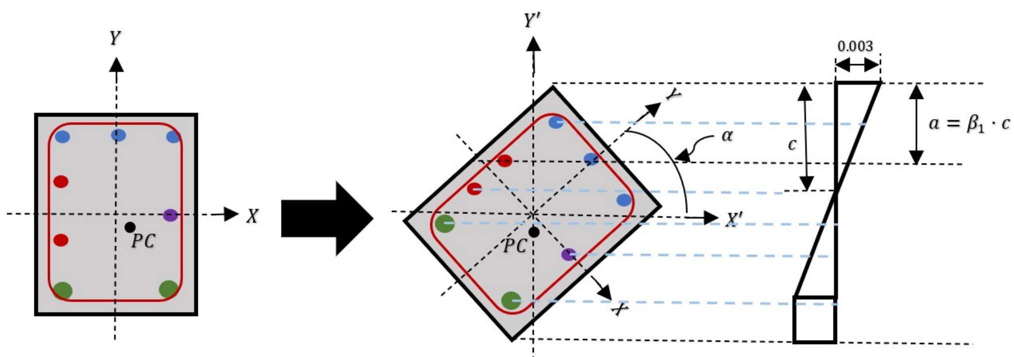


Figure 67: *Rotation of a rectangular column cross-section.*

For the transformation of the original coordinates into the new rotated system of reference, the function $rotReCol2$ (p. 113) is used.

5.3.42 Function: multiDiagAxisColRec

Purpose: To compute the structural efficiency of the most critical of the given biaxial load conditions for a rectangular asymmetrically reinforced column cross-section. A 3D analysis is carried on for the computation of the interaction diagrams in the plane in which each of the load conditions are applied.

Syntax:
$[tablaEff, iloadmax, maxLoadCondition, maxgamma, diagramIntAxis1, ...$
$rotdispositionRebar, rotSection, cmax, CP] = multiDiagAxisColRec(b, h, ...$
$load\_conditions, comborebar, npuntos, fy, fdpc, beta1, E, numberRebars1, ...$
$numberRebars2, numberRebars3, numberRebars4, rebarAvailable, dispositionRebar)$

System of units: Any.

Description: .

Output:

- $tablaEff$: is the resume table of the structural efficiency analysis for each load condition. Size: $nloads$ x 5, in format: $[P_u, M_u, P_R, M_R, Eff]$
- $iloadmax$: is the index of the most critical load condition (according to the order in which they were given in the "load condition" array)
- $maxLoadCondition$: is the structural efficiency analysis of the most critical load condition. Size: 1 x 3, in format: $[1, Pu, Mxy]$, where $Mxy = \sqrt{Mux^2 + Muy^2}$
- $maxgamma$: is the angle of rotation that the cross-section had to undergo for the most critical load condition
- $diagramIntAxis1$: is the array containing the the interaction diagram's data for the most critical of the given load conditions
- $rotdispositionRebar$: is the array containing the rotated rebar coordinates over the cross-section corresponding to the most critical of the given load conditions
- $rotSection$: are the coordinates of the rotated cross-section for the most critical of the given load conditions
- $cmax$: is the neutral axis depth value of the rotated cross-section for the most critical of the load conditions
- CP : are the Plastic Center depth values corresponding to the rotated cross-section for the most critical of the given load conditions

Input variables:

- b, h : are the cross-section dimensions: width and height, respectively
- $load\_conditions$: vector containing the multiple applied load combinations. Size: nloads x 4, in format: $[nload, Pu, Mux, Muy]$

- $comborebar$: is the vector containing the four rebar diameters' indices (from the give rebar database table), in format:

  $[index - bar - upper, index - bar - lower, index - bar - left, index - bar - right]$

- $npdiag$: number of points to be computed for the interaction diagram

- fy : yield stress of the reinforcing steel $(4200Kg/cm^2)$

- $fdpc$: is the concrete compressive strength $(4200Kg/cm^2)$

- $\beta 1$ is determined as following (195) in units Kg, cm

$$0.65 \leq (\beta_1 = 1.05 - \frac{f'_c}{1400}) \leq 0.85 \tag{186}$$

- E : is the Modulus of Elasticity of the reinforcing steel

- $numberRebars1, numberRebars2, numberRebars3, numberRebars4$: are the number of rebars on each cross-section's boundary (upper boundary, lower boundary, left boundary, right boundary)

- $rebarAvailable$: is the rebar database table

- $dispositionRebar$: is the array that contains the rebar coordinates over the cross-section

Theory:

The function computes an interaction diagram for each of the given load conditions, so that each interaction diagram is in the plane of action of each corresponding load; the function $InteractionDiagramAxis$ (p. 174) is used for that purpose. Then, once the interaction diagram is computed then structural efficiency is analysed with the function $effColsRot1DirecLS$ (p. 111) and saved. At the end of each loop (for each load condition), the load condition with the most critical (closer to 1.0) is saved along with all the corresponding design/analysis data such as the interaction diagram, the angle of rotation for the cross-section and rebar distribution.

5.3.43 Function: InteracSurfaceColRec

Purpose: to compute the 3D interaction surface of a reinforced rectangular column cross-section with symmetrical or asymmetrical rebar.

Syntax:
$[supX, supY, supZ] = InteracSurfaceColRec(b, h, ...$
$comborebar, npuntos, fy, fdpc, beta1, E, numberRebars1, ...$
$numberRebars2, numberRebars3, numberRebars4, rebarAvailable, dispositionRebar)$

System of units: Any.

Description: .

Output:

- $supX, supY, supZ$: are the points of the interaction surface in the 3D coordinate system as $[x, y, z]$

Input variables:

- b, h : are the cross-section dimensions: width and height, respectively
- $comborebar$: is the vector containing the four rebar diameters' indices (from the give rebar database table), in format:

  $[index - bar - upper, index - bar - lower, index - bar - left, index - bar - right]$
- $npdiag$: number of points to be computed for the interaction diagram at each angle step
- fy : yield stress of the reinforcing steel $(4200 Kg/cm^2)$
- $fdpc$: is the concrete compressive strength $(4200 Kg/cm^2)$
- $\beta 1$ is determined as following (195) in units Kg, cm

$$0.65 \leq (\beta_1 = 1.05 - \frac{f'_c}{1400}) \leq 0.85 \tag{187}$$

- E : is the Modulus of Elasticity of the reinforcing steel
- $numberRebars1, numberRebars2, numberRebars3, numberRebars4$: are the number of rebars on each cross-section's boundary (upper boundary, lower boundary, left boundary, right boundary)
- $dispositionRebar$: is the array that contains the rebar coordinates over the cross-section

Theory:

The function rotates a rectangular reinforced cross-section 360 so that on each angle step (of 1) an interaction diagram is computed with the aid of function $InteractionDiagramAxis$ (p. 174) to give way then to the interaction surface such as the one shown in **Fig. 68**:

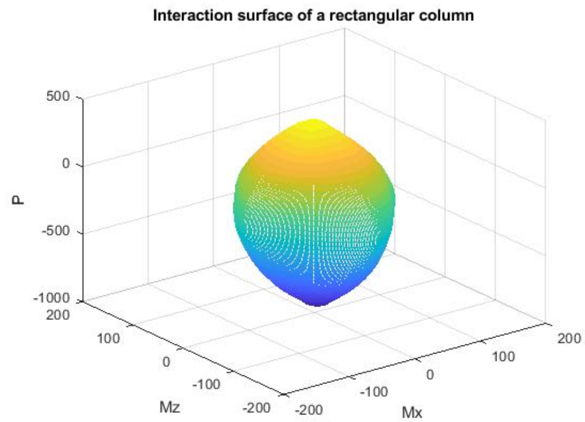


Figure 68: *Expected output of an interaction 3D surface of a rectangular column cross-section.*

5.3.44 Function: CostDimCurveOptimDesignCols

Purpose: to compute A Cost-Dimension curve for reliable optimal design options of a concrete column element subject to biaxial compression loads.

Syntax:
$[collectionDimCols, collectionISRareaCols, collectionISRcols, ...$
$collectionEffCols] = CostDimCurveOptimDesignCols(height, fc, rec, ...$
$ductility, pu\_cols, unit\_cost\_conc\_cols, load\_conditions, fy)$

System of units: SI - (Kg, cm)**:**

Description: .

Output:

- *collectionDimCols* : is the array containing the pair of dimensions (b,h) corresponding to each potential design option in the Cost-Dimension Curve
- *collectionISRareCols* : is the array containing the ISR area for each potential design option the Cost-Dimension curve
- *collectionISRcols* : is the array containing the ISR width values for each potential design option in the Cost-Dimension curve
- *collectionEffCols* : is the array containing the structural efficiency of each potential design option in the Cost-Dimension curve

Input variables:

- *height* : is the column's height
- fc : is the concrete compressive strength value Kg/cm^2
- *rec* : is the concrete cover for each cross-section direction: $cover - b, cover - h$
- fy : yield stress of the reinforcing steel $(4200Kg/cm^2)$
- *ductility* : parameter that indicates the ductility demand for the design of the column cross-section: (1),(2) or (3) for low, medium and high ductility
- *load_conditions* : is the array containing the applied biaxial load conditions. Size: *nloads* x 4 in format $[nload, Pu, Mux, Muy]$
- *unit_cost_conc_cols* : is the unit cost of the concrete per unit volume (according to the f'c value)
- *pu_cols* : is the unit construction cost of reinforcing steel assembly

Theory:

The function computes a curve of optimal designs of reinforced concrete columns, in which in the horizontal axis there is a range of cross-section b width dimensions and in the vertical axis their respective optimal construction construction costs, considering the reinforcing steel and the concrete.

The optimization process for each b width dimension is such that the optimal (or minimum) h height dimension with which the given set of load conditions would be withstood (that is, with which a structural efficiency of $Ef \leq 1.0$ could be reached). To do so, for each b width dimension an initial height dimension is the minimum one allowed by code ($30cm$) and it increases in case the structural efficiency against such set of load conditions overpasses $Ef > 1.0$. Such curve allows for a better assessment as to determine what would be the optimal cross-section dimensions to choose given certain load conditions for a column cross-section in function of the construction costs generated (see **Fig. 69**).

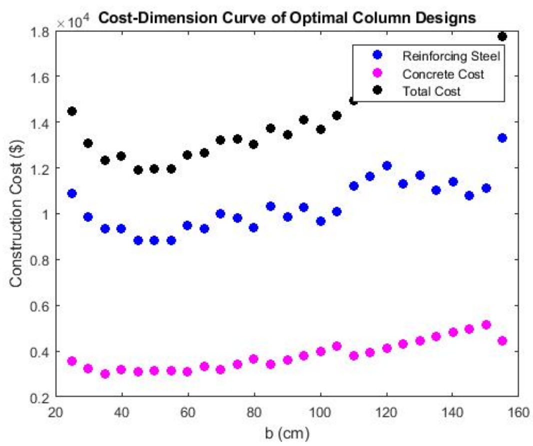


Figure 69: *Expected Cost-Dimension curve of optimal reinforced concrete column designs.*

5.3.45 Function: CostDimCurveOptimDesignBeam

Purpose: to compute A Cost-Dimension curve for reliable optimal design options of a concrete beam element subject to biaxial compression loads.

Syntax:
$[collectionDimBeams, collectionISRareaBeams, collectionISRbeams, ...$
$collectionEffBeams] = CostDimCurveOptimDesignBeam(span, fc, b_rec, ...$
$h\_rec, duct, pu\_beams\_steel, unit\_cost\_conc\_beams, ...$
$graphConvergencePlot, load\_conditions, fy)$

System of units: SI - (Kg, cm)**:**

Description: .

Output:

- *collectionDimBeams* : is the array containing the pair of dimensions (b,h) corresponding to each potential design option in the Cost-Dimension Curve
- *collectionISRareaBeams* : is the array containing the ISR area for each of the three cross-sections of each potential design option the Cost-Dimension curve
- *collectionISRbeams* : is the array containing the ISR width values for each of the three cross-sections of each potential design option in the Cost-Dimension curve
- *collectionEffBeams* : is the array containing the structural efficiencies for each of the three cross-sections of each potential design option in the Cost-Dimension curve

Input variables:

- *span* : is the beam's length
- fc : is the concrete compressive strength value Kg/cm^2
- *b_rec* : is the concrete cover in the width direction
- *h_rec* : is the concrete cover in the height direction
- fy : yield stress of the reinforcing steel $(4200Kg/cm^2)$
- *duct* : parameter that indicates the ductility demand for the design of the column cross-section: (1),(2) or (3) for low, medium and high ductility
- *load_conditions* : is the array containing the applied biaxial load conditions. Size: *nloads* x 4 in format $[nload, Pu, Mux, Muy]$
- *unit_cost_conc_beams* : is the unit cost of the concrete per unit volume (according to the f'c value)
- *pu_beams_steel* : is the unit construction cost of the reinforcing steel assembly

Theory:

The function computes a curve of optimal designs of rectangular reinforced concrete beams, in which in the horizontal axis there is a range of cross-section b width dimensions and in the vertical axis their respective optimal construction construction costs, considering the reinforcing steel and the concrete.

The optimization process for each b width dimension is such that the optimal (or minimum) h height dimension with which the given set of load conditions would be withstood (that is, with which a structural efficiency of $Ef \leq 1.0$ could be reached). To do so, for each b width dimension an initial height dimension is the minimum one allowed by code ($20cm$) and it increases in case the structural efficiency against such set of load conditions overpasses $Ef > 1.0$. Such curve allows for a better assessment as to determine what would be the optimal cross-section dimensions to choose given certain load conditions for a beam cross-section in function of the construction costs generated (see **Fig. 70**).

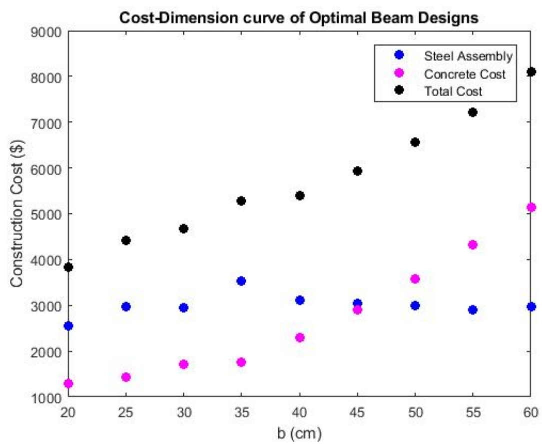


Figure 70: *Expected Cost-Dimension curve of optimal reinforced concrete beam designs.*

5.4 Rebar analysis for circular columns

5.4.1 Function: ExportResultsColumnCirc

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

Syntax:

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

System of units: Any.

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
- *bestArrangement* is the list of the rebar diameters' indices used in the element

5.4.2 Function: effCircColsLS

Purpose: To compute the structural efficiency of a circular reinforced column cross-section. The function deploys linear search to find the corresponding resistance for multiple load combinations according to their load eccentricity, from the interaction diagram's data.

Syntax:

$$[maxef, tableEff, c] = effCircColsLS(diagrama, load\_conditions, c\_vector)$$

System of units: Any.

Description: .

Output:

- $maxef$ is the critical structural efficiency of the column cross-section given different load conditions
- $tableEff$ is the resume table of results consisting of $nload\_conditions$ rows and 5 columns as:

$$[P_u, M_u, P_R, M_R, Eff]$$

- c is the neutral axis depth corresponding to the most critical load condition

Input variables:

- $diagrama$ is the interaction diagram data of size: $[npoints, 5]$ that can be obtained from function $diagCircColsISR$ (p. 74)
- $load\_conditions$ is the array containing the load conditions: size = $[nload, 3]$ in format $[nload, Pu, Mu]$
- c_vector is the array containing the neutral axis depth values for both cross-section axis directions for all interaction diagram points: size = $[npoints + 2, 1]$

Theory:

The structural efficiency Ef is computed as:

$$Ef = \frac{\sqrt{P_u^2 + M_u^2}}{\sqrt{P_R^2 + M_R^2}} \tag{188}$$

Analytic geometry is applied for the determination of the resistance corresponding to each load condition P_R, M_R as shown in **Fig. 71**):

$$M_r = \frac{P_{i+1} + (\frac{P_i - P_{i+1}}{M_{i+1} - M_i})}{\frac{P_u}{M_u} - \frac{P_{i+1} - P_i}{M_{i+1} - M_i}} \tag{189}$$

$$P_r = \frac{P_u}{M_u} M_r \tag{190}$$

Linear Search is applied to determine the points P_i, M_i and P_{i+1}, M_{i+1}.

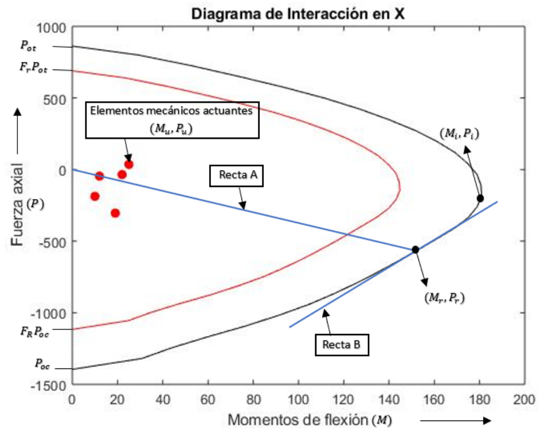


Figure 71: *Interaction diagram in the Cartesian plane as reference for comprehension of the application of analytical geometry for the computation of the structural efficiency of reinforced column cross-sections.*

5.4.3 Function: RebarDispositionCirc

Purpose: To compute the local position coordinates of rebars symmetrical over a circular column cross-section.

Syntax:

$$[dispositionRebar, separation] = RebarDispositionCirc(diam, rec, dv, nv)$$

System of units: Any.

Description:

Output variables:

- $dispositionRebar$ are the local position coordinates of the symmetrical rebar option over the circular column cross-section

Input variables:

- $diam$ cross-section diameter
- dv, nv are the rebar diameter of the rebar option and the number of rebars, respectively
- rec is the concrete cover

5.4.4 Function: optimalRebarCirc

Purpose: To determine an optimal symmetrical rebar design for a circular concrete column.

Syntax:

$[Mr\_col, Inertia, bestArea, bestCost, bestdiagram, bestnv, bestobar, ...$
$bestEf, bestType, bestDisposition, bestc] = optimalRebarCirc...$
$(diam, rec, act, Es, npdiag, fdpc, pu\_col\_circ, load\_conditions, plotRebarDesign)$

System of units:
SI - (Kg, cm)
US - (lb, in)

Description:

Output variables:

- Mr_col is the resistant bending moment of the optimal reinforced concrete column
- $Inertia$ is the momentum of inertia of the circular column cross-section
- $bestArea$ is the optimal rebar area
- $bestCost$ is the construction cost of the optimal rebar design
- $bestobar$ is the eight-of-inch of the rebar corresponding to the optimal rebar design
- $bestnv$ is the number of rebar of the optimal design
- $bestEf$ is the critical structural efficiency corresponding to the optimal rebar design against the critical load condition
- $bestType$ is the list of rebar types' index of each rebar used for the optimal design: size $[nbars, 1]$ (a list of numbers from 1 to 7, by default)
- $bestDisposition$ is an array containing the local coordinates of position of each rebar over the cross-section corresponding to the optimal rebar design

Input variables:

- rec is the concrete cover
- act is the minimum required reinforcement area (obtained from the function: $isrCircCols$ (p. 72)
- E is the Modulus of Elasticity of the reinforcement steel $E = 2.0x10^6 \frac{Kg}{cm^2}$
- $npdiag$ number of points to compute for the interaction diagram
- $load\_conditions$ load conditions for the column cross section: size = $[nload, 3]$ in format $[nload, Pu, Mu]$
- $fdpc$ is the f'_c reduced with the factor 0.85 according to the **ACI 318-19** code

- *pu_col_circ* is the database of unit cost reinforcement assembly: format by default $pu\_col = [PU_{\#4}, PU_{\#5}, PU_{\#6}, PU_{\#8}, PU_{\#9}, PU_{\#10}, PU_{\#12}]$;
- *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 linear search or exhaustive search, given the limited number of possibilities for reinforcement when only one rebar diameter is allowed for each rebar design. The restriction to accept or not a design is based entirely on the rebar separation constraint, taken as the greater of (191)

$$sep_{min} = \begin{cases} \frac{3}{2}d_b \\ \frac{4}{3}d_{ag}, d_{ag} = \frac{3}{4}in \\ 4cm \end{cases} \tag{191}$$

5.4.5 Function: diagCircColsRebar

Purpose: To compute the interaction diagram of a rebar design on a circular column cross-section and its structural efficiency given some load conditions.

Syntax:

$$[Eft, diagramaInteraccion, tablaEficiencias, c] = diagCircColsRebar(ast, ...$$
$$dispositionRebar, diam, rec, fy, npdiag, conditions, fdpc, av, E, fdpc, beta1)$$

System of units: Any.

Description:

Output variables:

- Eft Structural efficiency
- $diagramaInteraccion$ interaction diagram coordinates
- $tablaEficiencias$ is the resume table of results consisting of $nload\_conditions$ rows and five columns as $[P_u, M_u, P_R, M_R, Eff]$
- c is the neutral axis depth corresponding to the most critical load condition

Input variables:

- $diam$ is the cross-section diameter
- $npdiag$ is the number of points to be computed for the interaction diagram
- rec is the concrete cover
- E Modulus of elasticity of the reinforcing steel
- $fdpc$ 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 (195) in units Kg, cm

$$0.65 \leq (\beta_1 = 1.05 - \frac{f'_c}{1400}) \leq 0.85 \tag{192}$$

- $conditions$ is an array containing the load conditions, of size: $[nloads, 3]$ in format $[nload, Pu, Mu]$

Theory:

The max compression resistance of the column cross-section is determined as (193) 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 (194).

$$P_{oc} = 0.85 f'_c (A_c - A_s) + f_y (A_s) \tag{193}$$

$$P_{ot} = f_y(A_s) \tag{194}$$

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. 72**.

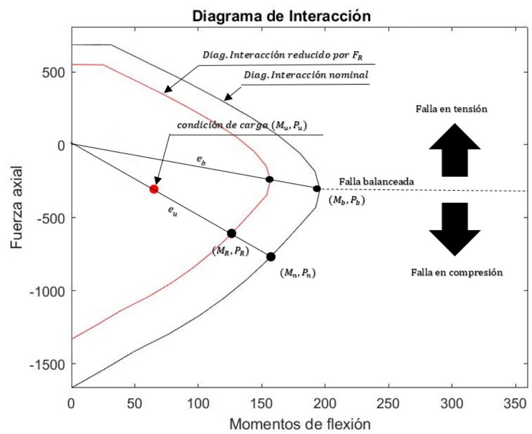


Figure 72: *Interaction diagram of reference.*

5.4.6 Function: bisectionMrRebarColsCir

Purpose: To determine the neutral axis depth, axial and bending resistance from the interaction diagram of a reinforced concrete column cross-section corresponding to a given axial force. The bisection root method is used.

Syntax:

$[raiz] = bisectionMrRebarColsCir(cUno, cDos, fr, E, diam, fdpc, beta1, ...$
$ea, av, rebarDisposition, rec)$

System of units: Any.

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
- $diam$ is the cross-section diameter
- $fdpc$ is the f'_c reduced with the factor 0.85 according to code
- $\beta1$ is determined as following (195) in units Kg, cm or as (196) in units psi

$$0.65 \leq (\beta_1 = 1.05 - \frac{f'_c}{1400}) \leq 0.85 \tag{195}$$

$$0.65 \leq (\beta_1 = 0.85 - 0.05\frac{f'_c - 4000}{1000}) \leq 0.85 \tag{196}$$

- ea is the approximation error to terminate the root bisection method
- av is the cross-section area of each rebar
- $rebarDisposition$ are the local coordinates of rebars over the cross-section

Theory:

The root bisection method is employed. For more reference of this method, see [5].

5.4.7 Function: eleMecanicosRebarColsCirc

Purpose: To determine the axial load and bending resistance of a circular reinforced column cross section reinforced symmetrically with rebars, according to a given neutral axis depth c.

Syntax:

$eleMec = eleMecanicosRebarColsCirc(c, fdpc, diam, rec, E, rebarDisposition, av, beta1)$

System of units: Any.

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 depth value
- $diam$ is the cross-section diameter
- rec is the concrete cover
- E Modulus of elasticity of the reinforcing steel, 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
- $rebarDisposition$ is the array containing the local rebar coordinates over the circular cross-section. Array size: $[nbars, 2]$ in format $[x_i, y_i]$

Theory:

The function considers the Geometric Center (GC) (which is at a depth $\frac{diam}{2}$) **Fig. 73**, so that the resistant moment is calculated as (197), where $Fs_i = As_i \cdot E_y \epsilon_i$ for reinforcement steel.

$$M_R = (\sum Fs_i + \beta_1 ab0.85f'_c)(\frac{diam}{2} - d_i) \tag{197}$$

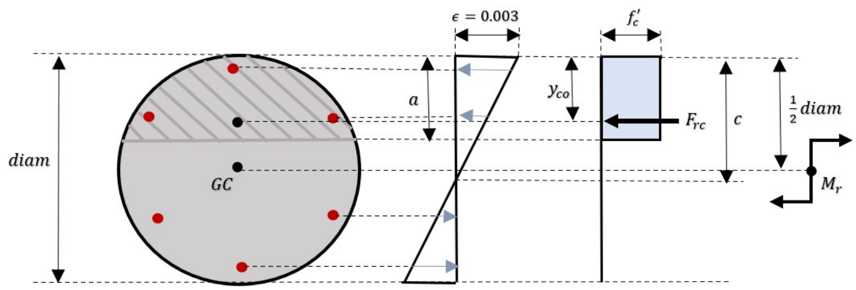


Figure 73: *Flexure-compression mechanism of a circular column cross-section.*

5.5 Rebar analysis for isolated footings

5.5.1 Function: RebarOptionsFootings

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

Syntax:

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

System of units: Any.

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.

5.5.2 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:

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

System of units: Any.

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.

5.5.3 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:

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

System of units: Any.

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. 74** and **Fig. 75**).

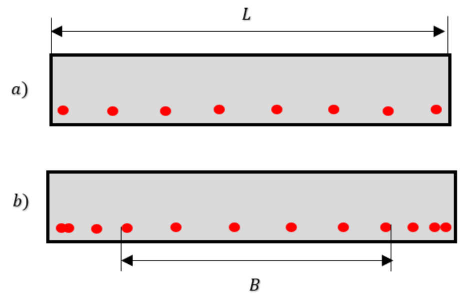


Figure 74: *Potential rebar solutions for transversal cross-sections of rectangular isolated footings.*

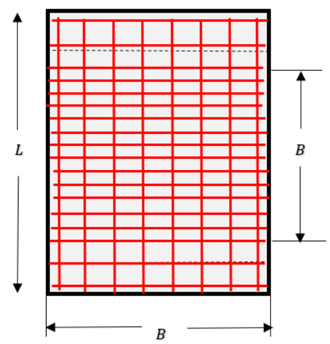


Figure 75: *Transversal reinforcement in a rectangular isolated footing.*

5.5.4 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:

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

System of units: Any.

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
- $fdpc$ is the f'_c used reduced by the 0.85 factor defined by code
- fy is the yield strength of the rebar steel
- mu_real_axis is the demanding bending moment

Theory:

The function only considers the steel rebar in tension, so that the equation (95) applies.

5.5.5 Function: EvaluateCostRebarFoot

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

Syntax:

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

System of units: Any.

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: $[noption, \#rebar, diam, area, lineal - weight]$

5.5.6 Function: EvaluateCostISRFoot

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

Syntax:

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

System of units: Any.

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)

5.5.7 Function: isrFootings

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

Syntax:

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

System of units: SI - (Kg, cm)

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
- *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 (198) and (199) respectively..

Low and Medium ductility:

$$\frac{0.7bd}{b-2rec}\frac{\sqrt{0.85f'_c}}{f_y} \leq t \leq \frac{0.9(0.85f'_c)}{(b-2rec)f_y}\frac{bd(6000\beta_1)}{(f_y+6000)} \tag{198}$$

High ductility:

$$\frac{0.7bd}{b-2rec}\frac{\sqrt{0.85f'_c}}{f_y} \leq t \leq \frac{0.75(0.85f'_c)}{(b-2rec)f_y}\frac{bd(6000\beta_1)}{(f_y+6000)} \tag{199}$$

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.

5.5.8 Function: ExportResultsIsolFootings

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

Syntax:

$ExportResultsIsolFootings(directionData, bestDispositionFootings, ...$
$dimensionFootingCollection, nbarsFootingsCollection, typesRebarFooting, ...$
$coordBaseFooting, cols\_sym\_asym\_isr)$

System of units: Any.

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, in format $[x, y, z]$ where z is the vertical axis
- $bestDispositionFootings$ 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
- $cols\_sym\_asym\_isr$ is the parameter that indicates which sort of reinforcement design was performed: either symmetrical rebar in columns, asymmetrical rebar in columns or ISR

6 Design-Analysis of 2D frames

6.0.1 Function: DesignRCPlaneFrameBCI

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:

$[totalWeightStruc, wsteelColsTotal, pacColsElem, Mp, dimensions, ...$
$unitWeightElem, wsteelConcBeamsElem, wsteelConcColsElem, ...$
$wsteelConcFootingsElem, hefootings, dimFoot, totalCostStruc, inertiaElem, ...$
$wsteelStructure] = DesignRCPlaneFrameBCI(puBeams, puCols, lenElem, fcElem, ...$
$inertiaElem, qadm, FSfootings, nodesSupportColumns, puSteelFootings, ...$
$dimensions, fcbeams, fccols, fcfootings, areaElem, colsSymAsymISR, conditionCracking, ...$
$duct, elem\_cols, elem\_beams, recxyCols, load\_conditions\_beams, load\_conditions_columns, reactions, ...$
$shearbeam, coordBaseCols, coordEndBeams, coordBaseFooting, directionData)$

System of units: SI - (Kg, cm)

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 = $[ncolumns, 1]$
- 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
- $unitWeightElem$ 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:

- *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)
- *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)
- *lenElem* is an array containing the length of each structural element composing a structural frame
- $fcElem$ 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 $[b_e, h_e]$
- $fcbeams, fccols, fcfootings$ is the f'_c used for beam, columns and isolated footings elements, respectively
- *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

- *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 (if required). Two options are possible:

  an empty vector [] if it is not required to export the design results

  the route direction to save the results, as a string

Theory:

The function designs optimally the reinforcement in all the beam elements, columns elements and isolated footing elements composing the structural frame (see **Fig. 76**). It uses 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** *beamsISR*, *isrColumnsSsymAsym* and *isrFootings* for beams, columns and isolated footings, respectively. The design results can be exported to .csv files so that they can be used by a VISUAL-CALRECOD library for the 3D visualization (either with ANSYS SpaceClaim or Dynamo).

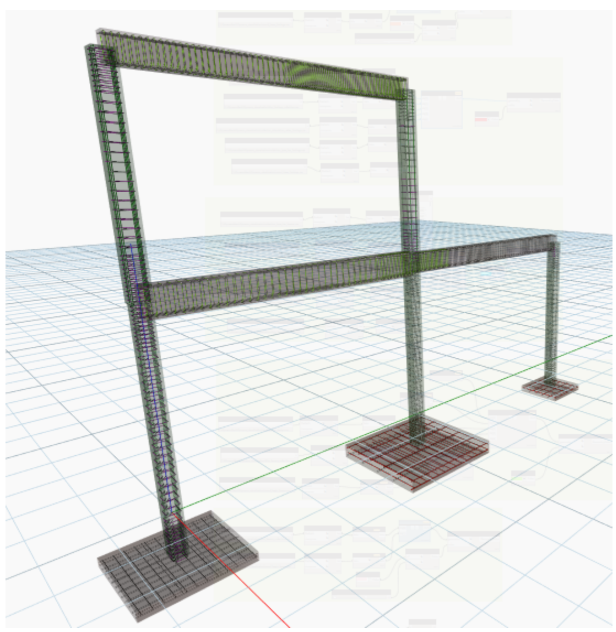


Figure 76: *3D visualization in Dynamo of an optimal design of an RC plane frame with Visual CALRECOD-Dynamo.*

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:

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

System of units: Any.

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
- $eRefN, eRefV, eRefM$; These are the structural elements of reference to scale the diagrams, respectively for axial loads (N), shear forces (V) and bending moment diagrams (M)

Theory:

For this function, the library CALFEM must be uploaded (see [2]).

7 Special structures

7.1 Retaining Walls

7.1.1 Function: RetainingRCWall

Purpose: To analyse and design the reinforcement of a reinforced concrete retaining wall for a given set of cross-sectional dimensions and soil's mechanical properties.

Syntax:

$[compliedRestric, areaWall, linearWeigthWall, tippingFS, slideFS, ...$
$LCap\_FS, sepheel, efHeel, sepfoot, effoot, septrunk, eftrunk] = RetainingRCWall...$
$(foot, heel, hf, b, FiFill, H, D, m1, m2, wvFill, beta, FiBackFill, alfa, FiFound, ...$
$fc, wvc, ductility, qadm, minFSqadm, SlideSF, TippingSF, typeRebar, sepMinRebars, ...$
$maxEf, qaf, qab, qs, LF\_DL)$

System of units: SI - (Kg, cm)

Description: .

Output variables:

- $areaWall$: is the wall's cross-sectional area
- $linearWeightWall$: is the linear weight (per cm) of the designed retaining wall
- $tippingFS$: is the final tipping Safety Factor for the designed wall
- $slideFS$: is the final slide Safety Factor for the designed wall
- $LCap\_FS$: is the final Safety Factor against the soil's bearing load capacity for the designed wall
- $sepheel$: is the final rebar separation for the designed reinforcement in the concrete wall's heel
- $efHeel$: is the final structural efficiency for the designed wall's heel
- $sepfoot$: is the final rebar separation for the designed reinforcement in the concrete wall's foot
- $effoot$: is the final structural efficiency for the designed wall's foot
- $septrunk$: is the final rebar separation for the designed reinforcement in the concrete wall's trunk
- $eftrunk$: is the final structural efficiency for the designed wall's trunk

Input variables:

- $foot$: is the cross-sectional length of the wall's foot
- $heel$: is the cross-sectional length dimension of the wall's heel
- hf: is the cross-sectional width of the wall's heel and foot
- b : is the upper cross-section wall's trunk width

- $FiFill$: is the fill soil's friction angle
- H : is the wall's stem height dimension
- D : is the soil's bakc fill depth
- $m1, m2$: are the wall's dowels' from and back slopes
- $wvFill$: is the soil fill's volumetric weight
- $beta$: is the front fill soil's upper-grade angle
- $FiBackFill$: is the back fill soil's friction angle
- $alfa$: is the back fill soil's upper-grade angle
- $FiFound$: is the foundation soil's friction angle
- fc : is the concrete compressive strength f'c (in units Kg/cm^2)
- wvc : is the concrete volumetric weight (Kg/cm^3)
- $ductility$: is the level of ductility demand for the design of the reinforcement
- $qadm$: is the soil's bearing load capacity, in units (Kg/cm^2)
- $minFSqadm$: is the design Safety Factor against the soil's bearing load capacity
- $SlideSF$: is the design Safety Factor against the slide forces over the wall
- $TippingSF$: is the design Safety Factor against the tipping forces over the wall
- $typeRebar$: is the rebar eigth-of-an-inch to be used for the wall's reinforcement
- $sepMinRebars$: is the minimum rebar separation restriction for the wall's reinforcement
- $maxEf$: is the critical structural efficiency for the design of the wall's heel, foot and trunk
- qaf, qab : is the front surcharge and the back surcharge, respectively (in units Kg/cm^2)
- qs : is the linear load that the wall may support over the top along its length
- LF_DL : is the Dead Load Design Factor

Theory:

The structural analysis is based on soil's mechanics theory. The trunk of the wall is idealized as a cantilever beam (see **Fig. 77**) as wells as the foot and the heel.

A linear distribution of stresses at the bottom of the wall's slab by the soil's contact pressures is considered. For such purpose the equation (200) is used, where e_x is the resultant load eccentricity (201)

$$q_{max-min} = \frac{P}{A} \pm \frac{6e_x}{B} \tag{200}$$

$$e_x = \frac{B}{2} - x_R \tag{201}$$

where x_R is computed as (202), being M_e and M_d the stabilizer and destabilizer moment, respectively, and N is the sum of the vertical forces.

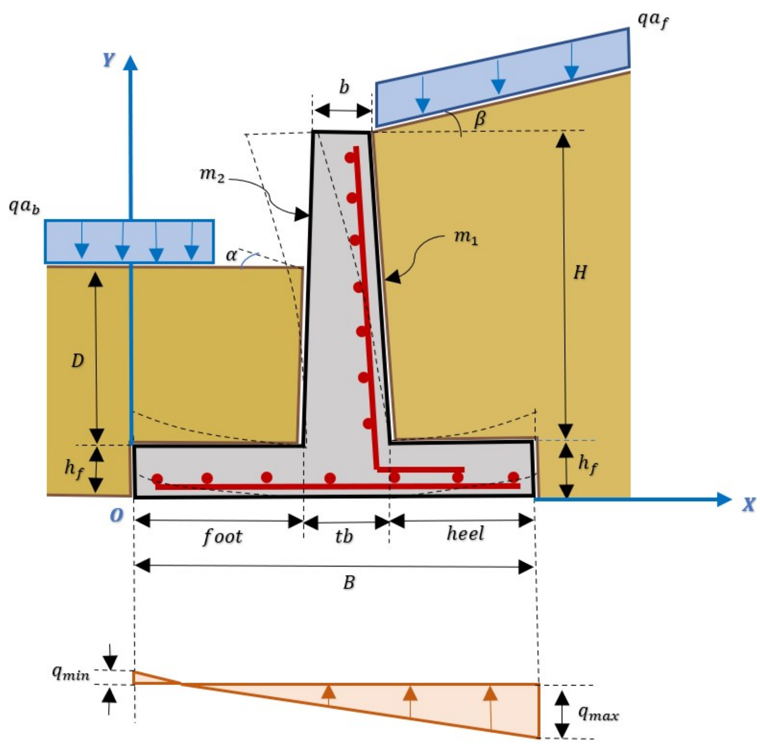


Figure 77: *Reinforced concrete retaining wall of reference.*

$$x_R = \frac{M_e - M_d}{N} \tag{202}$$

The retaining wall is revised by tipping failure (203), sliding failure (204) and failure by soil's load capacity (205).

$$F.S.(tipping) = \frac{0.7 \cdot M_e}{F_{DL} \cdot M_d} \tag{203}$$

$$F.S.(slide) = \frac{0.9 \cdot (F_r)}{F_{DL} \cdot (Ea_x + qa_f)} \tag{204}$$

$$F.S.(qadm) = \frac{q_{max}}{q_{adm}} \tag{205}$$

Finally, the mechanic elements over each of the wall's components are calculated (see **Fig. 78**) to then, design the reinforcement and compute the structural efficiencies.

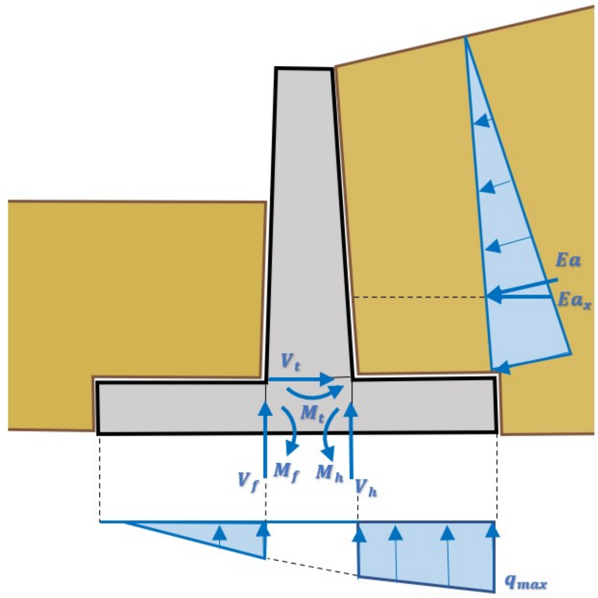


Figure 78: *Design forces and bending moments for each of the retaining wall's components.*

7.1.2 Function: DesignRetainingWallPSO

Purpose: To optimally design a reinforced concrete retaining wall with the PSO algorithm. The design variables $foot, heel, hf, b$ are taken as the optimization variables. The variables $m1$ and $m2$ (corresponding to the front and back wall's trunk slopes) remain constant throughout the process.

Syntax:

$[bestPerformance, bestPosition, besttippingFS, bestslideFS, ...$
$bestLCap_F S, bestsepheel, bestefHeel, bestsepfoot, besteffoot, bestseptrunk, ...$
$besteftrunk] = DesignRetainingWallPSO(minDim, MaxDim, H, D, m1, m2, FiFill, ...$
$wvFill, beta, FiBackFill, alfa, FiFound, fc, wvc, ductility, qadm, minFSqadm, ...$
$SlideSF, TippingSF, typeRebar, sepMinRebars, maxEf, qaf, qab, qs, LF_D L)$

System of units: SI - (Kg, cm)

Description: .

Output variables:

- $bestPerformance$: is the wall's cross-sectional area of the optimal design
- $besttippingFS$: is the final tipping Safety Factor for the optimally designed wall
- $bestslideFS$: is the final slide Safety Factor for the optimally designed wall
- $bestLCap\_FS$: is the final Safety Factor against the soil's bearing load capacity for the optimally designed wall
- $bestsepheel$: is the final rebar separation for the designed reinforcement in the optimally designed concrete wall's heel
- $bestefHeel$: is the final structural efficiency for the optimally designed wall's heel
- $bestsepfoot$: is the final rebar separation for the designed reinforcement in the optimally designed concrete wall's foot
- $besteffoot$: is the final structural efficiency for the optimally designed wall's foot
- $bestseptrunk$: is the final rebar separation for the designed reinforcement in the optimally designed concrete wall's trunk
- $besteftrunk$: is the final structural efficiency for the optimally designed wall's trunk

Input variables:

- $minDim$: is the vector containing the min dimension values that the optimization variables can have during the optimization process, in format: [foot,heel,hf,b]
- $MaxDim$: is the vector containing the max dimension values that the optimization variables can have during the optimization process, in format: $[foot, heel, hf, b]$

- $FiFill$: is the fill soil's friction angle
- H : is the wall's stem height dimension
- D : is the soil's back fill depth
- $m1, m2$: are the wall's dowels' from and back slopes
- $wvFill$: is the soil fill's volumetric weight
- $beta$: is the front fill soil's upper-grade angle
- $FiBackFill$: is the back fill soil's friction angle
- $alfa$: is the back fill soil's upper-grade angle
- $FiFound$: is the foundation soil's friction angle
- fc : is the concrete compressive strength f'c (in units Kg/cm^2)
- wvc : is the concrete volumetric weight (Kg/cm^3)
- $ductility$: is the level of ductility demand for the design of the reinforcement
- $qadm$: is the soil's bearing load capacity, in units (Kg/cm^2)
- $minFSqadm$: is the design Safety Factor against the soil's bearing load capacity
- $SlideSF$: is the design Safety Factor against the slide forces over the wall
- $TippingSF$: is the design Safety Factor against the tipping forces over the wall
- $typeRebar$: is the rebar eigth-of-an-inch to be used for the wall's reinforcement
- $sepMinRebars$: is the minimum rebar separation restriction for the wall's reinforcement
- $maxEf$: is the critical structural efficiency for the design of the wall's heel, foot and trunk
- qaf, qab : is the front surcharge and the back surcharge, respectively (in units Kg/cm^2)
- qs : is the linear load that the wall may support over the top along its length
- LF_DL : is the Dead Load Design Factor

Theory:

The function deploys the function $RetainingRCWall$ (p. 210) in the Particle Swarm Optimization algorithm (see **Algorithm 2.2**) as its Objective Function to evaluate the performance of each particle at each time step dt - being the performance the cross-sectional wall's area (which is the feature to minimize to reduce the weight of the structure and the cost) **Fig. 79**. As default, the algorithmic variables are set as $\alpha_{PSO} = 1$, $c_1 = 2$ (cognitive component), $c_2 = 2$ (social component), $dt = 0.5$ (time step), $inertia-Weight = 1.3$, $\beta_{PSO} = 0.99$, $N_{particles} = 50$, $N_{space-dim} = 4$ (number of variables), $N_{iter} = 40$.

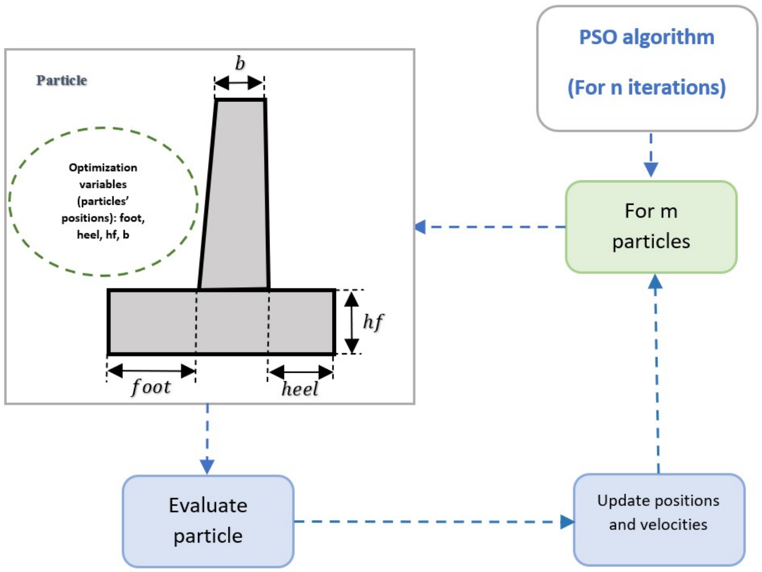


Figure 79: *Optimization design process for the reinforced concrete retaining wall with the PSO algorithm. A particle position is composed by the variables $foot, heel, hf, b$.*

8 Graphic functions

8.1 Reinforcement of beam cross-sections

8.1.1 Function: beamReinforcedSection

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

Syntax:

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

System of units: Any.

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

8.1.2 Function: MidLeftRightBeamReinforcedSections

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

Syntax:

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

System of units: Any.

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

8.1.3 Function: TbeamReinforcedSection

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

Syntax:

$TbeamReinforcedSection(bp, ht, ba, ha, dispositionRebar, barTypes1, barTypes2)$

Syntax: Any.

Description:

Input:

- ba : is the effective flange width of the T-beam cross-section
- ht : is total height of the T-beam cross-section
- bp : is the web width of the T-beam cross-section
- ha : is the flange thickness of the T-beam cross-section
- $barTypes1, barTypes2$ 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
- $dispositionRebar$ local coordinates of rebars over the T-beam cross-section

Output:

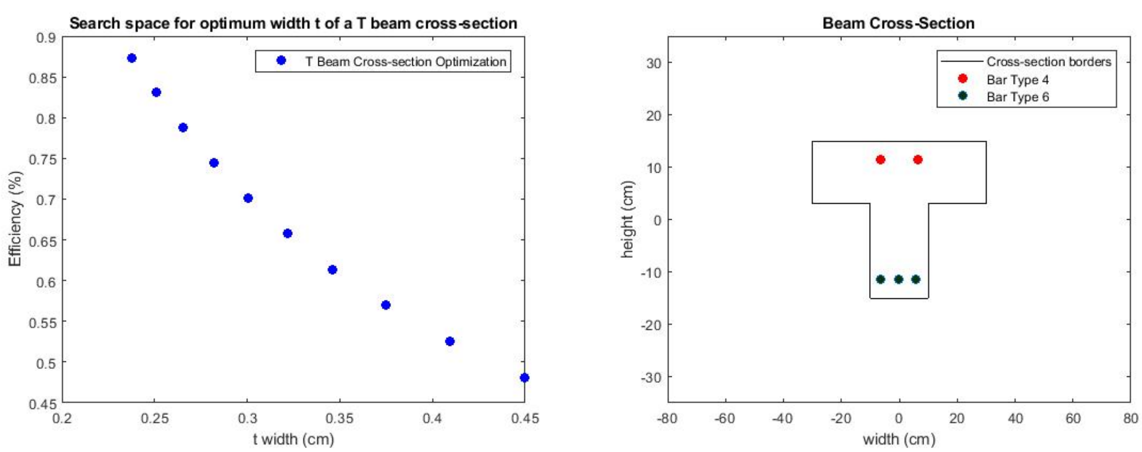


Figure 80: *Expected graphics output for this function. In the left panel it is shown the optimization convergence of the ISR's width t in tension. In the right panel it is shown the resultant optimum rebar layout for the T beam cross-section from the ISR.*

8.2 Interaction diagrams and rebar layout of column cross-sections

8.2.1 Function: diagramISR

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

Syntax:

$diagramISR(diagramInteraction, conditions)$

System of units: Any.

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. 68).

8.2.2 Function: plotdiagramCircISR

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

Syntax:

$plotdiagramCircISR(diagramInteraction, conditions)$

System of units: Any.

Description:

Input variables:

- $conditions$ load conditions in format of $n\_load\_conditions$ rows and three columns $[n\_load, P_u, Mu]$
- $diagramInteraction$ interaction diagram data computed by using the **function** $width\_efficiency$ (p. 74).

8.2.3 Function: diagramsFinalRebarCols

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

Syntax:

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

System of units: Any.

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. 127)
- *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

8.2.4 Function: plotdiagramCircRebar

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

Syntax:

$plotdiagramCircRebar(load\_conditions, diagram, dispositionRebar, diam, typeRebar)$

System of units: Any.

Description:

Input variables:

- *load_conditions* load conditions in format of *n_load_conditions* rows and three columns $[n\_load, P_u, Mu]$
- *diagram* interaction diagram data computed by using the **function** *diagramasDisposicion* (p. 190)
- *dispositionRebar* position local coordinates of a rebar design option
- *diam* cross-section diameter
- *typeRebar* is the list of rebar types for all rebars composing the rebar design: a list of size nbars consisting of a number between 1 to 7 (by default)

8.2.5 Function: PlotRotRecSecRebarCols

Purpose: To plot a rotated rectangular reinforced column cross-section as well as its interaction diagram.

Syntax:

$PlotRotRecSecRebarCols(coordenadas\_esq\_seccion, load\_conditions, ...$
$diagram, dispositionRebar, b, h, typeRebar)$

System of units: Any.

Description:

Input:

- $CoordCorners$ are the coordinates of the four corners of the rotated cross-section
- $load\_conditions$ is the given load condition. Size: 1 x 3, in format: $[1, Pu, Mu]$
- $diagram$ is the interaction diagram in the direction of the applied bending load
- b, h cross-section dimensions
- $dispositionRebar$: are the rebar local coordinates over the rotated cross-section: $[x, y]$
- $typeRebar$: list of rebar diameters' indices (according to the order in which they were given in the rebar database table)

Output:

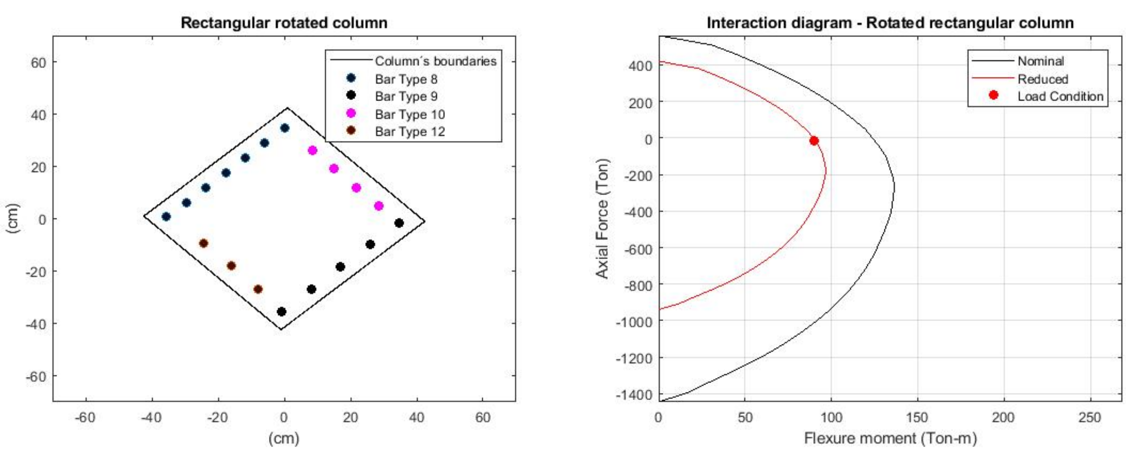


Figure 81: *Expected output. Left: rotated reinforced rectangular cross-section. Right: interaction diagram for the rotated cross-section in the direction of the applied load.*

8.2.6 Function: diagDoubleDirecAsymRebarCols

Purpose: To graph main four interaction diagrams of an rectangular column's cross-section (2 diagrams for each cross-section's axis) and the reinforced cross-section layout itself.

Syntax:

$diagDoubleDirecAsymRebarCols(load\_conditions, diagrama1, diagrama2,$
$dispositionRebar, h, b, arregloVar)$

System of units: Any.

Description:

Input variables:

- $load\_conditions$ load conditions in format of $n\_load\_conditions$ rows and four columns $[n\_load, P_u, Mu_x, Mu_y]$
- $diagrama1$ interaction diagram data with positive bending moments
- $diagrama2$ interaction diagram data with negative bending moments
- $dispositionRebar$ 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 $n\#$

8.3 Reinforcement of isolated rectangular footing cross-sections

8.3.1 Function: ReinforcedSectionsFooting

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

Syntax:

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

System of units: Any.

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

8.4 Retaining walls

8.4.1 Function: plotRCWallDesign

Purpose: To plot the cross-section of a retaining RC wall.

Syntax:

$plotRCWallDesign(H, m1, m2, toe, heel, hf, b, D, alfa, beta)$

System of units: Any.

Description:

Input variables:

- toe :: is the cross-sectional length of the wall's foot
- $heel$: is the cross-sectional length dimension of the wall's heel
- hf : is the cross-sectional width of the wall's heel and foot
- b: is the upper cross-section wall's trunk width
- H: is the wall's stem height dimension
- D: is the soil's bakc fill depth
- m1, m2: are the wall's dowels' from and back slopes
- $beta$: is the front fill soil's upper-grade angle
- $alfa$: is the back fill soil's upper-grade angle

9 Structural Frame System Functions

9.0.1 Function: WeightStruc

Purpose: To compute the weight of a reinforced concrete plane frame structure as well as all of its elements individually.

Syntax:

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

System of units: Any.

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

9.0.2 Function: CostStruc

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

Syntax:

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

System of units: Any:

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 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** $WeightStruc$

10 Visual CALREDOC

As a complement of the CALREDOC MatLab toolbox, the development of a library in Dynamo from Autodesk 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 this such Dynamo library computes the visualization of 2D Concrete Frames and its elements' longitudinal rebar and transversal rebar (for rectangular beams, rectangular and circular columns and rectangular isolated footings). It is expected that in the upcoming years and versions the library can be able to compute the visualization of 3D frames, applicable for other types of structural elements (such as slabs, continuing footings, retaining walls, etc.).

10.1 Dynamo: Beams

10.1.1 Program: RebarRectangBeams

Purpose: to compute the visualization of the concrete solid volumes of n rectangular beams, their longitudinal rebar and their transversal reinforcement.

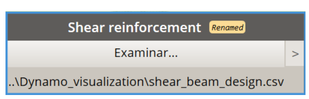
Description:

Input variables:

- *shear_beam_desgn.csv* is the csv file containing the design results of the transversal reinforcement, in format:

  $[s_1, s_2, s_3, d_1, d_2]$

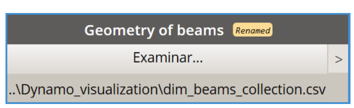
  The file is imported through a node called *"Shear reinforcement"*



- *dim_beams_collection*: is the csv file containing the geometry data of the beam elements, in format:

  $[b, h, span, concrete - cover]$

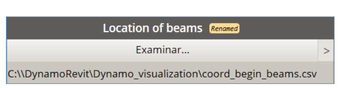
  The file is imported through a node called *"Geometry of beams":*



- *coord_begin_beams*: is the csv file containing the geometry data of the starting point of the beam elements (left end), in format:

  $[x, y, z]$

  The file is imported through a node called *"Location of beams":*

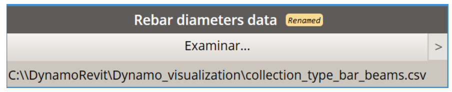


- *collection_type_bar_beams*: is the csv file containing the rebar diameter index from a given data base, in format

  $[index_{i,j}]$

  where i is the beam index and j goes from 1 to the number of rebars of each beam element (per cross-section, considering the three main cross-sections of each beam).

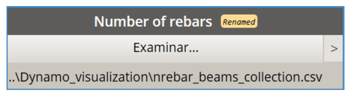
  The file is imported through a node called *"Rebar diameter data":*



- $nrebar\_beams\_collection$: is the csv file containing the number of rebars of each beam per cross-section, in format

  $[nrebars-tension-left, nrebars-compression-left, nrebars-tension-middle, nrebars-compression-middle, nrebars-tension-right, nrebars-compression-right]$

  The file is imported through a node called *"Number of rebars of each beam":*

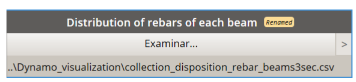


- $collection\_disposition\_rebar\_beams3sec$: is the csv file containing the number of rebars of each beam per cross-section, in format

  $[x_{i,j}, y_{i,j}]$

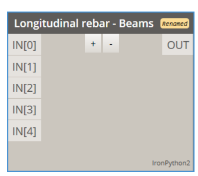
  where i is the index of the beam element and j goes from 1 to the total number of rebars in the beam element (considering each of the three main cross-section separately)

  The file is imported through a node called *"Distribution of rebars of each beam":*



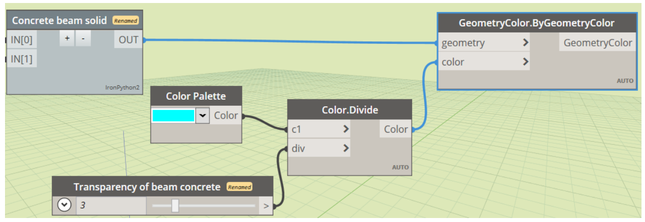
Output:

- Longitudinal rebar: visualization of the longitudinal rebar for each of the three main cross-sections of the beam element. The beam element's span is divided into three sub-elements of equal length. This task is executed through the node *"Longitudinal rebar - Beams"*



- Transversal reinforcement: visualization of the transversal reinforcement according to the design criteria from function $shearDesignBeams$

- Concrete beam solid: visualization of the rectangular concrete beam elements. There is the option to give it as much transparency as required to be able to clearly visualize also the inner reinforcement. This task is executed through the node *"Concrete beam solid":*


Concrete beam solid
Renamed
IN[0]
IN[1]
OUT
IronPython2
Color Palette
Color
Color.Divide
c1
div
Color
AUTO
GeometryColor.ByGeometryColor
geometry
color
GeometryColor
AUTO
Transparency of beam concrete
Renamed
3

10.2 Dynamo: Rectangular columns

10.2.1 Program: RebarRectangColumns

Purpose: to compute the visualization of the concrete solid volumes of n columns, their longitudinal rebar and their transversal reinforcement.

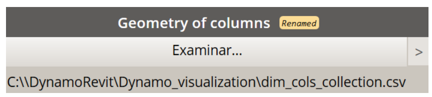
Description:

Input variables:

- *dim_cols_collection.csv* is the csv file containing the geometry data of each column to visualize, in format:

  $[b_i h_i column - length_i concrete - cover_i]$

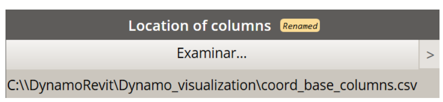
  The file is imported through a node called *"Geometry of columns"*



- *coord_base_columns.csv* is the csv file containing location of each column's base, in format:

  $[x_i, y_i, z_i]$

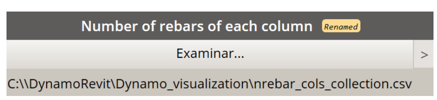
  The file is imported through a node called *"Location of columns"*



- *nbars_cols_collection.csv* is the csv file containing number of rebars of each column, in format:

  $[n - bars_i]$

  The file is imported through a node called *"Number of rebars of each column"*

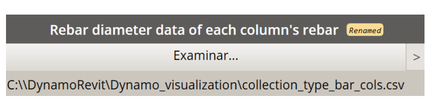


- *collection_type_bar_cols.csv* is the csv file containing rebar diamater's index of each rebar of each column, in format:

  $[index_{ij}]$

  where i is the column index and j goes from 1 to the number of rebars of each columns.

  The file is imported through a node called *"Rebar diameter data of each column's rebar"*

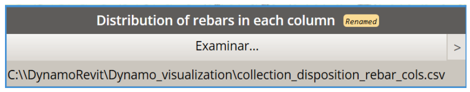


- *collection_disposition_rebar_cols.csv* is the csv file containing rebar's local coordinates over each column cross-section, in format:

  $[x_{ij}, y_{ij}]$

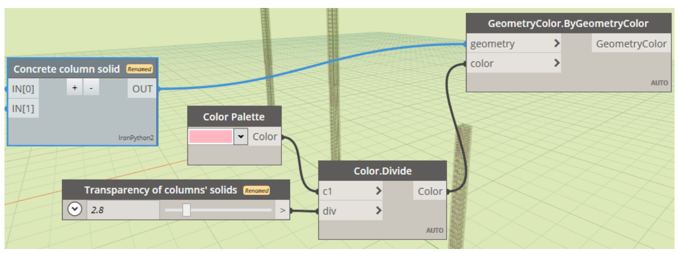
  where i is the column index and j goes from 1 to the number of rebars of each columns.

  The file is imported through a node called *"Distribution of rebars in each column"*

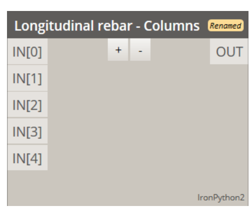


Output:

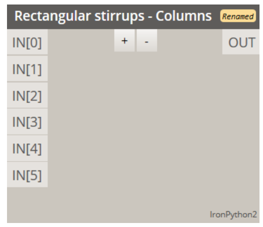
- Concrete column solid: the visualization of the rectangular concrete column. There is the option to give it as much transparency as required for the visualization of the inner reinforcement. This task is executed through the node *"Concrete column solid"*



- Longitudinal rebar: is the visualization of the longitudinal rebar. This task is executed through the node *"Longitudinal rebar - Columns"*



- Transversal reinforcement: is the visualization of the transversal reinforcement. This task is executed through the node *"Rectangular stirrups - Columns"*



Theory:

The transversal reinforcement is designed according to the **Fig. 83 (from the Mexican code *NTC-17*):**

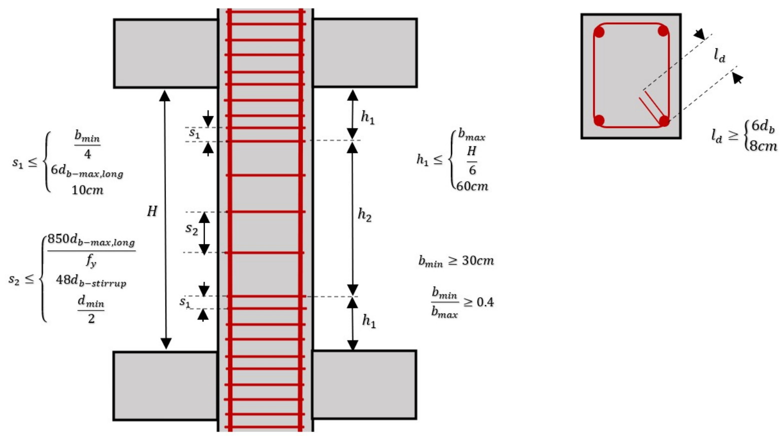


Figure 82: *Criteria for the design of transversal reinforcement in columns and its visualization.*

10.3 Dynamo: Circular columns

10.3.1 Program: RebarCircularColumns

Purpose: to compute the visualization of the concrete solid volumes of circular columns, their longitudinal rebar and their transversal reinforcement.

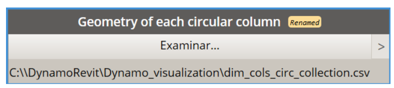
Description:

Input:

- *dim_cols_circ_collection*: is the csv file containing the geometry data of each circular column to visualize, in format:

  $[diam, length, concrete - cover]$

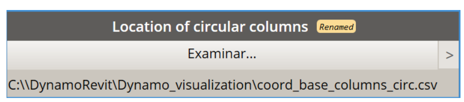
  The file is imported through a node called *"Geometry of each circular column"*



- *coord_base_columns_circ*: is the csv file containing the location coordinates of the columns' base, in format:

  $[x, y, z]$

  The file is imported through a node called *"Location of circular columns"*

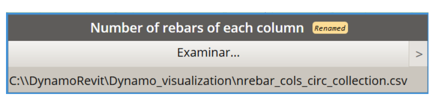


- *nrebars_cols_circ_collection*: is the csv file containing the number of rebars of each column, in format:

  $[n - rebars_i]$

  where i is the column index.

  The file is imported through a node called *"Number of rebars of each column"*

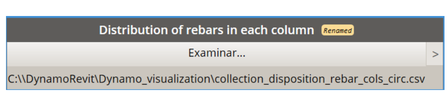


- *collection_disposition_rebar_cols_circ*: is the csv file containing the distribution of rebars over each column's cross-section, in format:

  $[x_{i,j}, y_{i,j}]$

  where i is the column index and j goes from 1 to the total number of rebars of each column.

  The file is imported through a node called *"Number of rebars of each column"*

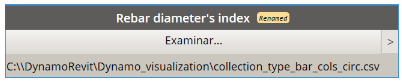


- *collection_type_bar_cols_circ*: is the csv file containing the rebar diameter's index of each rebar of each column, in format:

  $[index_{i,j}]$

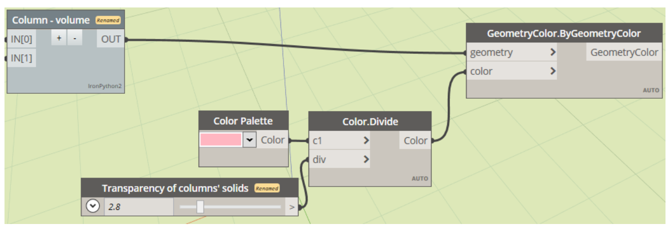
  where i is the column index and j goes from 1 to the total number of rebars of each column.

  The file is imported through a node called *"Rebar diameter's index"*

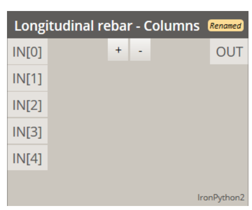


Output:

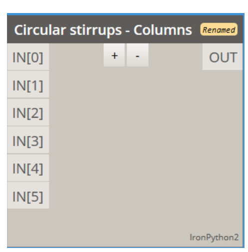
- Concrete column solid: the visualization of the circular concrete column. There is the option to give it as much transparency as required for the visualization of the inner reinforcement. This task is executed through the node *"Concrete volume"*



- Longitudinal rebar: is the visualization of the longitudinal rebar. This task is executed through the node *"Longitudinal rebar - Columns"*



- Transversal reinforcement: is the visualization of the transversal reinforcement. This task is executed through the node *"Circular stirrups - Columns"*



Theory:

The transversal reinforcement is designed according to the **Fig. 83 (from the Mexican code *NTC-17*):**

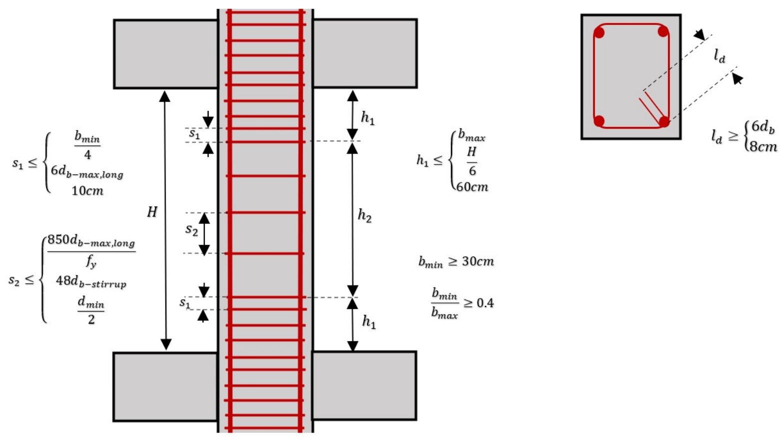


Figure 83: *Criteria for the design of transversal reinforcement in circular columns and its visualization.*

10.4 Dynamo: Rectangular isolated footings

10.4.1 Program: RebarIsolFootings

Purpose: to compute the visualization of the concrete solid volumes of rectangular isolated footings and their reinforcement.

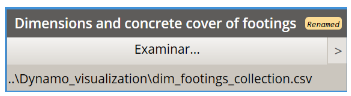
Description:

Input:

- *dim_footings_collection.csv* is the csv file containing the geometry of the designed isolated footings, in format:

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

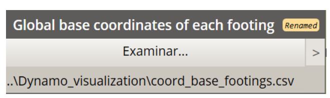
  The file is imported through a node called *"Dimensions and concrete cover of footings"*



- *coord_base_footings.csv* is the csv file containing the location coordinates of each design isolated footing (central point of the footing's base), in format:

  $[x, y, z]$

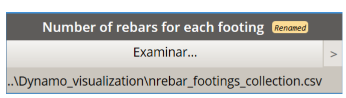
  The file is imported through a node called *"Global base coordinates of each footing"*



- *nrebar_footings_collection.csv* is the csv file containing the number of rebars for both cross-sections of each isolated footing, in format:

  $[n-rebars-B-tension, n-rebars-B-compression, n-rebars-L-tension, n-rebars-L-compression]$

  The file is imported through a node called *"Number of rebars for each footing"*

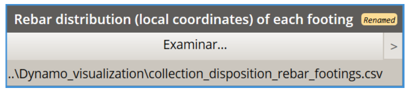


- *collection_disposition_rebar_footings.csv* is the csv file containing the local coordinates of the rebar over each of the two cross-section of each isolated footing, in format:

  $[x_{ij}, y_{ij}]$

  where i is the index of the isolated footing element and j goes from 1 to the total number of rebars for each isolated footing element.

  The file is imported through a node called *"Rebar distribution (local coordinates) for each footing"*

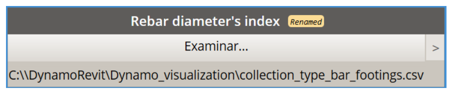


- *collection_type_bar_footings.csv* is the csv file containing the rebar diameters' index of each rebar for each isolated footing (according to the given rebar data base table), in format:

  $[index_{ij}]$

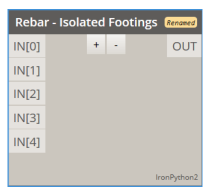
  where i is the index of the isolated footing element and j goes from 1 to the total number of rebars for each isolated footing element.

  The file is imported through a node called *"Rebar diameter's index"*

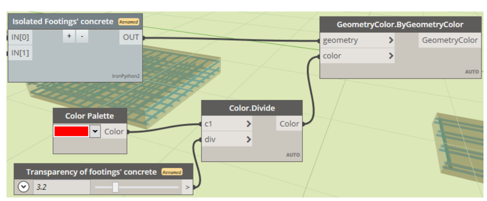


Output:

- Longitudinal rebar: visualization of the longitudinal rebar of both transversal cross-sections for each isolated footing. This task is executed through the node *"Rebar - Isolated Footings"*



- Concrete footing solid: visualization of the rectangular concrete isolated footing elements. There is the option to give it as much transparency as required to be able to clearly visualize also the inner reinforcement. This task is executed through the node *"Isolated footings' concrete":*



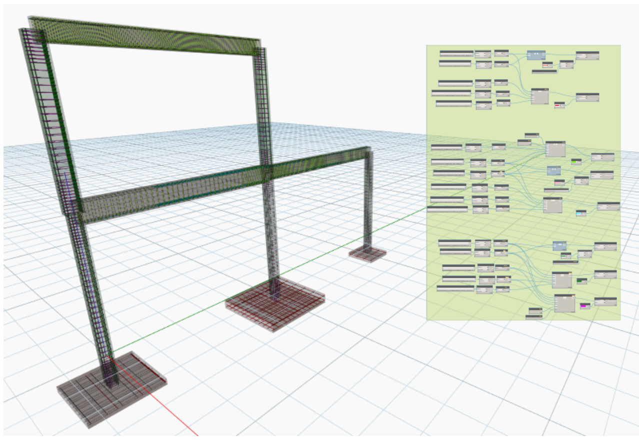
10.5 Dynamo: 2D RC frames

10.5.1 Program: RebarPlaneFramesBCIF

Purpose: to compute the visualization of reinforced concrete plane frames composed of rectangular beams, rectangular columns and rectangular isolated footings.

Description:

Output:



Theory:
The program is composed by the assembly of *RebarRectangBeams* (p. 231, *RebarRectangColumns* (p. 235) and *RebarIsolFootings* (p. 241).

11 User's Manual Examples

11.1 Ex1: Structural efficiency of a rebar reinforced T-beam cross-section

$StrucEfficiency\_TBeamsRebar\_Ex01$

Problem

It is required to assess if a rebar design of a T-beam cross-section complies with the structural efficiency Eff restriction $Eff < 1.0$ against a set of load conditions.

Solution

First off, the cross-section geometry and materials must be established, as well as the concrete cover:

```
%% Geometry

bp=20; % web width (cm)
ht=30; % total height (cm)
ba=60; % flange width (cm)
ha=12; % flange height or thickness (cm)
span=500; % cm

cover=3; % lateral concrete cover

%% Materials
fc=250; % Kg/cm2
fy=4200; % Yield stress of steel reinforcement (Kg/cm2)

fdpc=fc*0.85; % reduced f'c
beta1=0.85;
```

The load condition vector or array is given as follows, in which only the acting design bending moment is given (as beams are only design for pure flexure). Such bending moment(s) must be given in the second column of the array:

```
load_conditions=[1 700000.0]; % Kg-cm
```

Now, the available rebar types (eight-of-an-inch rebars) has to be given as an array with the following format:

```
% Database of the commercially available rebar
rebarAvailable=[3 3/8*2.54;
                4 4/8.*2.54;
                5 5/8*2.54;
                6 6/8*2.54;
                8 8/8*2.54;
                9 9/8*2.54;
                10 10/8*2.54;
                12 12/8*2.54];
```

Then, the rebar data is set, starting with the rebar local coordinates over the T-beam cross-section. For this purpose, an array of two columns is set as following, in which the first column corresponds to the x's coordinates and the second column to the y's coordinates:

```
%% Rebar
dispositionRebar=[-7 -12;
                   0 -12;
                   7 -12];
```

Given that each row of the previous *dispositionRebar* array corresponds to a rebar, another vector is necessary to indicate what eight-of-an-inch each rebar is. For this purpose the following *rebarType* vector is set, in which each of its elements are an index corresponding to a respective row of the previous given *rebarAvailable* array.

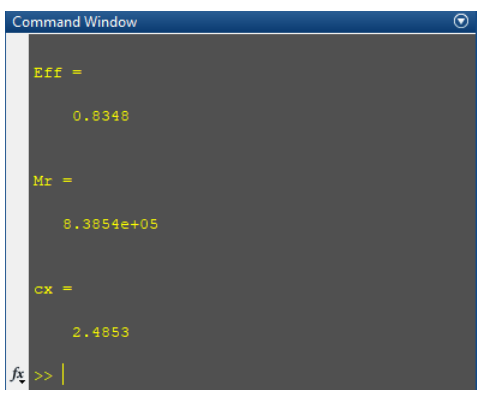
```
rebarType=[4;4;4];
```

Finally, the resistance efficiency is evaluated with the **Function** *EfRebarTBeams* as following:

```
%% Analysis of efficiency
[Eff,Mr,cx]=EfRebarTBeams(load_conditions,bp,ht,ba,ha,span,fdpc,...
                           rebarType,rebarAvailable,cover,beta1,...
                           dispositionRebar)
```

As it can be observed in the output set of the function, not only the structural efficiency can be obtained but also the resistant bending moment Mr and the corresponding neutral axis depth cx.

Results



Beware of the units. The results here presented are in the same units of the entered data.

11.2 Ex2: Structural efficiency of a rebar reinforced rectangular column through a 3D analysis with interaction surfaces

$Struc\_Eff\_3D\_ColumnRebar\_Ex01$

Problem

Due to advantages of computation time it is common to adopt the Breler's formula (Inverse Load method) and the Contour Load method instead of computing the interaction diagrams from the whole interaction surface of a column cross-section. When deploying the functions of CALRECOD in MatLab for this purpose the structure of a program will look as following.

Solution

Firstly, he cross-section geometry and materials are set, as well as some additional parameters:

```
%% Geometry
b=60; %cm
h=60; %cm

%% Materials
fc=300; %kg/cm2

if fc<280
    beta1=0.85;
elseif fc>=280
    beta1=1.05-fc/1400;
    if (beta1<0.65)
        beta1=0.65;
    elseif (beta1>0.85)
        beta1=0.85;
    end
end

E=2.1e6; % Modulus of Elasticity of reinforcing steel kg/cm2
fdpc=fc*0.85; % reduced f'c (Kg/cm2)
fy=4200; % yield stress of reinforcing steel Kg/cm2

%% Additional parameters
npdiag=40; % Number of points to compute for the interaction diagrams

concreteCover=[4 4]; % cm
```

Multiple biaxial bending-compression load conditions are then set. All of these loads will be assessed one by one so that the most critical load condition can be identified:

```
%% Loads
load_conditions=[1 -15000 -55e5 32e5;
                 2 -40000 38e5 53e5;
                 3 -31000 12e5 -5.5e5]; % [nload, Pu, Mx, My] (Ton,m)
```

As following, the rebar data is established, starting by the available commercial rebar diameters. For this purpose an array called *rebarsAvailable* is set as:

```
% Database of the commercially available rebar
rebarAvailable=[4 4/8*2.54;
                5 5/8*2.54;
                6 6/8*2.54;
                8 8/8*2.54;
                9 9/8*2.54;
                10 10/8*2.54;
                12 12/8*2.54];
```

Then, the number of rebars for each of the four cross-section's boundaries and their respective diameter index (from the *rebarAvailable*) array have to be also set. For this purpose, the following variables and vectors are created:

```
numberRebars1=5;
numberRebars2=7;
numberRebars3=4;
numberRebars4=3;

% Total number of rebars placed over the cross-section
nv=numberRebars1+numberRebars2+numberRebars3+numberRebars4;

RebarTypeIndex1=5;
RebarTypeIndex2=4;
RebarTypeIndex3=6;
RebarTypeIndex4=7;

% Combination of rebar diameters (vector containing the
% rebar diameters' indices for each of the four cross-section's boundary)
comborebar=[RebarTypeIndex1,RebarTypeIndex2,RebarTypeIndex3,RebarTypeIndex4];
```

Now, the rebar coordinates or distribution of rebars over the cross-section are computed with the **function** *dispositionRebarAsymmetric* as follows:

```
% Compute the distribution of rebars over the cross-section (local rebar
% coordinates)
[dispositionRebar,separacion_hor1,separacion_hor2,...
separacion_ver1,separacion_ver2]=dispositionRebarAsymmetric(b,...
h,concreteCover,nv,numberRebars1,numberRebars2,...
numberRebars3,numberRebars4,rebarAvailable,RebarTypeIndex1,...
RebarTypeIndex2,RebarTypeIndex3,RebarTypeIndex4);
```

Finally, the structural efficiency can be analysed. For this purpose, the **function** *multiDiagAxisColRec* is used as following:

```
%% Interaction diagram in the load directions (3D Interaction surfaces)
[tablaEff01,iloadmax,trans_load_condition,gamma,diagramIntAxis1,...
```

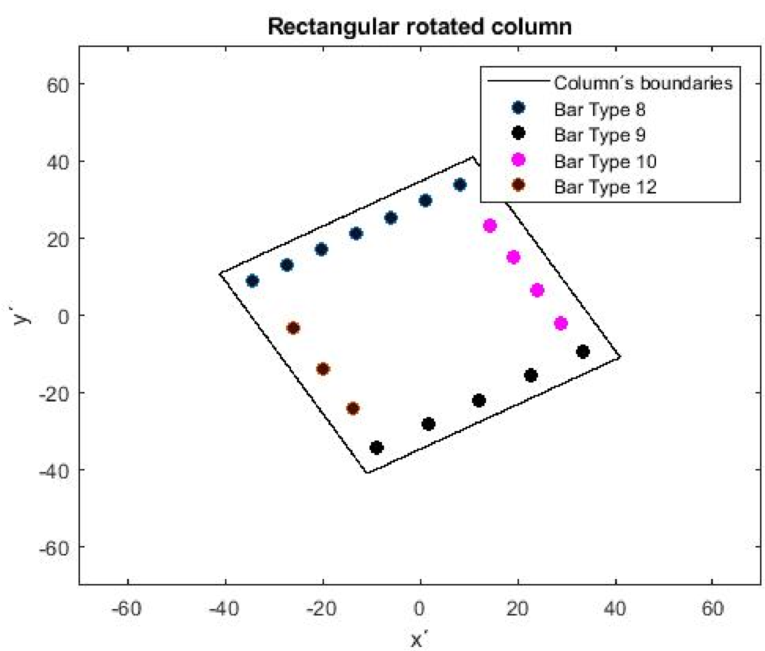
```
    newdispositionRebar,section,cmax,CP]=multiDiagAxisColRec(b,h,...
    load_conditions,comborebar,npdiag,fy,fdpc,beta1,E,numberRebars1,...
    numberRebars2,numberRebars3,numberRebars4,rebarAvailable,...
```

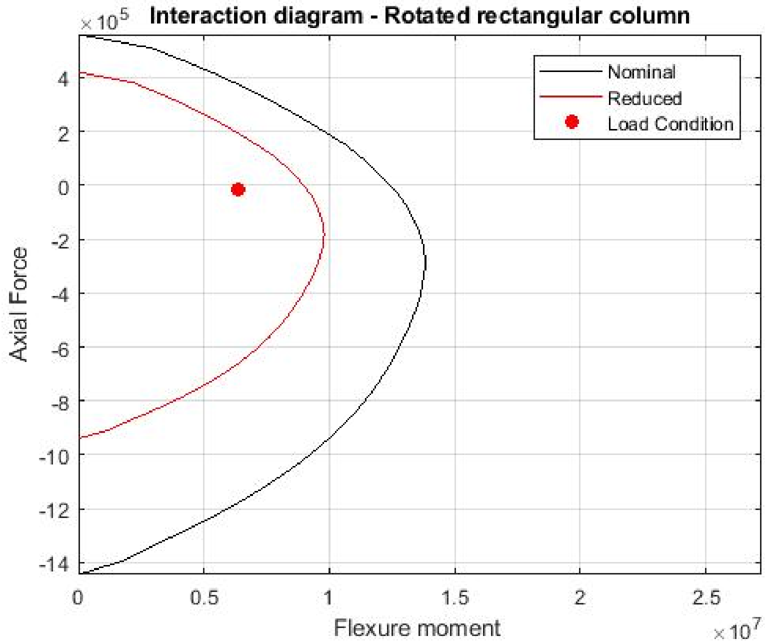
In order to visualize the results of the analysis, the **function** *PlotRotRecSecRebarCols* is used. But before using such function, a vector containing the rebar diameters' indices placed over the cross-section must be created, as following:

```
% Plotting results
rebarTypeslist(1:numberRebars1)=RebarTypeIndex1;
rebarTypeslist(numberRebars1+1:numberRebars1+numberRebars2)=RebarTypeIndex2;
rebarTypeslist(numberRebars1+numberRebars2+1:numberRebars1+numberRebars2+...
    numberRebars3)=RebarTypeIndex3;
rebarTypeslist(numberRebars1+numberRebars2+numberRebars3+1:nv)=RebarTypeIndex4;

PlotRotRecSecRebarCols(section,trans_load_condition,diagramIntAxis1,...
                    newdispositionRebar,b,h,rebarTypeslist);
```

Results

As a result, a rotated cross-section is obtained, as shown next. The angle of rotation of such cross-section corresponds to $\alpha = atan(Mux/Muy)$ of the given most critical load condition. See the functions' documentation p. 224 which for this example it is the first one from the array *load_conditions*:





Interaction diagram - Rotated rectangular column
Nominal
Reduced
Load Condition
Axial Force
Flexure moment
$\times 10^5$
$\times 10^7$
4
2
0
-2
-4
-6
-8
-10
-12
-14
0
0.5
1
1.5
2
2.5

11.3 Ex3: Comparison between a 3D-rotation analysis and the Breler's formula/Load contour method

Struc_Eff_Compare_Breler_3DRotation_Ex01

Problem

It is usually of interest to assess how much do the structural efficiency results by the Breler's formula/Load contour method and a 3D-rotation analysis vary for an asymmetrically reinforced rectangular column cross-section subject to multiple biaxial bending-compression loads. With the functions that CALRECOD offers this can be achieved very easily.

Solution

The cross-section geometry and materials are first set, as following:

```
%% Geometry
b=60; %cm
h=60; %cm

%% Materials
fc=300; %kg/cm2

if fc<280
    beta1=0.85;
elseif fc>=280
    beta1=1.05-fc/1400;
    if (beta1<0.65)
        beta1=0.65;
    elseif (beta1>0.85)
        beta1=0.85;
    end
end

E=2.1e6; % Modulus of Elasticity of reinforcing steel kg/cm2
fdpc=fc*0.85; % reduced f'c (Kg/cm2)
fy=4200; % yield stress of reinforcing steel Kg/cm2

%% Additional parameters
npdiag=40; % Number of points to compute for the interaction diagrams

concreteCover=[4 4]; % cm
```

Multiple biaxial bending-compression load conditions are then set. All of these loads will be assessed one by one (through each methodology) so that the most critical load condition can be identified:

```
%% Loads
load_conditions=[1 -15000 -25e5 22e5;
                 2 -40000 38e5 53e5;
```

```
                3 -31000 12e5 -5.5e5;
                4 -22400 -18.5e5 -41.3e5]; % [nload, Pu, Mx, My] (Ton,m)
```

As following the rebar data is established, starting by the available commercial rebar diameters. For this purpose an array called *rebarsAvailable* is set as:

```
% Database of the commercially available rebar
rebarAvailable=[4 4/8*2.54;
                5 5/8*2.54;
                6 6/8*2.54;
                8 8/8*2.54;
                9 9/8*2.54;
                10 10/8*2.54;
                12 12/8*2.54];
```

Then, the number of rebars for each of the four cross-section's boundaries and their respective diameter index (from the *rebarAvailable*) array have to be also set. For this purpose, the following variables and vectors are created:

```
numberRebars1=5;
numberRebars2=7;
numberRebars3=4;
numberRebars4=3;

% Total number of rebars placed over the cross-section
nv=numberRebars1+numberRebars2+numberRebars3+numberRebars4;

RebarTypeIndex1=5;
RebarTypeIndex2=4;
RebarTypeIndex3=6;
RebarTypeIndex4=7;

% Combination of rebar diameters (vector containing the
% rebar diameters' indices for each of the four cross-section's boundary)
comborebar=[RebarTypeIndex1,RebarTypeIndex2,RebarTypeIndex3,RebarTypeIndex4];
```

Now, the rebar coordinates or distribution of rebars over the cross-section are computed with the **function** *dispositionRebarAsymmetric* as follows:

```
% Compute the distribution of rebars over the cross-section (local rebar
% coordinates)
[dispositionRebar,separacion_hor1,separacion_hor2,...
separacion_ver1,separacion_ver2]=dispositionRebarAsymmetric(b,...
h,concreteCover,nv,numberRebars1,numberRebars2,...
numberRebars3,numberRebars4,rebarAvailable,RebarTypeIndex1,...
RebarTypeIndex2,RebarTypeIndex3,RebarTypeIndex4);
```

Finally, the structural efficiency by each numerical approach can be carried out. Let us start with the 3D rotation analysis through interaction surfaces. For this purpose, the **function** *multiDiagAxisColRec* is used as following:

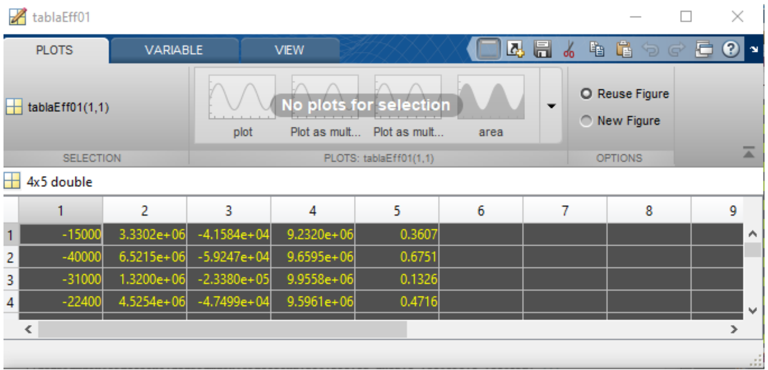
```
%% Interaction diagram in the load directions (3D Interaction surfaces)
[tablaEff01,iloadmax,trans_load_condition,gamma,diagramIntAxis1,...
    newdispositionRebar,section,cmax,CP]=multiDiagAxisColRec(b,h,...
    load_conditions,comborebar,npdiag,fy,fdpc,beta1,E,numberRebars1,...
    numberRebars2,numberRebars3,numberRebars4,rebarAvailable,...
```

Now, the Breler's formula (Inverse load method) and the Contour method are used. For this purpose the **function** *EvalAsymDoubleDirection* will be deployed to compute the interaction diagrams (positive and negative) for both cross-section axis (x and y). Then, the **function** *effRecColsDoubleDirecLS* will be used for the assessment of the structural efficiency as such:

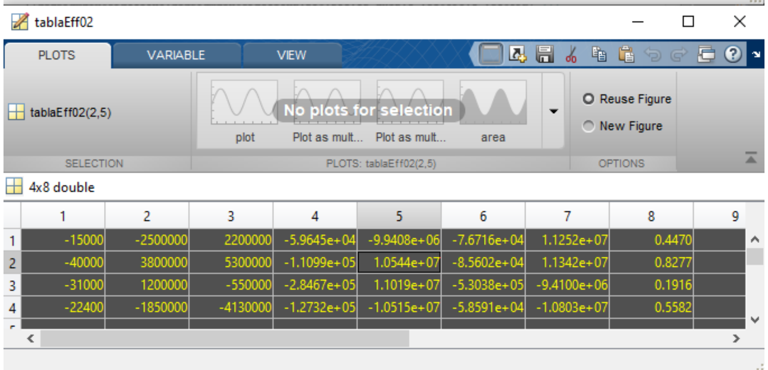
```
%% Bresler's formula (Inverse load method)
[diagramaInteraccion1,diagramaInteraccion2,pot,poc,cp_axis,c_vector1,c_vector2]=...
    EvalAsymDoubleDirection(npdiag,comborebar,b,h,...
    fy,fdpc,beta1,E,numberRebars1,numberRebars2,numberRebars3,...
    numberRebars4,rebarAvailable,dispositionRebar,concreteCover)

 % Determine the column's structural efficiency
[maxef,tablaEff02,cxy]=effRecColsDoubleDirecLS(diagramaInteraccion1,...
           diagramaInteraccion2,load_conditions,pot,poc,c_vector1,c_vector2);
```

Results

The output variables *tablaEff01* and *tablaEff02* from the functions *multiDiagAxisColRec* and *effRecColsDoubleDirecLS*,respectively, are the ones considered for comparison. The last column of these arrays will correspond to the sough structural efficiency in the range 0-1. Results are the following:



	1	2	3	4	5	6	7	8	9
1	-15000	3.3302e+06	-4.1584e+04	9.2320e+06	0.3607				
2	-40000	6.5215e+06	-5.9247e+04	9.6595e+06	0.6751				
3	-31000	1.3200e+06	-2.3380e+05	9.9558e+06	0.1326				
4	-22400	4.5254e+06	-4.7499e+04	9.5961e+06	0.4716				



	1	2	3	4	5	6	7	8	9
1	-15000	-2500000	2200000	-5.9645e+04	-9.9408e+06	-7.6716e+04	1.1252e+07	0.4470	
2	-40000	3800000	5300000	-1.1099e+05	1.0544e+07	-8.5602e+04	1.1342e+07	0.8277	
3	-31000	1200000	-550000	-2.8467e+05	1.1019e+07	-5.3038e+05	-9.4100e+06	0.1916	
4	-22400	-1850000	-4130000	-1.2732e+05	-1.0515e+07	-5.8591e+04	-1.0803e+07	0.5582	

As it can be observed, the assessment through the Breler's formula (Inverse load method) and the Contour load method are more conservative for all load conditions, although this fact could be modified by varying the power factors of the Load contour method's formula $(Mu_x/Mr_x)^{\beta_1} + (Mu_y/Mr_y)^{\beta_2}$. For the results shown, such factors were taken as 1.0 each.

11.4 Ex4: Structural efficiency of an asymmetrically reinforced column cross-section subject to compression and positive bending moments

$Efficiency\_AsymRebarDesign\_Column\_Ex\_01$

Problem

For purposes of computational costs savings, when only positive bending moments are present, it would be smart to only compute the cross-section's interaction diagrams of only one direction. For such task, the **function** *DiagramsAsymmetricRebar* is be used, as it is shown next.

Solution

Starting with the geometry, materials and additional parameters:

```
%% Geometry
b=60; %cm
h=60; %cm
concreteCover=[4 4]; % cm

%% Materials
fc=300; % Concrete compressive strength (Kg/cm2)
betac=0.85;

E=2.1e6; % Modulus of Elasticity of the reinforcing steel Kg/cm2
fdpc=fc*0.85; % (Kg/cm2)
fy=4200; % Yield stress of the reinforcing steel Kg/cm2

%% Additional parameters
npdiag=40; % number of points to be computed for the interaction diagrams
```

Now, only positive bending moment loads have to be set, as following:

```
load_conditions=[1 -40000 38e5 53e5;
                 2 -21000 24e5 34e5]; % [nload, Pu, Mx, My] (Ton-m)
```

As following the rebar data is established, starting by the available commercial rebar diameters. For this purpose an array called *rebarsAvailable* is set as:

```
% Database of the commercially available rebar
rebarAvailable=[4 4/8*2.54;
                5 5/8*2.54;
                6 6/8*2.54;
                8 8/8*2.54;
                9 9/8*2.54;
                10 10/8*2.54;
                12 12/8*2.54];
```

Then, the number of rebars for each of the four cross-section's boundaries and their respective diameter index (from the *rebarAvailable*) array have to be also set. For this purpose, the following variables and vectors are created:

```
numberRebars1=5;
numberRebars2=7;
numberRebars3=4;
numberRebars4=3;

% Total number of rebars placed over the cross-section
nv=numberRebars1+numberRebars2+numberRebars3+numberRebars4;

RebarTypeIndex1=5;
RebarTypeIndex2=4;
RebarTypeIndex3=6;
RebarTypeIndex4=7;

% Combination of rebar diameters (vector containing the
% rebar diameters' indices for each of the four cross-section's boundary)
comborebar=[RebarTypeIndex1,RebarTypeIndex2,RebarTypeIndex3,RebarTypeIndex4];
```

Now, the rebar coordinates or distribution of rebars over the cross-section are computed with the **function** *dispositionRebarAsymmetric* as follows:

```
% Compute the distribution of rebars over the cross-section (local rebar
% coordinates)
[dispositionRebar,separacion_hor1,separacion_hor2,...
separacion_ver1,separacion_ver2]=dispositionRebarAsymmetric(b,...
h,concreteCover,nv,numberRebars1,numberRebars2,...
numberRebars3,numberRebars4,rebarAvailable,RebarTypeIndex1,...
RebarTypeIndex2,RebarTypeIndex3,RebarTypeIndex4);
```

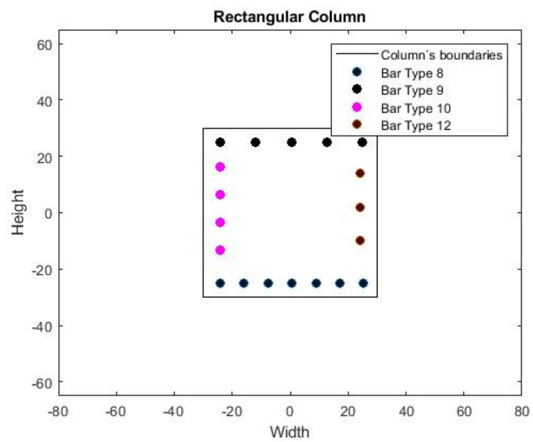
Finally, the interaction diagrams for each cross-section axis (x and y) are computed with the function *DiagramsAsymmetricR* to then, apply the Contour Load method and the Bresler's formula to determine the structural efficiency against the given load conditions:

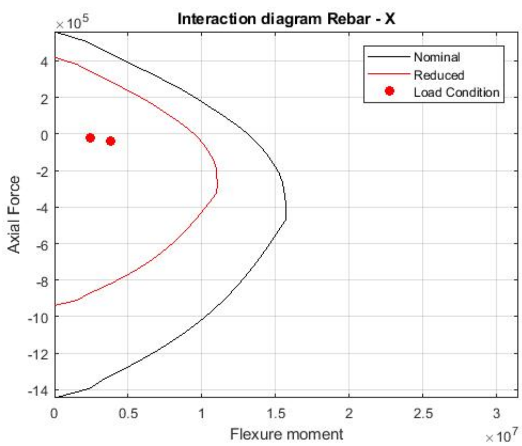
```
%% Compute the column's interaction diagram
[diagrama,cPoints,Poc,Pot]=DiagramsAsymmetricRebar...
    (npdiag,rebarcombo,b,h,fy,fdpc,betac,E,numberRebars1,...
    numberRebars2,numberRebars3,numberRebars4,rebarAvailable,...
    dispositionRebar);

%% Determine the column's structural resistance efficiency
[maxef01,eficiencia01,cxy01]=effRecColsLinearSearch...
        (diagrama,load_conditions,Pot,Poc,cPoints) ;
```

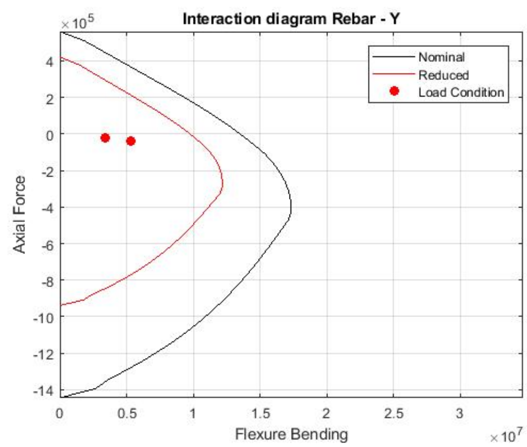
Results

In order to visualize the interaction diagrams for each cross-section axis, the function *diagramsFinalRebarCols* is used. But before using such function, a vector containing the rebar diameters' indices placed over the cross-section must be created, as following:

```
%% Plot the interaction diagram for better assessment
rebarTypeslist=zeros(nv,1);
rebarTypeslist(1:numberRebars1)=RebarTypeIndex1;
rebarTypeslist(numberRebars1+1:numberRebars1+numberRebars2)=RebarTypeIndex2;
rebarTypeslist(numberRebars1+numberRebars2+1:numberRebars1+numberRebars2+...
    numberRebars3)=RebarTypeIndex3;
rebarTypeslist(numberRebars1+numberRebars2+numberRebars3+1:nv)=RebarTypeIndex4;

diagramsFinalRebarCols(load_conditions,diagrama,dispositionRebar,...
                        h,b,rebarTypeslist);
```







11.5 Ex5: Interaction diagrams in both directions of an asymmetrically reinforced rectangular column cross-section

$Diagrams\_DoubleDirection\_Ex01$

Problem

When a rectangular column cross-section is reinforced asymmetrically with rebars, its interaction diagrams with respect to any local axis are also asymmetrical. For this reason, when computing the structural efficiency of a given set of load conditions it would be necessary to consider the sign of each bending moment, for which case the interaction diagrams in both direction with respect to any cross-section axis would be necessary to compute. Of course, this is required when the Bresler's formula and/or the Contour Load method is going to be applied for the computation of such structural efficiency.

Solution

Let us start by setting the geometry, materials and additional parameters:

```
%% Geometry
b=60; %cm
h=60; %cm

%% Materials
fc=300; % kg/cm2
betac=0.85;

E=2.1e6; % kg/cm2

fdpc=fc*0.85; % (Kg/cm2)
fy=4200; % Kg/cm2

%% Additional parameters
concreteCover=[4 4]; % cm
npdiag=40; % number of points to compute for the interaction diagrams
```

Now, only positive bending moment loads have to be set, as following:

```
%% Loads
load_conditions=[1 -15000 -25e5 22e5;
                 2 -40000 38e5 53e5;
                 3 -31000 12e5 -5.5e5]; % [nload, Pu, Mx, My] (Ton,m)
```

As following the rebar data is established, starting by the available commercial rebar diameters. For this purpose an array called *rebarsAvailable* is set as:

```
% Database of the commercially available rebar
rebarAvailable=[4 4/8*2.54;
```

```
            5 5/8*2.54;
            6 6/8*2.54;
            8 8/8*2.54;
            9 9/8*2.54;
            10 10/8*2.54;
            12 12/8*2.54];
```

Then, the number of rebars for each of the four cross-section's boundaries and their respective diameter index (from the *rebarAvailable*) array have to be also set. For this purpose, the following variables and vectors are created:

```
numberRebars1=5;
numberRebars2=2;
numberRebars3=6;
numberRebars4=1;

% Total number of rebars placed over the cross-section
nv=numberRebars1+numberRebars2+numberRebars3+numberRebars4;

RebarIndex1=5;
RebarIndex2=4;
RebarIndex3=6;
RebarIndex4=7;

rebarcombo=[RebarIndex1,RebarIndex2,RebarIndex3,RebarIndex4];
```

Now, the rebar coordinates or distribution of rebars over the cross-section are computed with the **function** *dispositionRebarAsymmetric* as follows:

```
% Distribution of rebars over the cross-section (local rebar
% coordinates)
[dispositionRebar,separacion_hor1,separacion_hor2,...
separacion_ver1,separacion_ver2]=dispositionRebarAsymmetric(b,...
h,concreteCover,nv,numberRebars1,numberRebars2,...
numberRebars3,numberRebars4,rebarAvailable,RebarIndex1,...
RebarIndex2,RebarIndex3,RebarIndex4);
```

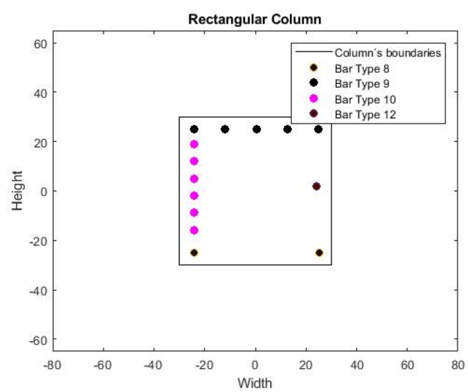
Finally, the interaction diagrams for each cross-section axis (x and y) in both directions (left and right) are computed with the function *EvalAsymDoubleDirection* to then, apply the Contour Load method and the Bresler's formula to determine the structural efficiency against the given load conditions:

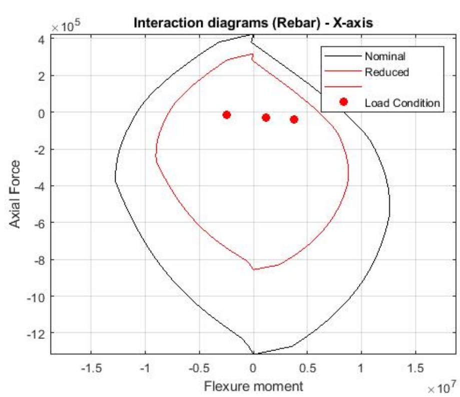
```
%% Compute the column's interaction diagram
[diagramaInteraccion1,diagramaInteraccion2,pot,poc,cp_axis,...
cvector1,cvector2]=EvalAsymDoubleDirection(npdiag,rebarcombo,b,h,...
fy,fdpc,betac,E,numberRebars1,numberRebars2,numberRebars3,...
numberRebars4,rebarAvailable,dispositionRebar,concreteCover);

%% Determine the column's structural efficiency
[maxef,tablaEficiencias,cxy]=effRecColsDoubleDirecLS(diagramaInteraccion1,...
           diagramaInteraccion2,load_conditions,pot,poc,cvector1,cvector2)
```

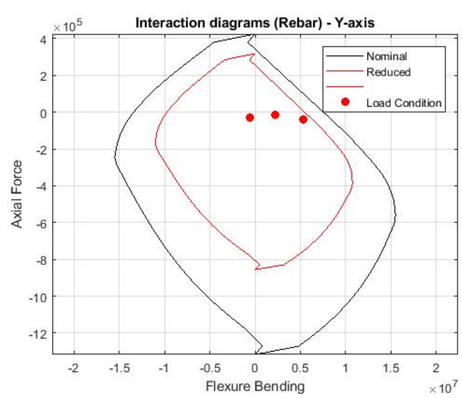
Results

In order to visualize the interaction diagrams for each cross-section axis, the function *diagDoubleDirecAsymRebarCols* is used. But before using such function, a vector containing the rebar diameters' indices placed over the cross-section must be created, as following:

```
%% Plot the interaction diagram for better assessment

% A list of the rebar diameters' index is created. Size=n-rebars x 1,
% starting with the rebars at the top of the cross-section, then at the
% bottom, then at the left side and finally at the right side
rebarTypeslist(1:numberRebars1)=RebarIndex1;
rebarTypeslist(numberRebars1+1:numberRebars1+numberRebars2)=RebarIndex2;
rebarTypeslist(numberRebars1+numberRebars2+1:numberRebars1+numberRebars2+...
    numberRebars3)=RebarIndex3;
rebarTypeslist(numberRebars1+numberRebars2+numberRebars3+1:nv)=RebarIndex4;

diagDoubleDirecAsymRebarCols(load_conditions,diagramaInteraccion1,...
    diagramaInteraccion2,dispositionRebar,h,b,rebarTypeslist);
```







11.6 Ex6: Amplification of moments for asymmetrically reinforced slender concrete columns

$StabilitySlenderRCols\_Rebar\_Ex01$

Problem

When designing slender columns, it is usually required to determine the amplification of moments due to the column's slenderness. For the computation of such amplified moments many methods are optional to apply (either numerical or empirical ones). One of the most accurate methods is the $P-\Delta$ method by considering directly Geometrical-Non Linearities. However, given that this method is iterative (and not very simple to compute) other empirical approximated methods are more often applied in industry.

On the other hand, in order to make the computation of such amplified moments more accurate, it would be also required to consider cross-section *"cracking mechanisms'*.

Solution

CALRECOD's function $MomAmpColsGeomNL$ executes the $P-\Delta$ iterative method to compute the amplified moments do to a column's slenderness by considering directly Geometrical Non-Linearities (see p. 43). Such function will be shown-case as following.

On the other hand, CALRECOD's function $CrackingColumnsAsym$ can be used to compute the modified momentum of inertia of a rectangular asymmetrically reinforced concrete column subject to a given biaxial bending-compression load condition, by considering its respective load eccentricity and its respective neutral axis depth value, among other factors, such as the amount of steel reinforcement (see p. 31).

With the purpose of show-case how such functions can be used in MatLab, let us start by setting the geometry, materials and additional parameters for a given column element as follows. As it can be observed, a vector named "$dimensionsColumn$" was set so that later on the column's dimensions could be rotated to compute the cross-section Plastic Center and the amplified moments for each cross-section axis. Note also that the slenderness factor k must be given to consider the slenderness of the column (which depends on how the columns is supported at its ends).

```
%% Geometry
b=40;
h=40;
dimensionsColumn=[b h];
height=500; % Column's length

%% Materials
fc=300;
Ec=14000*sqrt(fc); % Modulus of Elasticity of the concrete

betac=0.85;
E=2.1e6;%kg/cm2

fdpc=fc*0.85; % reduced concrete's compressive strength (Kg/cm2)
```

```
fy=4200; % Yield stress of reinforcing steel Kg/cm2

%% Additional structural parameters
k=2; % slederness factor
npdiag=50; % number of points for the computation of the interaction
           % diagram

concreteCover=[4 4]; %cm
```

Now, for the purpose of this program, only one load condition will be given to better appreciate its amplification later on. Note that lateral loads can also be considered (if wanted) for the computation of such amplification factors.

```
%% Loads
load_conditions=[1 -15000 28e5 22e5]; % [nload, Pu, Mx, My] (Ton-m)
P=load_conditions(2);
Mx=load_conditions(3);
My=load_conditions(4);
Vx=0; % Lateral loads
Vy=0;
```

As following the rebar data is established, starting by the available commercial rebar diameters. For this purpose an array called *rebarsAvailable* is set as:

```
% Database of the commercially available rebar
rebarAvailable=[4 4/8*2.54;
                5 5/8*2.54;
                6 6/8*2.54;
                8 8/8*2.54;
                9 9/8*2.54;
                10 10/8*2.54;
                12 12/8*2.54];
```

Then, the number of rebars for each of the four cross-section's boundaries and their respective diameter index (from the *rebarAvailable*) array have to be also set. For this purpose, the following variables and vectors are created:

```
% Number of rebars at each cross-section boundary
numberRebars1=2;
numberRebars2=8;
numberRebars3=0;
numberRebars4=0;

% Total number of rebars placed at each cross-section boundary
nv=numberRebars1+numberRebars2+numberRebars3+numberRebars4;

% Rebar diameter's indices (from the "rebarAvailable" array) at each
% cross-section boundary
RebarIndex1=4;
RebarIndex2=4;
RebarIndex3=4;
```

```
RebarIndex4=4;

rebarcombo=[RebarIndex1,RebarIndex2,RebarIndex3,RebarIndex4];
```

Now, the rebar coordinates or distribution of rebars over the cross-section are computed with the **function** *dispositionRebarAsymmetric* as follows:

```
% Rebar coordinates
[dispositionRebar,separacion_hor1,separacion_hor2,...
separacion_ver1,separacion_ver2]=dispositionRebarAsymmetric(b,...
h,concreteCover,nv,numberRebars1,numberRebars2,...
numberRebars3,numberRebars4,rebarAvailable,RebarIndex1,...
RebarIndex2,RebarIndex3,RebarIndex4);
```

In order to compute the cross-section Plastic Center (by considering the reinforcing steel and its distribution) the **function** *PlastiCenterAxis* takes places for each cross-section axis interaction diagrams for each cross-section axis (x and y). For such purpose a vector containing the rebar diameters' indices must be set and rotated for each axis. The rebar coordinates must be also be rotated:

```
%% Computation of the Plastic Center with respect to the X-axis
rebarTypeslist(1:numberRebars1)=RebarIndex1;
rebarTypeslist(numberRebars1+1:numberRebars1+numberRebars2)=RebarIndex2;
rebarTypeslist(numberRebars1+numberRebars2+1:numberRebars1+numberRebars2+...
    numberRebars3)=RebarIndex3;
rebarTypeslist(numberRebars1+numberRebars2+numberRebars3+1:nv)=RebarIndex4;

[PCX]=PlastiCenterAxis(fy,fdpc,b,h,dispositionRebar,rebarTypeslist,...
                            rebarAvailable)

%% Computation of the Plastic Center with respect to the Y-axis
rebarCoordinates=[dispositionRebar(:,1) dispositionRebar(:,2)];

% Invert rebar local coordinates (the cross-section is rotated 90°)
dispositionRebar(:,1)=-rebarCoordinates(:,2);
dispositionRebar(:,2)=rebarCoordinates(:,1);

% Invert cross-section dimensions
h=dimensionsColumn(1);
b=dimensionsColumn(2);

% Invert rebar diameters over the cross-section
rebarTypeslist2(1:numberRebars3)=RebarIndex3;
rebarTypeslist2(numberRebars3+1:numberRebars3+numberRebars4)=RebarIndex4;
rebarTypeslist2(numberRebars3+numberRebars4+1:numberRebars3+numberRebars4+...
    numberRebars2)=RebarIndex2;
rebarTypeslist2(numberRebars3+numberRebars4+numberRebars2+1:nv)=RebarIndex1;

[PCY]=PlastiCenterAxis(fy,fdpc,b,h,dispositionRebar,rebarTypeslist2,...
                            rebarAvailable)
```

Note in the previous code piece that in order to rotate the rebar coordinates, cross-section dimensions and rebar diameters, auxiliary vectors were set, so that the original data could be saved for later for the computation of the interaction diagrams and structural efficiency, as well as the reinforced cross-section plot, as it is following shown:

```
%% Interaction diagrams with respect to the X and Y axis
h=dimensionsColumn(2);
b=dimensionsColumn(1);

[diagrama,cPoints,Poc,Pot]=DiagramsAsymmetricRebar...
    (npdiag,rebarcombo,b,h,fy,fdpc,betac,E,numberRebars1,...
    numberRebars2,numberRebars3,numberRebars4,rebarAvailable,...
    rebarCoordinates);

%% Structural resistance efficiency according to applied load combinations
[maxef01,eficiencia01,cxy01]=effRecColsLinearSearch...
        (diagrama,load_conditions,Pot,Poc,cPoints)

%% Plot the interaction diagram for better assessment
diagramsFinalRebarCols(load_conditions,diagrama,rebarCoordinates,...
                        h,b,rebarTypeslist);
```

Now, for the computation of the modified cross-section inertia (through the **function** *CrackingColumnsAsym*) due to cracking-mechanisms the following variables and vector must be set:

```
%% Modified inertia momentum of the cross-section

% Cracking mechanisms are considered with the "transformed
% cross-section method"
eccentricityXY=[abs(Mx/P),abs(My/P)]; % loads eccentricities
conditionCrack="Cracked";
PC=[PCX,PCY];

% Total rebar area at each cross-section's boundary
ab1=rebarAvailable(RebarIndex1,2)^2*pi/4*numberRebars1;
ab2=rebarAvailable(RebarIndex2,2)^2*pi/4*numberRebars2;
ab3=rebarAvailable(RebarIndex3,2)^2*pi/4*numberRebars3;
ab4=rebarAvailable(RebarIndex4,2)^2*pi/4*numberRebars4;

t1bar=ab1/(b-2*concreteCover(1));
t2bar=ab2/(b-2*concreteCover(1));
t3bar=ab3/(h-2*concreteCover(2));
t4bar=ab4/(h-2*concreteCover(2));

InertiaXY=[1/12*b*h^3,1/12*h*b^3]; % gross cross-section's inertia

[InertiaXY,Atransfxy,elimxy]=CrackingColumnsAsym(h,b,fdpc,concreteCover,...
    eccentricityXY,t1bar,t2bar,t3bar,t4bar,P,cxy01,conditionCrack,PC);
```

Note in the previous piece of code that it must established that a *Cracked* cross-section mechanism is required, because a *Non-cracked* one is also an option. Thus, for the computation of the amplified moments, the following is done. Note that for the computation of such amplified moments the cross-section dimensions are rotated with the auxiliary vector *dimensionsColumn*:

```
%% Amplification bending moments for each axis direction
[Deltax,Mampx]=MomAmpColsGeomNL(fc,k,InertiaXY(1),height,Vx,P,...
                                Mx,b,h,0); % In the x-axis
h=dimensionsColumn(1);
b=dimensionsColumn(2);
[Deltay,Mampy]=MomAmpColsGeomNL(fc,k,InertiaXY(2),height,Vy,P,...
                                My,b,h,0); % In the y-axis

%% Amplified load bending moments
load_conditions_modif=[1 P Mampx Mampy]
```

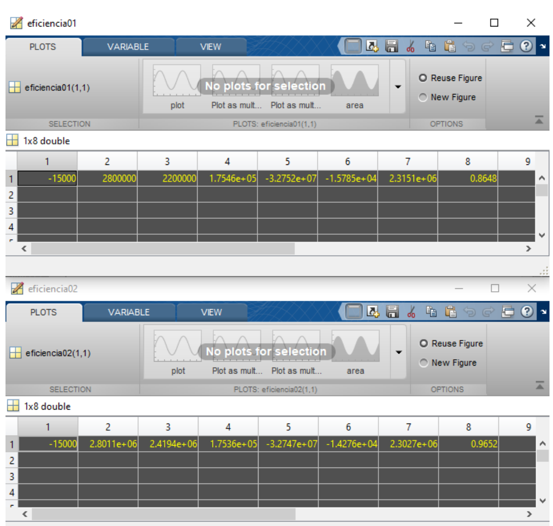
Finally, the decrement of the cross-section structural efficiency due to such amplified moments is computed as following:

```
%% Reduced resistance efficiency due to the amplified moments
% by considering geometrical non-linearities (P-Delta effects)
[maxef02,eficiencia02,cxy02]=effRecColsLinearSearch...
        (diagrama,load_conditions_modif,Pot,Poc,cPoints)

diagramsFinalRebarCols(load_conditions_modif,diagrama,rebarCoordinates,...
                        h,b,rebarTypeslist);
```

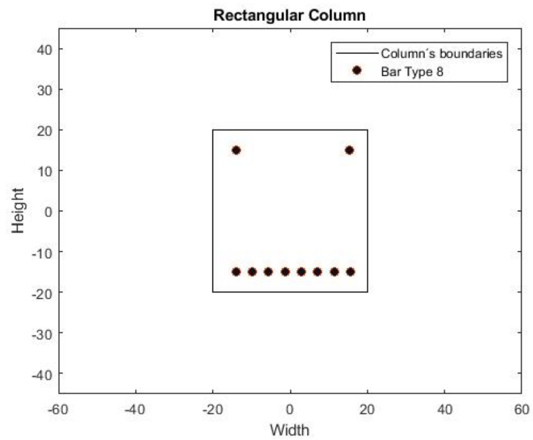
Note that the function *diagramsFinalRebarCols* is again used to better appreciate visually the amplification of the bending moments as points in the interaction diagram plots.

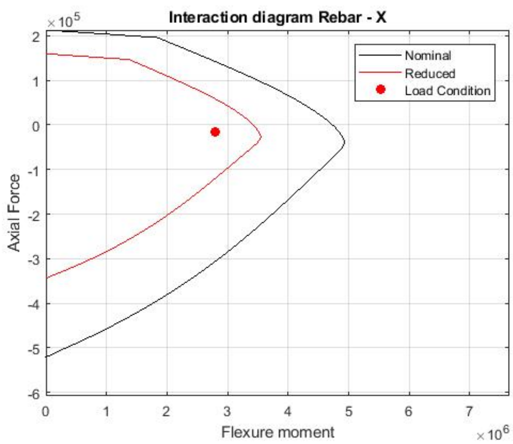
Results

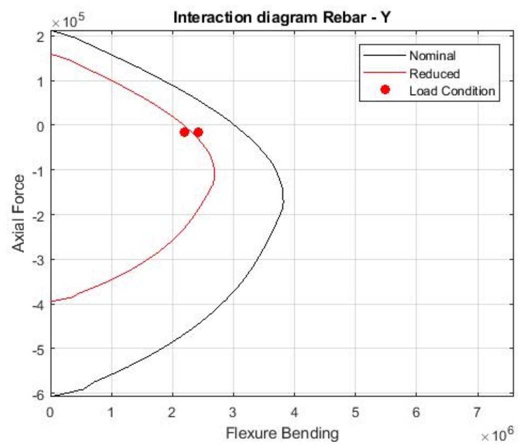
When running the program, the following structural efficiency resume of analysis are obtained, both with the original given load conditions and with the amplified moments:



As it can be observed in the previous resume tables of efficiency analysis, the lower one presents the most critical load conditions due to the amplification of the moment loads with a respective structural efficiency closer to 1. It is to note also that the greatest amplification factor is presented in the Y-axis direction of the cross-section given that the momentum of inertia with respect to that axis is severely reduced due to cracking mechanisms (as the contribution of the reinforcing steel is not that significant as it is for the X-axis direction) as it is shown in the following plots:







11.7 Ex7: Amplification of moments for symmetrically reinforced slender concrete columns

$StabilitySlenderRCols\_Rebar\_Ex02$

Problem

When designing slender columns, it is usually required to determine the amplification of moments due to the column's slenderness. For the computation of such amplified moments many methods are optional to apply (either numerical or empirical ones). One of the most accurate methods is the $P - \Delta$ method by considering directly Geometrical-Non Linearities. However, given that this method is iterative (and not very simple to compute) other empirical approximated methods are more often applied in industry.

On the other hand, in order to make the computation of such amplified moments more accurate, it would be also required to consider cross-section *"cracking mechanisms'*.

Solution

CALRECOD's function $MomAmpColsGeomNL$ executes the $P - \Delta$ iterative method to compute the amplified moments do to a column's slenderness by considering directly Geometrical Non-Linearities (see p. 43). Such function will be shown-case as following.

On the other hand, CALRECOD's function $CrackingColumnsSym$ can be used to compute the modified momentum of inertia of a rectangular symmetrically reinforced concrete column subject to a given biaxial bending-compression load condition, by considering its respective load eccentricities and its respective neutral axis depth value, among other factors, such as the amount of steel reinforcement (see p. 28).

With the purpose of show-case how such functions can be used in MatLab, let us start by setting the geometry, materials and additional parameters for a given column element as follows. As it can be observed, a vector named "*dimensionsColumn*" was set so that later on the column's dimensions could be rotated to compute the amplified moments for each cross-section axis. Note also that the slenderness factor k must be given to consider the slenderness of the column (which depends on how the columns is supported at its ends). Contrary to the previous example for asymmetrically reinforced cross-section in this case it is not necessary to compute the Plastic Center of the cross-section given that it would be in the same spot as the Geometrical Center.

```
%% Geometry
b=40;
h=40;
dimensionsColumn=[b h];
height=500; % column's length

%% Materials
fc=300; % concrete's compressive strength
Ec=14000*sqrt(fc); % Modulus of elasticity of the concrete

fdpc=fc*0.85;
betac=0.85;
```

```
fy=4200; % yield stress of the reinforcing steel (Kg/cm2)
E=2.1e6; % Modulus of elasticity of the reinforcing steel (kg/cm2)

%% Additional structural parameters
k=2; % slenderness factor
npdiag=50; % number of points to be computed for the int. diagrams
concreteCover=[4 4]; % cm
```

Now, for the purpose of this program, only one load condition will be given to better appreciate its amplification later on. Note that lateral loads can also be considered (if wanted) for the computation of such amplification factors.

```
%% Loads
load_conditions=[1 -15000 28e5 22e5]; % [nload, Pu, Mx, My] (Ton-m)
P=load_conditions(2);
Mx=load_conditions(3);
My=load_conditions(4);
Vx=0; % Lateral loads
Vy=0;
```

As following the rebar data is established, starting by the available commercial rebar diameters. For this purpose an array called *rebarsAvailable* is set as:

```
% Database of the commercially available rebar
rebarAvailable=[4 4/8*2.54;
                5 5/8*2.54;
                6 6/8*2.54;
                8 8/8*2.54;
                9 9/8*2.54;
                10 10/8*2.54;
                12 12/8*2.54];
```

Then, the number of rebars for each of the four cross-section's boundaries and their respective diameter index (from the *rebarAvailable*) array have to be also set. For this purpose, the following variables and vectors are created:

```
% Distribution of rebar
numberRebars_hdimension=6;
numberRebars_bdimension=4;
nv=2*numberRebars_hdimension+2*numberRebars_bdimension;

% Rebar diameter and area
RebarTypeIndex=4;

ov=rebarAvailable(RebarTypeIndex,1);
dv=rebarAvailable(RebarTypeIndex,2); % cm
av=pi/4*dv^2;

% Total rebar area over the cross-section
As=nv*av;
```

```
% Rebar area horizontally and vertically
ab1=numberRebars_bdimension*rebarAvailable(RebarTypeIndex,2)^2*pi/4;
ab2=numberRebars_hdimension*rebarAvailable(RebarTypeIndex,2)^2*pi/4;
```

Now, the rebar coordinates or distribution of rebars over the cross-section are computed with the **function** *RebarDisposition* as follows:

```
% Rebar coordinates over the cross-section
[dispositionRebar]=RebarDisposition(b,h,concreteCover,dv,nv,...
    numberRebars_hdimension,numberRebars_bdimension);
```

Thus, the computation of the interaction diagrams and structural efficiency, as well as the reinforced cross-section plot without considering the amplification of moments is then carried out as:

```
%% Compute the column's interaction diagram
[diagrama,cPoints,Poc,Pot]=diagramRColumnSymRebar(As,b,h,E,npdiag,...
                       fdpc,nv,betac,ov,av,dispositionRebar);

%% Column's structural resistance efficiency
% according to the initially applied load combination
[maxef01,eficiencia01,cxy01]=effRecColsLinearSearch...
        (diagrama,load_conditions,Pot,Poc,cPoints)

bestArrangement=zeros(nv,1)+RebarTypeIndex;

%% Plot the interaction diagram for better assessment
diagramsFinalRebarCols(load_conditions,diagrama,dispositionRebar,...
                        h,b,bestArrangement);
```

Now, for the computation of the modified cross-section inertia (through the **function** *CrackingColumnsSym*) due to cracking-mechanisms the following variables and vector must be set:

```
%% Modified inertia momentum of the cross-section
% considering mechanism with the transformed cross-section method
eccentricityXY=[abs(Mx/P),abs(My/P)]; % cm
conditionCrack="Cracked";

t1bar=ab1/(b-2*concreteCover(1));
t2bar=ab2/(b-2*concreteCover(1));
InertiaXY=[1/12*b*h^2,1/12*h*b^3];
[InertiaXY,Atransf_xy,elimxy]=CrackingColumnsSym(h,b,fdpc,concreteCover,...
          t1bar,eccentricityXY,t2bar,P,cxy01,conditionCrack,E);
```

Note in the previous piece of code that it must established that a *Cracked* cross-section mechanism is required, because a *Non-cracked* one is also an option. Thus, for the computation of the amplified moments, the following is done. Note that for the computation of such amplified moments the cross-section dimensions are rotated with the auxiliary vector *dimensionsColumn*:

```
%% Amplification bending moment for each axis direction
[Deltax,Mampx]=MomAmpColsGeomNL(fc,k,InertiaXY(1),height,Vx,P,Mx,b,h,0); % In the x-axis
h=dimensionsColumn(1);
b=dimensionsColumn(2);
[Deltay,Mampy]=MomAmpColsGeomNL(fc,k,InertiaXY(2),height,Vy,P,My,b,h,0); % In the y-axis

%% Amplified load bending moments
load_conditions_modif=[1 P Mampx Mampy];
```

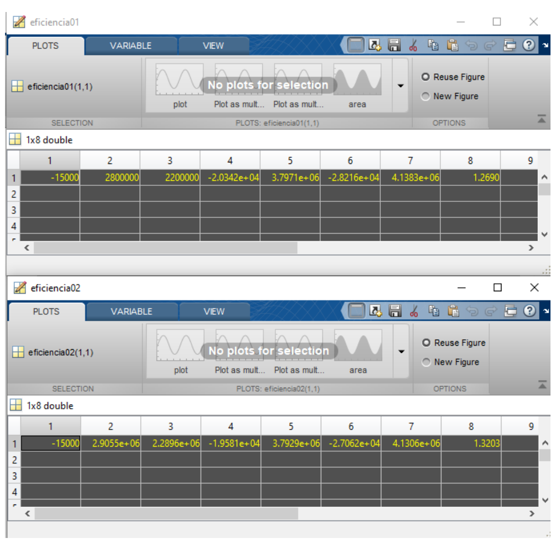
Finally, the decrement of the cross-section structural efficiency due to such amplified moments is computed as following:

```
%% Reduction in resistance due to the amplified moment by
% geometrical non-linearities (P-Delta effects)
[maxef02,eficiencia02,cxy02]=effRecColsLinearSearch...
        (diagrama,load_conditions_modif,Pot,Poc,cPoints)

%% Plot the interaction diagram for better assessment of amplified moments
diagramsFinalRebarCols(load_conditions,diagrama,dispositionRebar,...
                          h,b,bestArrangement);
```

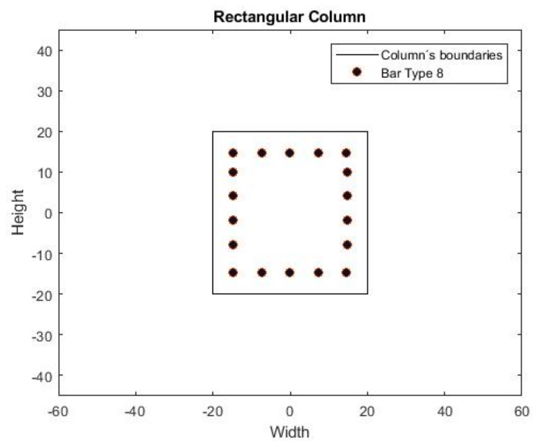
Note that the function *diagramsFinalRebarCols* is again used to better appreciate visually the amplification of the bending moments as points in the interaction diagram plots.

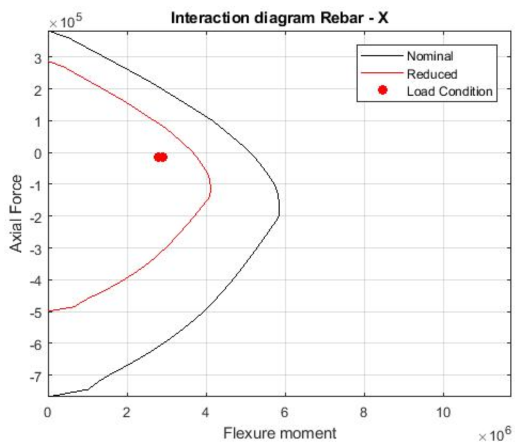
Results

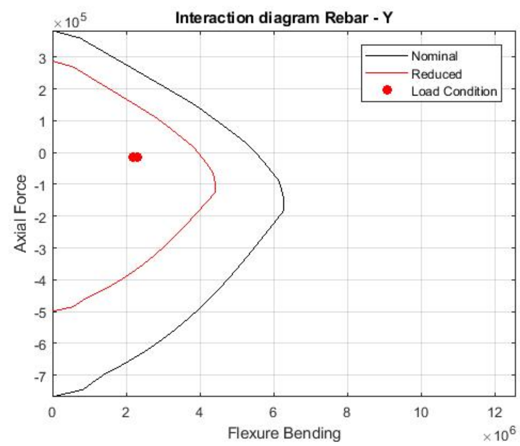
When running the program, the following structural efficiency resume of analysis are obtained, both with the original given load conditions and with the amplified moments:



As it can be observed in the previous resume tables of efficiency analysis, the lower one presents the most critical load conditions due to the amplification of the moment loads. It is to note also that the greatest amplification factor is presented in the X-axis direction of the cross-section given that the momentum of inertia with respect to that axis is the most reduced one due to cracking mechanisms (as the contribution of the reinforcing steel is not that significant as it is for the Y-axis direction) as it is shown in the following plots:







11.8 Ex8: Interaction Surfaces of rectangular reinforced concrete column cross-sections

$Interaction\_Surface\_ColumnRebar\_Ex01$

Problem

The visualization of a concrete column can provide a great overall insight about the structural efficiency of the element, however, they are not exactly that easy to compute by oneself and it is often the case that designers and academics rely on those provided by commercial software without being capable to have a more profound control over such numerical computation.

Solution

With the **function** *InteracSurfaceColRec* that CALRECOD offers the computation and manipulation of such 3D interaction surface can be easily performed.

To show-case how this function can be used in MatLab let us consider an asymmetrically reinforced rectangular cross-section. For such purpose let us start by defining the geometry of the column, its materials and other additional parameters, as follows:

```
%% Geometry
b=60; % cross-section width (cm)
h=60; % cross-section height (cm)

% Materials
fc=300; % concrete's compressive strength kg/cm2

E=2.1e6; % Modulus of Elasticity of the reinforcing steel (kg/cm2)

fdpc=fc*0.85; % reduces f'c (Kg/cm2)
fy=4200; % Yield stress of the reinforcing steel (Kg/cm2)
if fc<280
    beta1=0.85;
elseif fc>=280
    beta1=1.05-fc/1400;
    if (beta1<0.65)
        beta1=0.65;
    elseif (beta1>0.85)
        beta1=0.85;
    end
end

%% Additional parameters
npdiag=40;
concreteCover=[4 4]; %cm
```

As following the rebar data is established, starting by the available commercial rebar diameters. For this purpose an array called *rebarsAvailable* is set as:

```
% Database of the commercially available rebar
rebarAvailable=[4 4/8*2.54;
                5 5/8*2.54;
                6 6/8*2.54;
                8 8/8*2.54;
                9 9/8*2.54;
                10 10/8*2.54;
                12 12/8*2.54];
```

Then, the number of rebars for each of the four cross-section's boundaries and their respective diameter index (from the *rebarAvailable*) array have to be also set. For this purpose, the following variables and vectors are created:

```
% Number of rebars at of the four cross-section's boundaries
numberRebars1=5;
numberRebars2=7;
numberRebars3=4;
numberRebars4=3;

% Calculation of rebar area at each cross-section boundary
nv=numberRebars1+numberRebars2+numberRebars3+numberRebars4;

% Rebar diameter indices at each of the four cross-section's boundaries
RebarIndex1=5;
RebarIndex2=4;
RebarIndex3=6;
RebarIndex4=7;

% eight-of-an-inch (or rebar diameter) rebar at each cross-section boundary
comborebar=[RebarIndex1,RebarIndex2,RebarIndex3,RebarIndex4];
```

Now, the rebar coordinates or distribution of rebars over the cross-section are computed with the **function** *dispositionRebarAsymmetric* as follows:

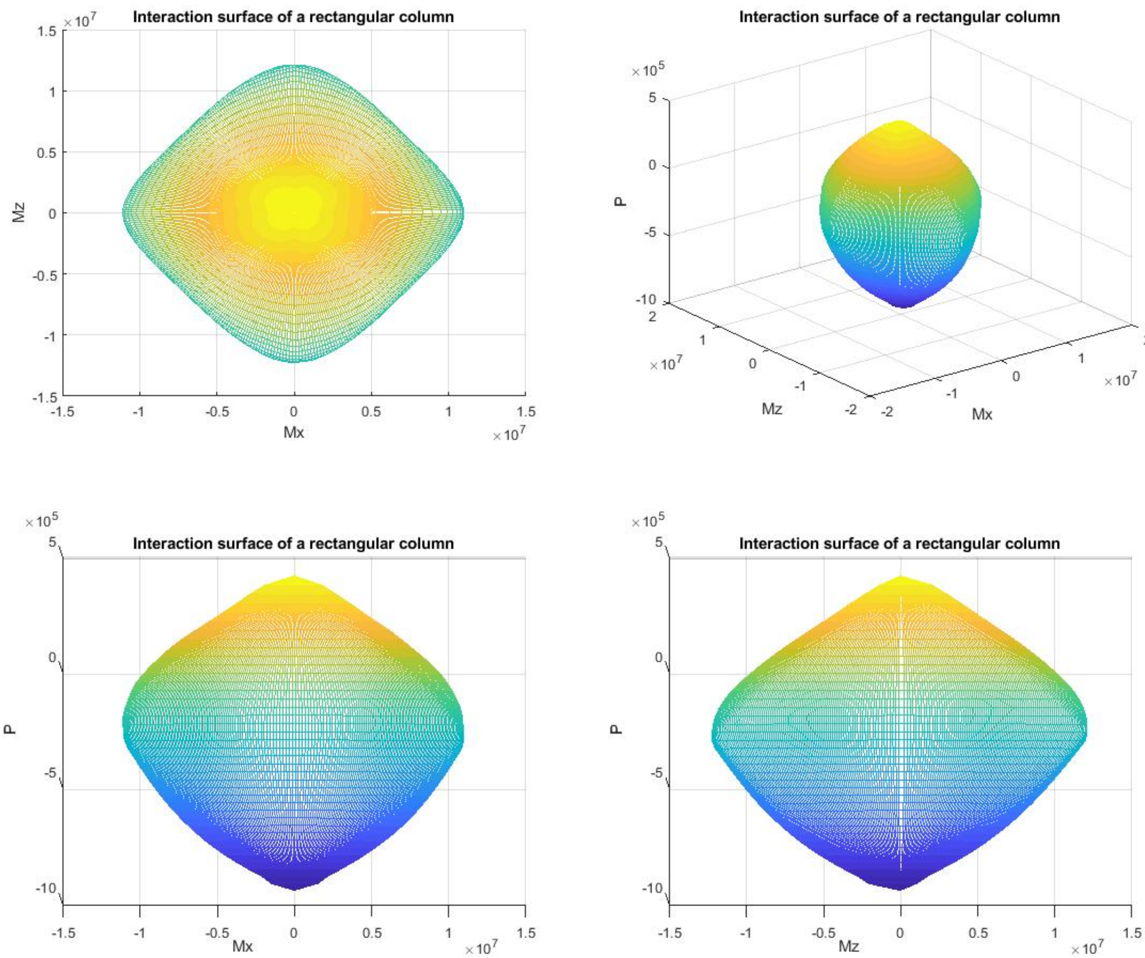
```
% Compute the distribution of rebars over the cross-section (local rebar
% coordinates)
[dispositionRebar,separacion_hor1,separacion_hor2,...
separacion_ver1,separacion_ver2]=dispositionRebarAsymmetric(b,...
h,concreteCover,nv,numberRebars1,numberRebars2,...
numberRebars3,numberRebars4,rebarAvailable,RebarIndex1,...
RebarIndex2,RebarIndex3,RebarIndex4);
```

Finally, the interaction surface can then be computed by:

```
%% Interaction surface
[supX,supY,supZ]=InteracSurfaceColRec(b,h,comborebar,npdiag,fy,fdpc,...
beta1,E,numberRebars1,numberRebars2,numberRebars3,numberRebars4,...
rebarAvailable,dispositionRebar);
```

Results

Thanks to the graphical functions that MatLab offers for the manipulation of 3D plots, the following views can be set once the 3D interaction surface is popped out:



11.9 Ex9: Structural efficiency of an symmetrically reinforced column cross-section subject to biaxial bending-compression

$EfficiencyRebarColumnDesign\_Ex01$

Problem

When dealing with symmetrical rebar designs over concrete columns only one interaction diagrams with respect to each cross-section axis would be necessary (positive moments) given that the bending loads could be considered in absolute value. For such cases, the computation of rotated interaction diagrams (or interaction surfaces) could be saved by deploying in this sense the Bresler's formula and the Contour Load method.

Solution

For these such cases the **function** *diagramasDisposicion* that CALRECOD offers is a great option, as it will be shown-case next.

Let us first begin by entering the geometry, materials and additional parameters of the concrete column:

```
%% Geometry
b=30; % cross-section width (cm)
h=25; % cross-section height (cm)

%% Materials
E=2.1e6; % Modulus of Elasticity of the reinforcing steel (kg/cm2)
fc=250; % Concrete's compressive strength
fdpc=fc*0.85; % reduced f'c
betac=0.85;

%% Additional parameters
npdiag=30; % Number of points to be computed for the interaction diagram
concreteCover=[3 3]; %cm
```

Now, only positive bending moment loads have to be set, as following:

```
%% Loads
load_conditions=[1 -9.95e3 5.26e5 0.97e5]; % [nload, Pu, Mx, My] (Ton-m)
```

Next, the rebar data is established, starting by the available commercial rebar diameters. For this purpose an array called *rebarsAvailable* is set as:

```
% Database of the commercially available rebar
rebarAvailable=[4 4/8*2.54;
                5 5/8*2.54;
                6 6/8*2.54;
                8 8/8*2.54;
                9 9/8*2.54;
                10 10/8*2.54;
                12 12/8*2.54];
```

Then, the number of rebars to be placed vertically and horizontally over each of cross-section's boundary must be set.

```
% Number of rebars placed vertically and horizontally
numberRebars_hdimension=0; % number of rebars placed vertically (per side)
numberRebars_bdimension=3; % number of rebars placed horizontally

% Total number of rebars placed over the cross-section
nv=2*numberRebars_hdimension+2*numberRebars_bdimension;
```

For this case, only a single rebar diameter design is to be considered, thus, the following variables are set:

```
RebarTypeIndex=2; % rebar diameter index

ov=rebarAvailable(RebarTypeIndex,1); % eight-of-an-inch rebar
dv=rebarAvailable(RebarTypeIndex,2); % rebar diameter (cm)
av=pi/4*dv^2; % rebar area
```

Now, the rebar coordinates or distribution of rebars over the cross-section are computed with the **function** *RebarDisposition* as follows:

```
% Rebar distribution over the cross-section
[dispositionRebar]=RebarDisposition(b,h,concreteCover,dv,nv,...
    numberRebars_hdimension,numberRebars_bdimension);
```

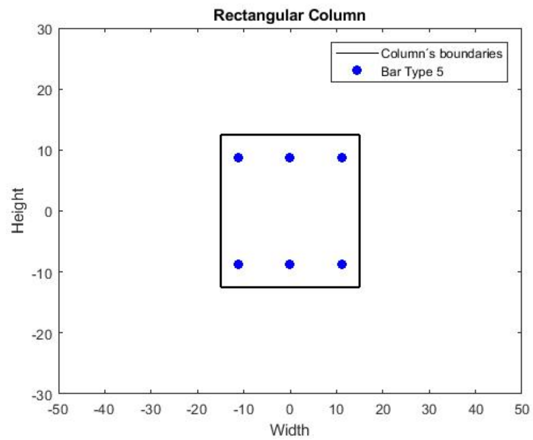
Finally, the interaction diagram for each cross-section axis (x and y) are computed with the function *diagramasDisposicion* to then, apply the Contour Load method and the Bresler's formula to determine the structural efficiency against the given load conditions:

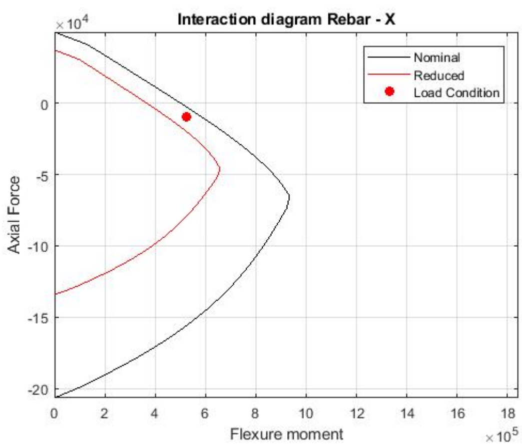
```
%% Structural efficiency
[diagrama,maxef,eficiencia,cxy]=diagramasDisposicion(As,b,h,E,npdiag,...
               fdpc,nv,betac,ov,av,dispositionRebar,load_conditions);
```

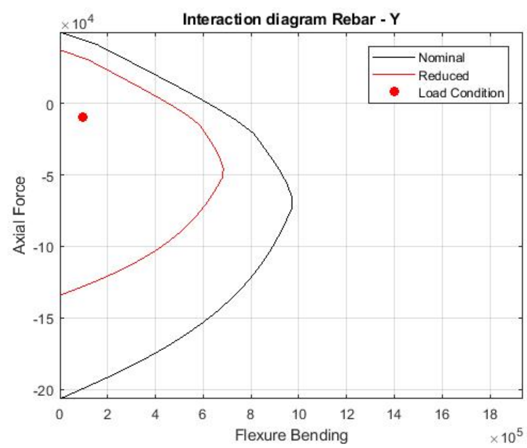
Results

In order to visualize the interaction diagram for each cross-section axis and the applied load conditions the function *diagramsFinalRebarCols* is used. But before using such function, a vector containing the rebar diameters' indices placed over the cross-section must be created, as following:

```
%% Plotting results
bestArrangement=zeros(nv,1)+RebarTypeIndex;

diagramsFinalRebarCols(load_conditions,diagrama,dispositionRebar,...
                h,b,bestArrangement);
```


Rectangular Column
Column´s boundaries
Bar Type 5
Height
Width
30
20
10
0
-10
-20
-30
-50
-40
-30
-20
-10
0
10
20
30
40
50


Interaction diagram Rebar - X
$\times 10^4$
Nominal
Reduced
Load Condition
Axial Force
Flexure moment
$\times 10^5$
0
-5
-10
-15
-20
0
2
4
6
8
10
12
14
16
18


Interaction diagram Rebar - Y
$\times 10^4$
Nominal
Reduced
Load Condition
Axial Force
Flexure Bending
$\times 10^5$
0
-5
-10
-15
-20
0
2
4
6
8
10
12
14
16
18

11.10 Ex10: Optimization design of a rectangular reinforced concrete column

Design_Columns_Ex01

Problem

Optimization design practices are everyday more popular in structural engineering to reduce material usage. When designing reinforced concrete structures, the optimization of rebar is of paramount importance given that it is the material with the highest rate of cost increment in the construction industry. The basic optimization task consists mainly in determining the rebar design with the lowest material volume.

Solution

CALRECOD offers functions focused on optimal design of rebar, either of asymmetrical or symmetrical distribution - for which asymmetrical distributions have usually the least area usage - such as the **function** *isrColumnsSymAsym*.

Let us, for instance, consider the RC column with the following geometry, materials:

```
%% GEOMETRY
height=300; % column's length
b=30; % cross-section width
h=30; % cross-section height

%% MATERIALS
fy=4200; % yield stress of rebars
fc=250; % concrete's compressive strength Kg/cm2
ws=0.0078; % volumetric weight of reinforcing steel (Kg/cm3)
```

And the following bending-compression loads:

```
%% Loads
load_conditions=[1 -53.52e3 9.602e5 1e-5]; % [nload, Pu, Mx, My]
```

Next, the rebar data is established, starting by the available commercial rebar diameters. For this purpose an array called *rebarsAvailable* is set as:

```
%% Rebar data
% Commerically available rebars
RebarAvailable=[4 4/8*2.54;
                5 5/8*2.54;
                6 6/8*2.54;
                8 8/8*2.54;
                9 9/8*2.54;
                10 10/8*2.54;
                12 12/8*2.54];
```

Before beginning the optimization process it would be necessary to set additional parameters and variables, such as *cols_sym_asym_isr* which indicates the desired rebar distribution (symmetrical or asymmetrical). For this first example, a symmetrical distribution will be established. Also, it is necessary to indicate if cracking mechanisms are required; this is done with the variable *condition_cracking*:

```
%% ADITIONAL PARAMETERS
rec=[4 4]; % [coverx covery] - concrete cover

cols_sym_asym_isr="Symmetric";
if cols_sym_asym_isr=="Symmetric" || cols_sym_asym_isr=="Asymmetric"
    pu_cols=[29.19, 29.06, 28.93, 28.93, 28.93, 28.93, 28.93]; % symmetric rebar

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

condition_cracking="Cracked"; % "Cracked" or "Non-cracked"

% Plot options:
optimaPlot=1; % optima reinforcement area convergence plot
plotsISRdiagrams=1; % interaction diagrams
plotRebarDesign=1; % detailed reinforced cross-section

% Ductility demand
ductility=3;
```

Now, the **function** *isrColumnsSymAsym* can be finally called:

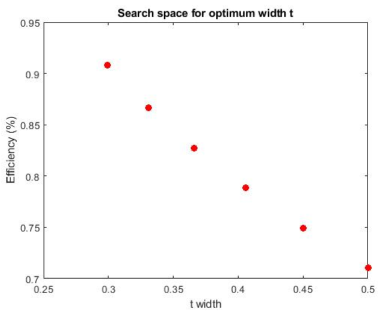
```
%% Optimization process
[Inertia_xy_modif,b,h,bestArrangement,bestdisposicionRebar,cost_elem_col,...
Ac_sec_elem,Ef_sec_col,Mr_col]=isrColumnsSymAsym(pu_cols,height,b,h,rec,...
fy,fc,load_conditions,cols_sym_asym_isr,condition_cracking,ductility,...
ws,RebarAvailable,optimaPlot,plotsISRdiagrams,plotRebarDesign);
```

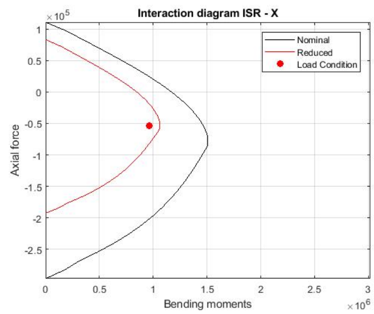
And the results can be then exported as .csv files to be visualized by the functions of Visual CALRECOD in Dynamo:

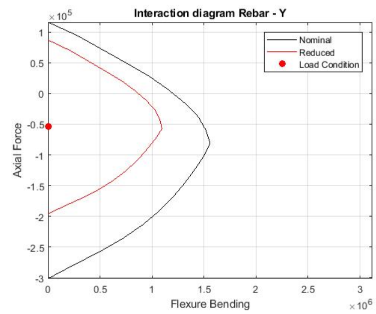
```
dimColumnsCollection=[b h height rec(1)];
nbarColumnsCollection=length(bestdisposicionRebar(:,1));

directionData='C:\Users\lfver\OneDrive\Desktop\OneDrive\DynamoRevit\Dynamo_visualization\';

ExportResultsColumn(directionData,dimColumnsCollection,bestdisposicionRebar,...
    nbarColumnsCollection,bestArrangement,cols_sym_asym_isr,[0,0,0],1);
```

Results

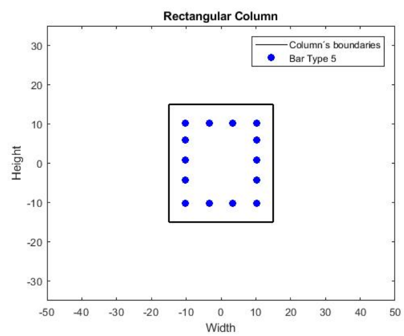
Firstly, the ISR optimization process takes place to determine an optimal reinforcement area to transform later to an optimal rebar design. For this reason, the optimization convergence graph of the ISR's width is shown below at the left panel, along with the interaction diagrams of such optimal ISR:

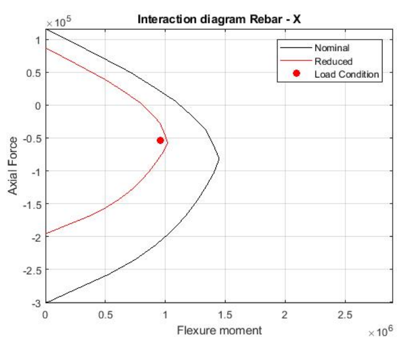


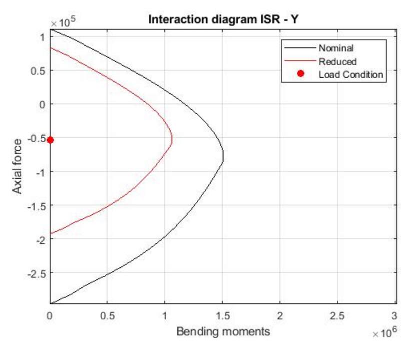




At last, the optimal symmetrical rebar design is plotted with its respective interaction diagrams in both cross-section axis:







11.11 Ex11: Optimization of asymmetrical design of a rectangular concrete column with packages of two rebars - Problem 1

Design_Columns_TwoPack_Ex01

Problem

In some cases, it is necessary to take advantage of the dimensions of a column cross-section so that the minimum rebar separation restriction established by codes my comply, even when the cross-section dimensions are two small (indicated for architectural purposes, for instance) and the required rebar quantities too high (given the load conditions). For such situations, rebars in packages of two are a very reliable option and are also usually allowed by design codes. This issue, combined with the requirement of placing as less rebar volumes as possible (optimization design process) gives way to relatively complex algorithms that are not that easy to compute for anyone.

Solution

CALRECOD offers functions for this purpose, either of symmetrical distribution or asymmetrical one, such as *superOptimalRebarSym2Pack*, *optimRebarColsSym2Pack*, *Asym2pack4Diam*, *Asym2pack1Diam*, *Asym2packSym4Diam*. For this example, a rebar design prototype of as many as four rebar diameters distributed in a symmetrical fashion as shown in **Fig. 84**, for which case, the **function** *Asym2packSym4Diam*:

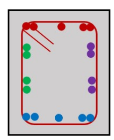


Figure 84: *Asymmetrical rebar design prototype consisting of as many as four rebar diameters distributed in a symmetrical fashion over a rectangular column cross-section.*

Let us start first by stating the column element's geometry and its materials as:

```
%% GEOMETRY
height=300; % column's length
b=45; % cross-section width
h=45; % cross-section height

%% MATERIALS
fy=4200; % yield stress of rebars
Es=fy/0.0021; % Modulus of elasticity of reinforcing steel

fc=250; % concrete's compressive strength Kg/cm2
fdpc=0.85*fc; % reduced f'c

if fc<280
    beta1=0.85;
elseif fc>=280
    beta1=1.05-fc/1400;
    if (beta1<0.65)
        beta1=0.65;
    elseif (beta1>0.85)
        beta1=0.85;
```

```
    end
end

ws=0.0078; % volumetric weight of reinforcing steel (Kg/cm3)
```

Now, let us state the following biaxial bending-compression load condition:

```
%% Loads
load_conditions=[1 -53.52e3 -15.602e5 19e5]; % [nload, Pu, Mx, My]
```

The available commercial rebars must also be established for the design process to consider. For this purpose the array *RebarAvailable* is created containing the eight-of-an-inch rebar data to choose from:

```
%% Rebar data
% Commerically available rebars
RebarAvailable=[4 4/8*2.54;
                5 5/8*2.54;
                6 6/8*2.54;
                8 8/8*2.54;
                9 9/8*2.54;
                10 10/8*2.54;
                12 12/8*2.54];
```

Other additional parameters and variables must be set too before beginning the optimization process, such as the concrete cover, demand of ductility, etc., as shown next:

```
%% ADITIONAL PARAMETERS
rec=[4 4]; % [coverx covery] - concrete cover
npdiag=30; % number of points to compute for the interaction diagram

pu_cols=[29.19,29.06,28.93,28.93,28.93,28.93,28.93]; % symmetrical
                                                      % rebar

pu_asym_cols=1/0.7*sum(pu_cols)/length(pu_cols); % asymmetrical
                                                      % rebar
pu_cols_isr=[28.93];

condition_cracking="Cracked"; % "Cracked" or "Non-cracked"

% Ductility demand
ductility=3;
```

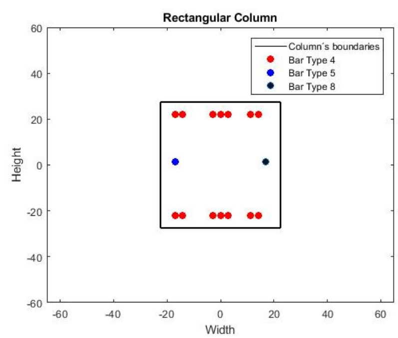
As it can be observed in the previous piece of code, a vector containing the unit-cost of basic symmetrical rebar assembly for each rebar diameter is required to then apply to it a such *Complexity Factor of Assembly* (CFA) - which is between 0 and 1 - so that, the average unit cost of rebar assembly for asymmetrical designs (more complex to perform by workers) may be increased, as it is shown below through the variable pu_asym_cols:

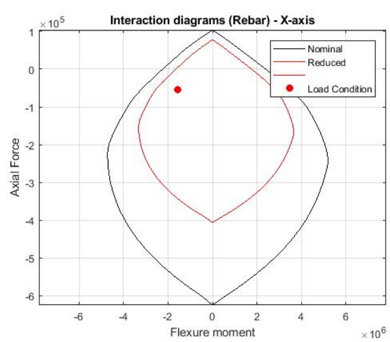
```
%% ISR optimization
[b,h,cost_elem_col,Ac_sec_elem,Ef_sec_col,Mr_col,t_value_01,t_value_03,cxy]=...
```

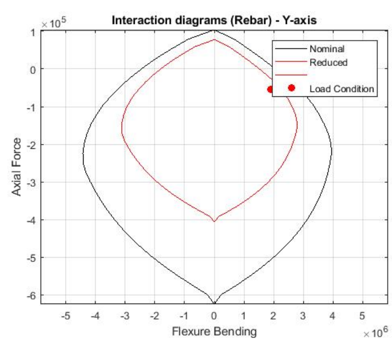
```
isrColumns(pu_cols_isr,height,b,h,rec,fy,fc,load_conditions,ws,ductility,...
0,0);

%% Optimization rebar design - RP: (Asym-2pack-Sym4Diam)

[Mr_col,h,Inertia_xy_modif,bestArea8,bestCost,bestdiagram,bestdiagram2,bestnv,...
bestEf,bestArrangement,bestDisposition,nv4,bestcxy,bestCP]=...
Asym2packSym4Diam(b,h,rec,Ac_sec_elem,Es,npdiag,fdpc,...
beta1,load_conditions,pu_asym_cols,ws,height,RebarAvailable,...
condition_cracking,ductility);
```

As it can be observed, an ISR optimization process is first performed before the rebar optimization design process in order to determine an approximated minimum area of reinforcement to comply with the given load conditions.

Results







11.12 Ex12: Optimization of asymmetrical design of a rectangular concrete column with packages of two rebars - Problem 2

Design_Columns_TwoPack_Ex02

Problem

Similar as the previous problem example Ex11, it would be sometimes required for columns subject to highly asymmetrical loads to go even beyond with unconventional rebar design, for which case, highly asymmetrical ones would turn out to be the optimum ones.

Solution

For such cases, the **function** *Asym2pack4Diam* that CALRECOD offers could be a great option with which rebar design prototypes such as the one shown in **Fig. 85** could be obtained, consisting of asymmetrical distributions of rebars with as many as four different rebar diameters:

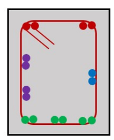


Figure 85: *Asymmetrical rebar design prototype consisting of as many as four rebar diameters distributed asymmetrically over a rectangular column cross-section.*

In order to use effectively this function the following should be recommended to be written in a prgram, beginning by stating the column element's geometry and its materials as:

```
%% GEOMETRY
height=300; % column's length
b=45; % cross-section width
h=45; % cross-section height

%% MATERIALS
fy=4200; % yield stress of rebars
Es=fy/0.0021; % Modulus of elasticity of reinforcing steel

fc=250; % concrete's compressive strength Kg/cm2
fdpc=0.85*fc; % reduced f'c

if fc<280
    beta1=0.85;
elseif fc>=280
    beta1=1.05-fc/1400;
    if (beta1<0.65)
        beta1=0.65;
    elseif (beta1>0.85)
        beta1=0.85;
    end
end
```

```
ws=0.0078; % volumetric weight of reinforcing steel (Kg/cm3)
```

Now, let us state the following biaxial bending-compression load condition:

```
%% Loads
load_conditions=[1 -53.52e3 -25.602e5 39e5]; % [nload, Pu, Mx, My]
```

The available commercial rebars must also be established for the design process to consider. For this purpose the array *RebarAvailable* is created containing the eight-of-an-inch rebar data to choose from:

```
%% Rebar data
% Commerically available rebars
RebarAvailable=[4 4/8*2.54;
                5 5/8*2.54;
                6 6/8*2.54;
                8 8/8*2.54;
                9 9/8*2.54;
                10 10/8*2.54;
                12 12/8*2.54];
```

Other additional parameters and variables must be set too before beginning the optimization process, such as the concrete cover, demand of ductility, etc., as shown next:

```
%% ADITIONAL PARAMETERS
rec=[4 4]; % [coverx covery] - concrete cover
npdiag=30; % number of points to compute for the interaction diagram

pu_cols=[29.19,29.06,28.93,28.93,28.93,28.93,28.93]; % symmetrical
                                                      % rebar

pu_asym_cols=1/0.6*sum(pu_cols)/length(pu_cols);      % asymmetrical
                                                      % rebar
pu_cols_isr=[28.93];

condition_cracking="Cracked"; % "Cracked" or "Non-cracked"

% Ductility demand
ductility=3;
```

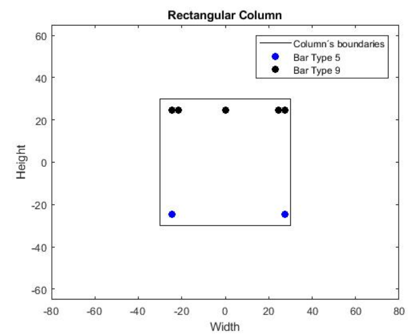
As it can be observed in the previous piece of code, a vector containing the unit-cost of basic symmetrical rebar assembly for each rebar diameter is required to then apply to it a such *Complexity Factor of Assembly* (CFA) - which is between 0 and 1 - so that, the average unit cost of rebar assembly for asymmetrical designs (more complex to perform by workers) may be increased, as it is shown through the variable pu_asym_cols:

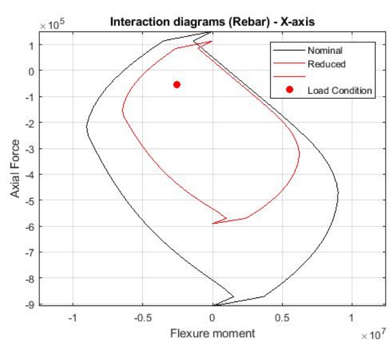
```
%% ISR optimization
[b,h,cost_elem_col,Ac_sec_elem,Ef_sec_col,Mr_col,t_value_01,t_value_03,cxy]=...
isrColumns(pu_cols_isr,height,b,h,rec,fy,fc,load_conditions,ws,ductility,...
```

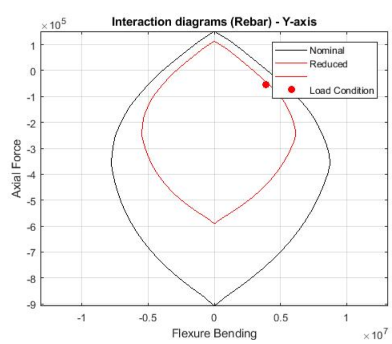
```
0,0);

%% Optimization rebar design - RP: (Asym-2pack-4Diam)

[Mr_col,h,Inertia_xy_modif,bestArea,bestCost,bestdiagram,bestdiagram2,...
bestnv,bestEf,bestArrangement,bestDisposition,nv4,bestcxy,bestCP]=...
Asym2pack4Diam(b,h,rec,Ac_sec_elem,npdiag,fdpc,beta1,...
fy,load_conditions,pu_asym_cols,RebarAvailable,ws,height,...
condition_cracking,ductility);
```

As it can be observed, an ISR optimization process is first performed before the rebar optimization design process in order to determine an approximated minimum area of reinforcement to comply with the given load conditions.

Results







11.13 Ex13: Required increment in reinforcing area in tension

RequiredAreaPercentageBeam_Ex01

Problem

It is often required in structural engineering to determine how much area in tension of a concrete beam would be needed so that the bending resistance can be increased in a certain amount.

Solution

Thanks to the function that both CALRECOD and MatLab offers such assessment can be carried out relatively easy with only a few lines of code. To show case the solution of this problem with CALRECOD let us consider the rectangular concrete beam element of the following dimensions, materials and reinforcement:

```
%% Geometry
b=30;
h=70;

b_rec=4;
h_rec=4;
d=h-h_rec;

%% Materials
fc=250;
factor_fc=0.85;
E=2e6;
fy=4200;

%% Rebar data
disposition_rebar=[-12 42;
                    -6 42;
                    0 42;
                    6 42;
                    12 42;
                    -12 -42;
                    -6 -42;
                    0 -42;
                    6 -42;
                    12 -42];

                %type diam          area
varDisponibles=[1 4 4/8.*2.54 (4/8*2.54).^2*pi./4;
                2 5 5/8*2.54 (5/8*2.54).^2*pi./4;
                3 6 6/8*2.54 (6/8*2.54).^2*pi./4;
                4 8 8/8*2.54 (8/8*2.54).^2*pi./4;
                5 9 9/8*2.54 (9/8*2.54).^2*pi./4;
                6 10 10/8*2.54 (10/8*2.54).^2*pi./4;
                7 12 12/8*2.54 (12/8*2.54).^2*pi./4];
```

```
ast=5*varDisponibles(2,4); % rebar area in tesnion
asc=ast; % rebar area in compression
```

Let us also set some additional variables and parameters such as the ductility demand of the reinforcing steel (variable *duct*):

```
%% Additional parameters
duct=3; % high ductility demand
```

Now, the number of times to increase the initial bending resistance would be necessary, therefore:

```
mmu=3.5; % number of times to increment the initial bending resistance

amax=(0.025*b*d); % max rebar area allowed in tension
```

Finally, the **function** *Efrec2tBeams* will be the one used in a for loop to determine the bending resistance of the beam cross-section for any amount of rebar area in tension, starting with the initial one until the required bending resistance increment is reached:

```
%% Main process

% Initial resistance with the current reinforcement quantity
[eftk1,mrast1,ck1]=Efrec2tBeams([1 20e5],fc,factor_fc,E,h,b,ast,asc,...
                                        b_rec,h_rec,0.85);
mrast=mrast1;
mmax=mmu*mrast;

% Loop
ast_vector=[];
Mrast_vector=[];
mrast=0;
while mrast<mmax
    [eftk1,mrast,ck1]=Efrec2tBeams([1 20e5],fc,factor_fc,E,h,b,ast,asc,...
                                        b_rec,h_rec,0.85);
    ast=ast+0.1;
    atotal=ast;
    ast_vector=[ast_vector,atotal];
    Mrast_vector=[Mrast_vector,mrast];
end
```

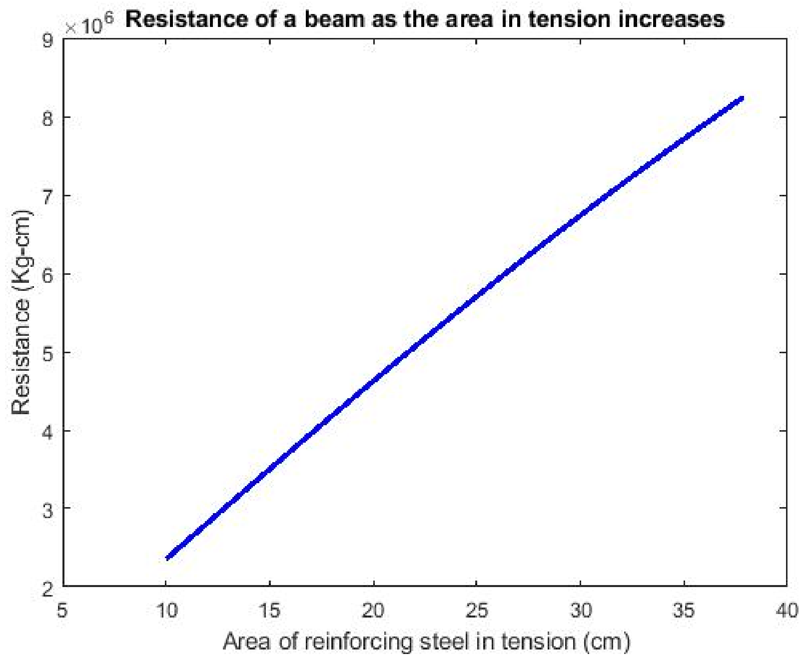
At the end, for better assessment, the results can be visualized, that is, the increment rate of the bending resistance as the rebar area in tension was increasing in the for loop:

```
%% Plotting results
figure(4)
plot(ast_vector,...
   Mrast_vector*1e-5,'b -','MarkerFaceColor','blue')
hold on
```

```
xlabel('Area of reinforcing steel in tension (cm)')
ylabel('Resistance (Ton-m)')
title('Resistance of a beam as the area in tension increases')
hold on
```

Results

As it can be seen in the following resulting plot, the increment of the bending resistance is not proportional to the rebar area increment, but instead a slight concave curve is obtained, which tells that in order for the beam cross-section's resistance to be incremented 3.5 times, the rebar area in tension must be increased up to $37.9cm^2$.



Beware of the units. The results here presented are in the same units of the entered data, by default.

11.14 Ex14: Structural efficiency of a retaining RC wall

Design_Retaining_RC_Wall_Ex01

Problem

When designing and assessing Retaining Reinforced Concrete Walls it is required to make sure that several restriction comply simultaneously, such as the Safety Factor against tipping, slide or the soil's bearing load capacity, as well as the structural efficiency limit for each wall's element.

Solution

With CALRECOD's **function** *RetainingRCWall* it is possible to assess such efficiencies and safety factors for a given set of geometry parameters, applied loads and the soil's mechanical properties.

Let us consider for instance the following wall's dimensions:

```
%% Dimensions (geometry)

H=220; % total wall's height
D=0; % depth of the back soil

m1=1e15; % wall's front stem's slope
m2=1e15; % wall's back stem's slope

toe=25; % toe's length
heel=120; % heel's length
hf=25; % width of toe and heel
b=20; % stem's upper width
```

Now, not only the mechanical properties of the reinforcing steel and concrete with which the wall is made have to be set in the program, but also those of the soil over which the wall is supported and the soil's fill at the front and back of the wall.

```
%% Materials

% Concrete and reinforcement ----------------------------------------
fc=250;
wvc=0.0024; % unit volume weight of the concrete
ductility=2; % ductility demand of the reinforcing steel

fy=4200; % yield stress of the reinforcing steel

% Soil fill ---------------------------------------------------------
FiFill=30; % friction coefficient of the front soil fill
wvFill=0.0018; % unit volume weight of the front soil fill
beta=0; % fill's top slope

% Backing ground fill -----------------------------------------------
```

```
FiBackFill=30; % friction coefficient of the back soil fill
alfa=0; % back fill's top slope

% Foundation's soil ---------------------------------------------------
FiFound=30; % friction coefficient of the foundation soil
```

Certain design restrictions must be given for the function to compute the structural efficiency of the reinforcing steel rebar design that it proposes, as well as other parameters such as the soil's bearing load capacity:

```
%% Design restrictions
qadm=1.25; % soil's bearing load capacity (Kg/cm2)
minFSqadm=1.0; % limit of the soil's load capacity safety factor

SlideSF=1.1; % safety factor against sliding forces over the wall
TippingSF=2.0; % safety factor against tipping forces over the wall

% Rebar separation ----------------------------------------------------
typeRebar=4;
sepMinRebars=10; % minimum rebar separation restriction for the
                 % reinforcement on each wall's element

% Structural efficiency -----------------------------------------------
maxEf=1.0; % limit of the structural efficiency of each
           % reinforced wall's element
```

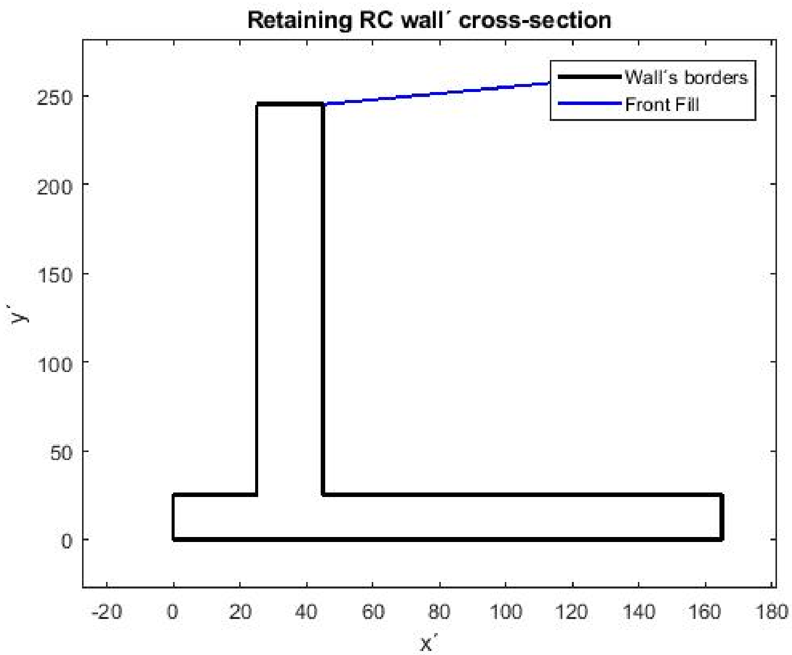
Finally, the external or additional loads aside of the fills pressures have to be set, such as overloads at the top of the back and front fill, as well as at the top of the wall's stem. In some cases, retaining walls also have the function to be the support of other structures such as bridges, therefore, the wall's stem are also subject to linear compression loads along the wall's length; that is the reason of the parameter qs

```
%% Overload
qaf=0.01; % over the front soil's fill
qab=0; % over the back soil's fill

%% Support load
qs=34.67; % vertical linear load acting along the top of the wall's stem

%% Load Factors
LF_DL=1.3; % Dead Load Factor for design
```

Finally, the **function** $RetainingRCWall$ is called as:

```
%% Design function
[compliedRestric,areaWall,linearWeigthWall,tippingFS,slideFS,LCap_FS,...
sepheel,efHeel,sepfoot,effoot,septrunk,eftrunk]=RetainingRCWall(toe,...
heel,hf,b,FiFill,H,D,m1,m2,wvFill,beta,FiBackFill,alfa,FiFound,fc,fy,wvc,...
ductility,qadm,minFSqadm,SlideSF,TippingSF,typeRebar,sepMinRebars,maxEf,...
qaf,qab,qs,LF_DL)
```

Results


Retaining RC wall´ cross-section
Wall´s borders
Front Fill
250
200
150
100
50
0
-20
0
20
40
60
80
100
120
140
160
180
y´
x´

11.15 Ex15: Optimization design of a retaining RC wall

PSO_Optim_RetainingRCWall_Ex01

Problem

When designing a retaining wall it is usually of the most importance to minimize as much as possible the wall's volume or weight. For such purpose the design process is an iterative one in which multiple dimension values for the wall's elements are changed, so that all the design restrictions are complied simultaneously.

Solution

Nowadays, thanks to certain algorithms and methods these such iterative processes in structural engineering can be automated easily by simply stating the proper objective function and the optimization variables. When dealing with more than two optimization variables meta-heuristic algorithms are of the most deployed ones for their versatility to adapt to these kind of problems. One of the most deployed optimization algorithms is the Particle Swarm Optimization (PSO) algorithm due to its simplicity in code structure and the efficiency with which optimum results are found.

CALRECOD offers the **function** *DesignRetainingWallPSO* with which the optimum design of a retaining reinforced concrete wall can be obtained based on certain given parameters and design restriction to comply with (see p. 214). This goal is achieved by taking the wall's dimensions $toe, heel, hf, b$ as the optimization design variables.

Let us consider for instance the following wall's dimensions which would remain constant through the design process:

```
%% Dimensions (geometry) - Optimization variables's range

H=220; % Total wall's height
D=0; % Back fill's depth

m1=1e15; % wall's stem's front slope
m2=1e15; % wall's stem's back slope
```

Now, not only the mechanical properties of the reinforcing steel and concrete with which the wall is made have to be set in the program, but also those of the soil over which the wall is supported and the soil's fill at the front and back of the wall.

```
%% Materials

% Soil fill -----------------------------------------------------------
FiFill=30; % friction coefficient
wvFill=0.0018; % unit volume weight
beta=0; % surface slope

% Backing ground fill ------------------------------------------------
FiBackFill=30; % friction coefficient
alfa=0; % surface slope

% Foundation's soil
```

```
FiFound=30; % friction coefficient

% Concrete and reinforcement ----------------------------------------
fc=250; % concrete's compressive strength (Kg/cm2)
wvc=0.0024; % concrete unit volume weight
ductility=2; % ductility demand for the reinforcing steel
fy=4200; % yield stress of the reinforcing steel (Kg/cm2)
```

The following design restrictions must be given to be complied for each potential design:

```
%% Design restrictions
qadm=1.25; % soil's bearing load capacity
minFSqadm=1.0; % safety factor against the soil's bearing load capacity

SlideSF=1.5; % safety factor against the sliding forces over the wall
TippingSF=2.0; % safety factor against tipping forces over the wall

% Rebar separation
typeRebar=4; % eight-of-an-inch rebar diameter to be used
sepMinRebars=10; % minimum rebar separation restriction to be complied
                 % for the reinforcement design of the wall

% Structural efficiency limit for each wall's element
maxEf=1.0;
```

Finally, the external or additional loads aside of the fills pressures have to be set, such as overloads at the top of the back and front fill, as well as at the top of the wall's stem. In some cases, retaining walls also have the function to be the support of other structures such as bridges, therefore, the wall's stem are also subject to linear compression loads along its length (qs):

```
%% Overload
qaf=0.01; % over the front fill (Kg/cm2)
qab=0; % over the back fill (Kg/cm2)

%% Support load
qs=34.67; % Kg/cm

%% Load Factors
LF_DL=1.3; % Dead Load Factor design
```

Finally, the **function** $DesignRetainingWallPSO$ can be called. However, before doing so, the search space for the optimization variables have to be established, that is, the dimension range values for the optimization process to look into, as following:

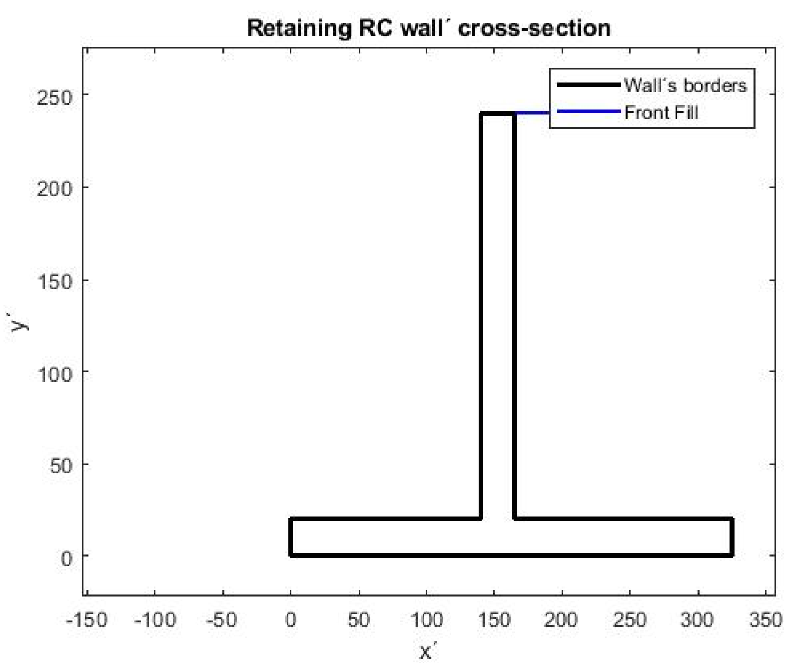
```
%% Optimization variables' search space
%      foot, heel, hf,  b
minDim=[20,  20,   15, 15];
MaxDim=[160, 160,  40, 40];
```

```
%% Optimization design function
[bestPerformance,bestPosition,besttippingFS,bestslideFS,...
bestLCap_FS,bestsepheel,bestefHeel,bestsepfoot,besteffoot,bestseptrunk,...
besteftrunk]=DesignRetainingWallPSO(minDim,MaxDim,H,D,m1,m2,FiFill,...
wvFill,beta,FiBackFill,alfa,FiFound,fc,fy,wvc,ductility,qadm,minFSqadm,...
SlideSF,TippingSF,typeRebar,sepMinRebars,maxEf,qaf,qab,qs,LF_DL)
```

After such optimization process is finished, the optimal design can be visualized by calling the function $plotRCWallDesign$, as shown in the following piece of code. Note that the optimal design variables are extracted from the vector $bestPosition$:

```
%% Ploting optimal design
toe=bestPosition(1);
heel=bestPosition(2);
hf=bestPosition(3);
b=bestPosition(4);

plotRCWallDesign(H,m1,m2,toe,heel,hf,b,D,alfa,beta)
```

Results



It is to stress that given the stochastic nature of the optimization algorithm PSO the optimal results might be slightly different from to run to run, although with only little variations in the wall's linear weight (or cross-section area). Therefore, it is recommended to run the process at least twice to choose the most convenient optimal design.

11.16 Ex16: Optimal rebar design of a T-beam cross-section

$OptimalDesign\_TBeams\_Ex01$

Problem

Optimization design practices are everyday more popular in structural engineering to reduce material usage. When designing reinforced concrete structures, the optimization of rebar is of paramount importance given that it is the material with the highest rate of cost increment in the construction industry. The basic optimization task consists mainly in determining the rebar design with the lowest material volume.

Solution

CALRECOD also offers functions focused on optimal design of rebar for T-beam cross-sections, either in individual rebars or in packages of two - such as the **function** *ISR1tRebarTBeamsOptim* (p. 96).

Next, an example of how to use this function is shown case. First off, the cross-section geometry and materials must be established, as well as the concrete cover:

```
%% Geometry
bp=20; % web width (cm)
ht=30; % total height (cm)
ba=60; % flange width (cm)
ha=12; % flange height or thickness (cm)
span=500; % beam's length (cm)

cover=3; % concrete cover

%% Material
fc=250; % concrete's compressive strength Kg/cm2
fy=4200; % Yield stress of steel reinforcement (Kg/cm2)
Es=2.0e6;
ffc=0.85;
fdpc=fc*ffc;
```

The load condition vector or array is given as follows, in which only the acting design bending moment is given (as beams are only design for pure flexure). Such bending moment(s) must be given in the second column of the array:

```
load_conditions=[1 6.55e5]; % [n-load, Mu]
```

Now, the available rebar types (eight-of-an-inch rebars) has to be given as an array with the following format:

```
%% Rebar data
% Available commercial rebar diameters (in eight-of-an-inch)
                %type diam
rebarAvailable=[4 4/8.*2.54;
                5 5/8*2.54;
```

```
                6 6/8*2.54;
                8 8/8*2.54;
                9 9/8*2.54;
                10 10/8*2.54;
                12 12/8*2.54];
wac=7.8e-3; % unit volume weight of the reinforcing steel
```

Additional parameters such as the ductility demand of for the reinforcing steel and unit-cost of the rebar assembly must also be set:

```
%% Additional data:
duct=3;

cols_sym_asym_isr="Rebar"; % ''Rebar'' or ''ISR''

% Plot options:
rebarDesignPlots=1; % for reinforced cross-section plots (0-No,1-Yes)
graphConvergencePlot=1; % to visualize the ISR optimization convergence

puTbeams=41.6; % unit construction assembly cost of steel reinforcement
```

Now, the optimization process starts with the ISR optimization to determine an approximate min required reinforcing area quantity in tension for the beam cross-section. For this purpose, the **function** $SGD1tTBeamsISR$ is called:

```
%% Optimization through the ISR analogy:
[cbest,bestMr,bestef,best_Area,tbest]=SGD1tTBeamsISR(bp,ht,ba,ha,span,duct,...
      cover,fc,load_conditions,ffc,Es,graphConvergencePlot);

% Compute cost with the ISR:
bpp=bp-2*cover;
tmin=(0.7*sqrt(fdpc)/fy*(bp*(d-ha)+ha*ba))/bpp; % min ISR's width in
                                                 % compression
t2Best=[tbest,tmin]; % ISR's widths in tension and compression
```

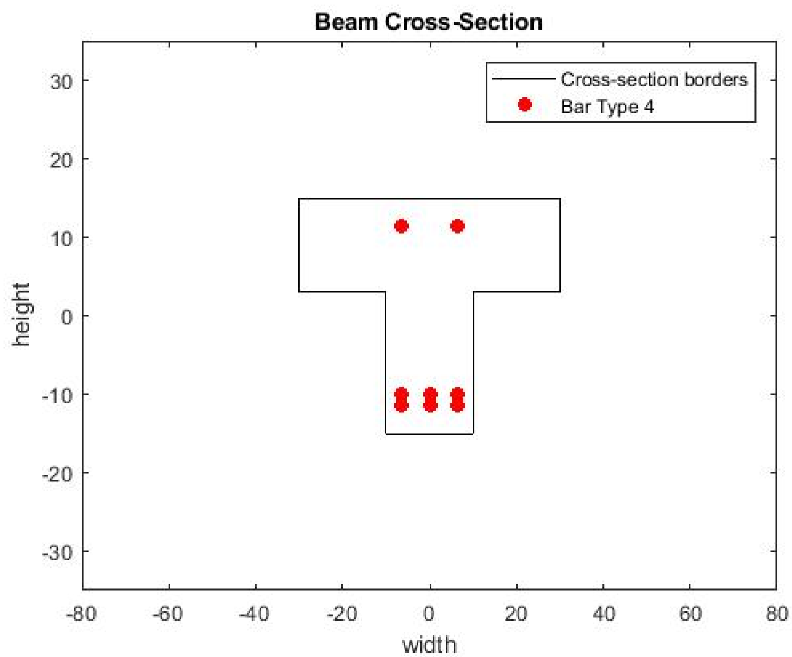
Finally, after the optimum ISR's width in tension is determined then the rebar optimization design process takes place through the **Function** *ISR1tRebarTBeamsOptim* as following:

```
%% Rebar design optimization:
if cols_sym_asym_isr~="ISR"

    [sepbarsRestric,cbest,bestBarDisposition,bestCostRebar,barTypes1Comp,...
        barTypesTen,ef,bestMr,areaRebar]=ISR1tRebarTBeamsOptim(bp,ht,...
        ba,ha,fc,cover,load_conditions,t2Best,puTbeams,span,rebarAvailable,...
        wac);
end
```

After the rebar optimization process is finished, the optimal design is visualized with the **function** $TbeamReinforcedSection$:

```
%% Plotting reinforced cross-section:
if rebarDesignPlots==1
    TbeamReinforcedSection(bp,ht,ba,ha,bestBarDisposition,...
        barTypes1Comp,barTypesTen);
end
```

Results



11.17 Ex17: Approximate optimal reinforcing area in tension of a rectangular beam cross-section

OptimaAreaBeamSection_Ex01

Problem

When designing a beam cross-section or of any other type it is recommendable first to estimate a quantity of reinforcing area with which the applied load conditions could be withstood. This first estimation could help to determine which rebar diameter(s) would be good to place in the element so that rebar separation restriction and other ones are complied.

Solution

For this purpose, CALRECOD contains functions to estimate the minimum required reinforcing area of a concrete section to withstand certain load conditions based on an idealization of the reinforcing steel as laminated (ISR - Idealized Smeared Reinforcement). The Steepest Gradient Descent method is used for this purpose. For rectangular beam cross-section the corresponding function is **function** *SGD1tBeamsISR* (p. 52).

Like any other function the element's geometry and materials must be first given, as well as other parameters:

```
%% Geometry
b=20; % cross-section width
h=40; % cross-section height
b_rec=4; % concrete cover in the horizontal direction
h_rec=3; % concrete cover in the vertical direction

%% Materials
fc=280; % concrete's compressive strength
E=2e6; % Modulus of Elasticity of the reinforcing steel

%% Additional parameters
duct=3;
factor_fc=0.85; % to compute the reduced f'c
```

The load condition vector or array is given as follows, in which only the acting design bending moment is given (as beams are only design for pure flexure). Such bending moment(s) must be given in the second column of the array:

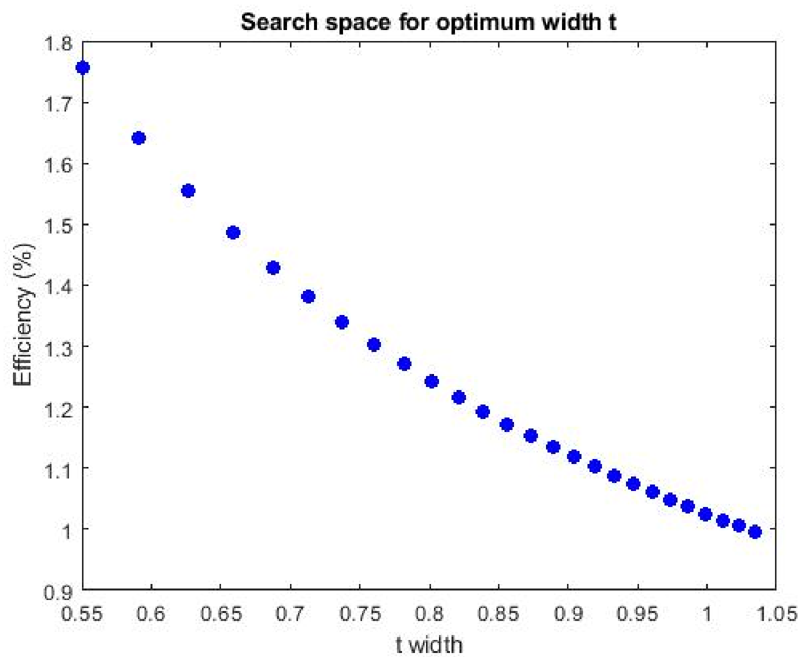
```
load_conditions=[1 15e5]; % [n-load, Mu]
```

Given that no rebar design will take place here, these are the only parameters required to call the main optimization function as following:

```
%% Optimization process
[cbest,bestMr,bestef,best_Area,tbest,h]=SGD1tBeamsISR(b,h,duct,...
            b_rec,h_rec,fc,load_conditions,factor_fc,E,1)
```

Results

As a result, the ISR's optimization convergence can be visualized to better appreciate the variation in the structural efficiency as the ISR's width changes.



As it can be observed in the plot, for this case the initial ISR's width was a bit small to withstand the applied load condition, therefore it had to be increased to reach an optimal structural efficiency of 0.996 (barely at the limit of 1.0).

11.18 Ex18: Comparison of resistance of a T-beam and a rectangular beam with the variation of the neutral axis depth

$NeutralAxisRecTbeams\_Ex01$

Problem

The reason to choose T-beam cross-sections instead of rectangular cross-sections is to obtain considerable increments in the bending resistance without increasing significantly the usage of concrete volumes. This is accomplished by the designing properly the dimension of the T-beam's flange and web. For such purpose, it would be useful to assess the variation of bending resistance as the neutral axis depth varies for various design options.

Solution

With the functions $eleMecanicosRebarTBeams$ and $eleMecanicosRebarBeams$ for T-beam cross-sections and rectangular beam cross-sections, respectively, it is possible to easily make such assessment. To do so, it would be recommendable to build a graph that depicts such relationship. An example is shown-case next.

Like any other function the element's geometry and materials must be first given, as well as other parameters. Let us start with the T-beam cross-section data as:

```
%% Geometry of the T-beam cross-section
bp=20; % web width (cm)
ht=40; % total height (cm)
ba=60; % flange width (cm)
ha=12; % flange height or thickness (cm)
span=500; % cm

cover=4; % lateral concrete cover

%% Materials
fc=250; % Kg/cm2
fy=4200; % Yield stress of steel reinforcement (Kg/cm2)
Es=fy/0.0021; % Modulus of elasticity of the reinforcing steel
fdpc=fc*0.85;

%% Additional general parameters
beta1=0.85; % For the reduction of the neutral axis depth as specified
            % by code
```

The following rebar data, such as its distribution over the cross-section will apply for both the T-beam and the rectangular-beam cross-sections, as both will have the same height dimension:

```
%% Rebar
dispositionRebar=[-7 -17;
                   0 -17;
                   7 -17;
                  -7 17;
                   7 17];
```

```
rebarType=[4;4;4;4;4];

% Database of the commercially available rebar
rebarAvailable=[4 4/8.*2.54;
                5 5/8*2.54;
                6 6/8*2.54;
                8 8/8*2.54;
                9 9/8*2.54;
                10 10/8*2.54;
                12 12/8*2.54];

As=sum(rebarAvailable(rebarType,2).^2.*pi./4); % Total rebar area
```

Now, for the T-beam cross-section a for loop will be computed in which the function *eleMecanicosRebarTBeams* is called in every iteration to generate the desired curve for assessment, thus:

```
%% Mr in function of the neutral axis (T-beam cross-section)
ce=[];
valuesMr=[];
for c=0:0.1:ht
    a=beta1*c;
    eMecVar=eleMecanicosRebarTBeams(c,a,fdpc,ha,ba,bp,ht,span,Es,rebarType,...
                                    dispositionRebar,rebarAvailable);

    Mr=eMecVar(1,2)+eMecVar(2,2);

    valuesMr=[valuesMr,Mr];
    ce=[ce,c];
end

figure(2)
plot(ce,valuesMr,'k -')
legend('T-beam section')
xlabel('Neutral axis depth c (cm)')
ylabel('Bending resistance (Kg-cm)')
title('Bending resistance in function of the neutral axis depth for a beam cross-section')
grid on
hold on
```

Now, for the rectangular beam cross-section the following geometry applies:

```
%% Geometry of a rectangular beam cross-section
b=20; % section's width (cm)
h=40; % section's height (cm)
```

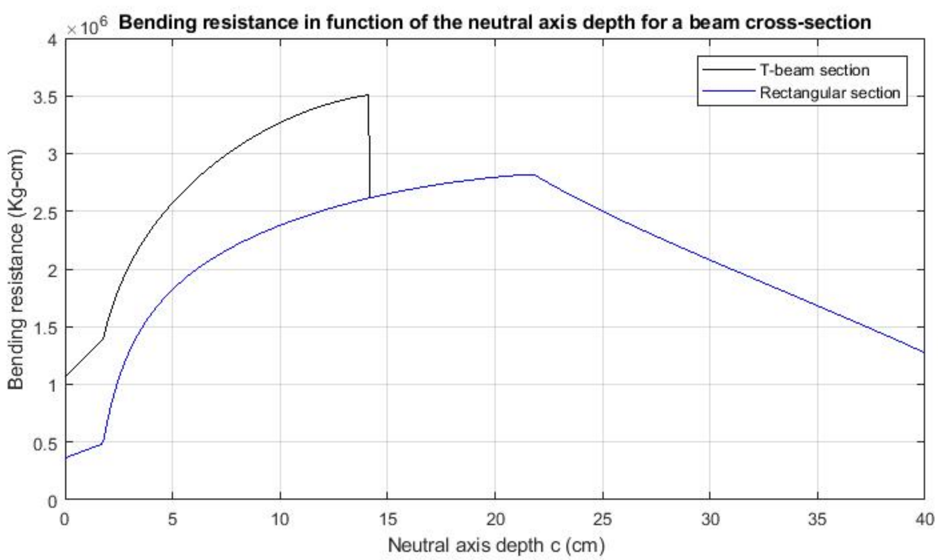
Therefore, its corresponding loop for assessment will be computed as:

```
%% Mr in function of the neutral axis (rectangular cross-section)
ce=[];
```

```
valuesMr=[];
for c=0:0.1:h
    a=beta1*c;

    eMecVar=eleMecanicosRebarBeams(c,a,fdpc,h,b,cover,Es,rebarTypeT,...
                        rebarTypeC,dispositionRebar,rebarAvailable);

    Mr=eMecVar(1,2)+eMecVar(2,2);

    valuesMr=[valuesMr,Mr];
    ce=[ce,c];
end

figure(2)
plot(ce,valuesMr,'b -','DisplayName','Rectangular section')
grid on
hold on
```

Results

As it can be observed in the following plot, the increase in bending moment resistance is indeed larger for the T-beam cross-section but only for small values of the neutral axis depht, which are the common ones for the design of beam elements given that for those small values the T-beam cross-section is actually working a T-beam cross-section (as specified by code). As the neutral axis depth value overpasses the flange thickness ($ha = 12cm$ in this case) the T-beam cross-section will work more likely as a rectangular cross-section, therefore, with an equal resistance than the rectangular cross-section.



As it could be thought, many other interesting plots could be computed with these functions, such as the increase in bending moment resistance as the usage of concrete increases, etc.

11.19 Ex19: Structural efficiency of a rebar reinforced rectnagular beam cross-section

$Efficiency\_Beam\_RebarDesign\_Ex01$

Problem

It is required to assess if a rebar design of a rectangular beam cross-section complies with the structural efficiency Eff restriction $Eff < 1.0$ against a set of load conditions.

Solution

First off, the cross-section geometry and materials must be established, as well as the concrete cover:

```
%% Geometry
b=25; % cross-section width
h=50; % cross-section height

b_rec=3; % concrete cover on each direction
h_rec=3;

d=h-h_rec; % effective cross-section height

%% Materials
fc=250; % concrete's compressive strength
factor_fc=0.85; % reduction factor for the f'c
fdpc=factor_fc*fc; % reduced f'c
E=2e6; % Modulus of elasticity of the reinforcing steel
fy=4200; % Yield stress of the reinforcing steel

%% Additional parameters
duct=3; % ductility demand level
```

The load condition vector or array is given as follows, in which only the acting design bending moment is given (as beams are only design for pure flexure). Such bending moment(s) must be given in the second column of the array:

```
%% Loads
load_conditions=[1 -5.26e5]; % [n-load, Mu] (Kg-cm)
```

Now, the available rebar types (eight-of-an-inch rebars) has to be given as an array with the following format:

```
%% Rebar data
% Commercially available rebar diameters
                %type diam
rebarAvailable=[4 4/8.*2.54;
                5 5/8*2.54;
                6 6/8*2.54;
                8 8/8*2.54;
```

```
                9 9/8*2.54;
                10 10/8*2.54;
                12 12/8*2.54];
```

Then, the rebar data is set, starting with the rebar local coordinates over the beam cross-section. For this purpose, an array of two columns is set as following, in which the first column corresponds to the x's coordinates and the second column to the y's coordinates:

```
% Distribution of rebars over the cross-section
disposition_rebar=[-9 21;
                    9 21;
                    -9 -21;
                    0 -21;
                    9 -21];
```

Given that each row of the previous *dispositionRebar* array corresponds to a rebar, another vector is necessary to indicate what eight-of-an-inch each rebar is. For this purpose the following vectors are set, both for the rebar in tension and compression, in which each of its elements are an index corresponding to a respective row of the previous given *rebarAvailable* array.

```
RebarIndexTen=[1;1;1]; % rebar diameters to use for the reinforcement
                    % in tension (indices from the "rebarAvailable" array)

RebarIndexCom=[1;1]; % rebar diameters to use for the reinforcement
                  % in compression (indices from the "rebarAvailable" array)
```

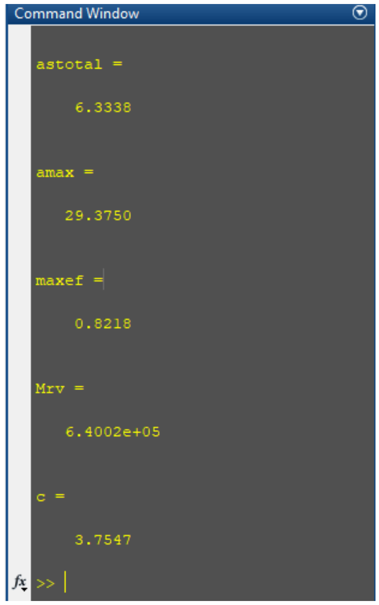
Finally, the resistance efficiency is evaluated with the **Function** *EfcriticalRebarbeams* as following:

```
%% Structural efficiency
[maxef,Mrv,c]=EfcriticalRebarbeams(load_conditions,b,E,fdpc,RebarIndexTen,...
    RebarIndexCom,rebarAvailable,d,h_rec,0.85,dispositionRebar)
```

As it can be observed in the output set of the function, not only the structural efficiency can be obtained but also the resistant bending moment Mr and the corresponding neutral axis depth cx.

Results

As it can be observed in the following output, the resistance efficiency of 0.82 indicates that the design complies with the restriction $Ef < 1.0$.



Beware of the units.

11.20 Ex20: Optimal rebar design of a rectangular-beam element

Optimal_Design_RecBeams_Ex01

Problem

Optimization design practices are everyday more popular in structural engineering to reduce material usage. When designing reinforced concrete structures, the optimization of rebar is of paramount importance given that it is the material with the highest rate of cost increment in the construction industry. The basic optimization task consists mainly in determining the rebar design with the lowest material volume.

For the design of beam elements usually the three main cross-sections are considered simultaneously for the overall design of the whole element so that there may be rebar cuts and laps at certain locations of the beam length. However, this is not an easy task to compute, nor less to optimize.

Solution

CALRECOD offers a function to optimally design the rebar for the three main cross-sections of a beam element (at the left end, at the middle and at the right end) - named *beamsISR*, although it can only do it separately for each cross-section without considering cuts and overlapping reinforcement criteria. In any case, the potential results can be very useful and insightful. It is expected, however, that in future versions of CALRECOD such design functions will be included.

To illustrate the use of such function let us consider the following example of beam element with the geometry and materials shown next:

```
%% Geometry
span=500; % cm
b=30; % width (cm)
h=60; % height (cm)

h_rec=5; %
b_rec=3; % lateral concrete cover

%% Materials
fc=280; % Kg/cm2
fy=4200; % Yield stress of steel reinforcement (Kg/cm2)
```

The following additional variables must be also set, such as the ductility demand level of the reinforcing steel and the unit-cost of rebar assembly (per unit of weight):

```
%% Additional parameters
cols_sym_asym_isr="Rebar"; % ''Rebar'' or ''ISR''

% Plot options
plots=1; % for reinforced cross-section plotts (0-No,1-Yes)
graphConvergence=1; % for optimal ISR area convergence (0-No,1-Yes)
```

```
pu_beams=38.6; % unit construction assembly cost of steel reinforcement
duct=1; % high ductility demand
```

The bending load condition vector or array *load_conditions* is given as follows (as beam cross-sections are only designed for pure flexure). Such bending moment(s) must be given in the second column of the array. Also, the shear design forces are given to design the separation of the stirrups along the length of the whole element:

```
%% Load conditions
load_conditions=[1 -33.0e5 29.0e5 -31.0e5]; %Kg-cm (flexure)
shear_beam=linspace(12e3,-22e3,7); %Kg (shear)
```

Now, the available rebar types (eight-of-an-inch rebars) has to be given as an array with the following format:

```
%% Rebar data
% Available commercial rebar diameters (in eight-of-an-inch)
                %type diam
rebarAvailable=[4 4/8.*2.54;
                5 5/8*2.54;
                6 6/8*2.54;
                8 8/8*2.54;
                9 9/8*2.54;
                10 10/8*2.54;
                12 12/8*2.54];
wac=7.8e-3; % unit volume weight of the reinforcing steel
```

Finally, the optimization design process for the longitudinal rebar takes place. For this purpose, the **function** *beamsISR* is called:

```
%% OPTIMAL DESIGN
[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,wac,b,h,h_rec,rebarAvailable,...
fc,fy,load_conditions,cols_sym_asym_isr,duct,b_rec,plots,graphConvergence);
```

Then, the design process for the transversal reinforcement, by calling the function *shearDesignBeams*:

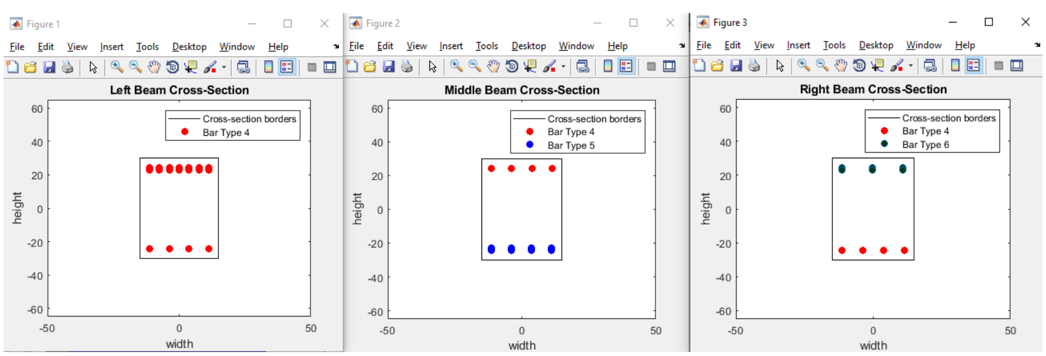
```
%% SHEAR DESIGN
rho=(sum(minAreaVar_3sec)/length(minAreaVar_3sec))/(b*h);
[s1,s2,s3,d1,d3]=shearDesignBeams(span,b,h,h_rec,rho,fc,fy,shear_beam);
```

After the whole reinforcement design process is finished, the resulting design parameters are exported in .csv files so that they can be later visualized through Visual CALRECOD's functions in Dynamo:

```
%% Exporting results
dim_beams_collection(1,:)=[b h span h_rec]; % general geometry data
```

```
shear_beam_design_collec=[s1,s2,s3,d1,d3]; % shear design data

disposition_rebar_beams3sec=[dispositionBar_Izq; % rebar design coordinates
                             dispositionBar_Center;
                             dispositionBar_Der];

              % number of rebar for each cross-section
              % both in tension and compression
nbar_beams_collection(1,:)=[length(barArrangementIzqTens)...
                            length(barArrangementIzqComp)...
                            length(barArrangementCentralTens)...
                            length(barArrangementCentralComp)...
                            length(barArrangementDerTens)...
                            length(barArrangementDerComp)];

              % rebar diameters at each cross-section
              % in tension and compression
beamsDiamIndex=[barArrangementIzqTens;
                    barArrangementIzqComp;
                    barArrangementCentralTens;
                    barArrangementCentralComp;
                    barArrangementDerTens;
                    barArrangementDerComp];

collection_beams_diamIndex=[beamsDiamIndex];

coordEndBeams=[0 0 300]; % global location of the left end of the beam

directionData='C:\Users\luizv\OneDrive\DynamoRevit\Dynamo_visualization\';
ExportResultsBeam(directionData,dim_beams_collection,coordEndBeams,...
                disposition_rebar_beams3sec,nbar_beams_collection,...
                collection_beams_diamIndex,cols_sym_asym_isr,...
                shear_beam_design_collec);
```

Results



11.21 Ex21: Plastic Center of an asymmetrically reinforced rectangular column

$PlasticCenter\_Ex01$

Problem

When designing asymmetrically reinforced columns in order to obtain better savings in rebar volumes it is of paramount importance to determine the Plastic Center of the column cross-section in order to accurately compute its resistance. Given the asymmetrical disposition of the reinforcement, such Plastic Center location will differ from the Geometrical Center location (which is at a depth of $[h/2, b/2]$) for each cross-section axis (x and y). The highest the asymmetry of the reinforcement the more the Plastic Center depth values will differ from the Geometrical Center depths.

Solution

With the **function** *PlastiCenterAxis* that CALRECOD offers such Plastic Center depths turn out quite easy to determine.

Let us consider for instance the following column element with the next geometry and materials:

```
%% Column's geometry
b=40; % cross-section width dimension (cm)
h=40; % cross-section height dimension (cm)

concreteCover=[4 4]; % concrete cover in both cross-section directions (cm)

%% Materials
fc=300; % concrete's compressive strength (Kg/cm2)
fdpc=fc*0.85; % reduced concrete's strength (Kg/cm2)
fy=4200; % yield stress of the reinforcing steel (Kg/cm2)
```

Now, the rebar data must be set, starting with the available rebar types (eight-of-an-inch rebars) as an array with the following format:

```
%% Rebar data
% Commercially available rebar diameters. [eight-of-an-inch, diam(cm)]
rebarAvailable=[4 4/8*2.54;
                5 5/8*2.54;
                6 6/8*2.54;
                8 8/8*2.54;
                9 9/8*2.54;
                10 10/8*2.54;
                12 12/8*2.54];
```

Next, the rebar diameters distribution and the number of rebars to be places for each cross-section boundary are set as:

```
% number of rebars at each cross-section's boundary
numberRebars1=9;
```

```
numberRebars2=3;
numberRebars3=5;
numberRebars4=2;

% rebar diameter at each cross-section's boundary
RebarTypeIndex1=4;
RebarTypeIndex2=3;
RebarTypeIndex3=5;
RebarTypeIndex4=3;

% Total number of rebars placed over the cross-section
nv=numberRebars1+numberRebars2+numberRebars3+numberRebars4; % number of
                                                            % rebars
```

Now, the rebar coordinates can be computed with the **function** *dispositionRebarAsymmetric* as shown below:

```
% Rebar distribution
[dispositionRebar,separacion_hor1,separacion_hor2,...
separacion_ver1,separacion_ver2]=dispositionRebarAsymmetric(b,...
h,concreteCover,nv,numberRebars1,numberRebars2,...
numberRebars3,numberRebars4,rebarAvailable,RebarTypeIndex1,...
RebarTypeIndex2,RebarTypeIndex3,RebarTypeIndex4);
```

Finally, the Plastic Center depth values for each cross-section axis can be computed. For this purpose, the **function** *PlastiCenterAxis* must be called twice (once for each cross-section axis). Let us start with depth with respect to the x axis, as shown next:

Note that a vector containing the rebar diameters' indices must be given, starting with the rebars at top boundary of the cross-section, then at the bottom, left and right, in that order.

```
%% Computation of the Plastic Center with respect to the X-axis
rebarTypeslist(1:numberRebars1)=RebarTypeIndex1;
rebarTypeslist(numberRebars1+1:numberRebars1+numberRebars2)=RebarTypeIndex2;
rebarTypeslist(numberRebars1+numberRebars2+1:numberRebars1+numberRebars2+...
    numberRebars3)=RebarTypeIndex3;
rebarTypeslist(numberRebars1+numberRebars2+numberRebars3+1:nv)=RebarTypeIndex4;

[PCX]=PlastiCenterAxis(fy,fdpc,b,h,dispositionRebar,rebarTypeslist,...
                            rebarAvailable)
```

Then, in order to accurately compute the Plastic Center's depth with respect to the y axis, the rebar coordinates, cross-section dimensions and rebar diameters distributon must be inverted (or rotated 90 degrees) so that the **function** *PlastiCenterAxis* can be used again in the same format. Thus:

```
%% Computation of the Plastic Center with respect to the Y-axis
rebar=dispositionRebar;

% Invert rebar local coordinates (the cross-section is rotated 90°)
```

```
dispositionRebar(:,1)=-rebar(:,2);
dispositionRebar(:,2)=rebar(:,1);

% Invert cross-section dimensions
dimensionesColumna=[b h];
h=dimensionesColumna(1);
b=dimensionesColumna(2);

% Invert rebar diameters
rebarTypeslist(1:numberRebars3)=RebarTypeIndex3;
rebarTypeslist(numberRebars3+1:numberRebars3+numberRebars4)=RebarTypeIndex4;
rebarTypeslist(numberRebars3+numberRebars4+1:numberRebars3+numberRebars4+...
    numberRebars2)=RebarTypeIndex2;
rebarTypeslist(numberRebars3+numberRebars4+numberRebars2+1:nv)=RebarTypeIndex1;

[PCY]=PlastiCenterAxis(fy,fdpc,b,h,dispositionRebar,rebarTypeslist,...
                             rebarAvailable)
```

After both Plastic Center's depths have been determined, then it is possible to visualize the results, both of the reinforced cross-section layout and the Plastic Center location. Let us begin by plotting the cross-section layout, for which the **function** $beamReinforcedSection$ can be called as:

```
%% Plotting results
% Rebar diameters' indices
barTypes1(1:numberRebars1)=RebarTypeIndex1;
barTypes1(numberRebars1+1:numberRebars1+numberRebars2)=RebarTypeIndex2;
barTypes2(1:numberRebars3)=RebarTypeIndex3;
barTypes2(numberRebars3+1:numberRebars3+numberRebars4)=RebarTypeIndex4;

% Calling function "beamReinforcedSection"
beamReinforcedSection(h,b,rebar,barTypes1,barTypes2)
```

Note that the function requires two vector containing the rebar diameters' indices of the rebars distributed horizontally and vertically, respectively.

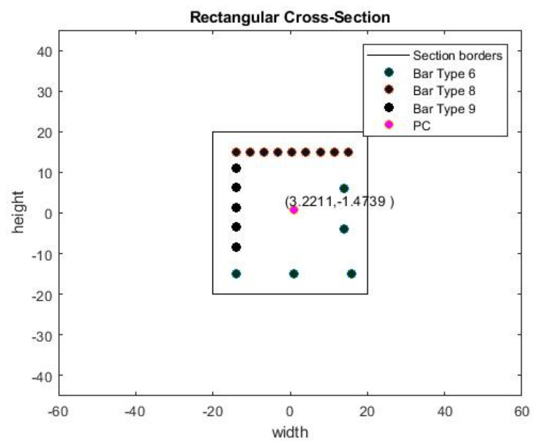
Now, in order to visualize the location of the computed Plastic Center in the same system of reference of the reinforced cross-section layout, the coordinates of such Plastic Center must be determined (given that the output of the function $PlastiCenterAxis$ are not coordinates but depth values with respect to the outer most cross-section boundary in the axis of reference). Therefore:

```
% Coordinates of the Plastic Center
xpc=h/2-PCX;
ypc=b/2-PCY;

% Ploting the Plastic Center location
PCXText=num2str(xpc); % To convert the numerical values in strings
PCYText=num2str(ypc);
PCXY=strcat('( ',PCXText,', ',PCYText,' )'); % To concatenate strings
figure(2)
```

```
plot(ypc+2.5,xpc-2.5,'o','MarkerFaceColor','magenta','DisplayName','PC')
text(ypc,xpc,PCXY) % To plot the coordinates as text
```

Note that the figure(2) was called given that the reinforced cross-section layout was plotted in that figure dimension in the function beamReinforcedSection as it is shown next:

Results



11.22 Ex22: Distribution of soil pressures in a rectangular isolated footing

$EffectivePressureIsoFooting\_Ex01$

Problem

When designing isolated footings t is very important to have a good insight of how the soil pressures are distributed at the bottom of the footing, according to the biaxial loads that a column may transfer. For this purpose, it is quite acceptable to adopt the theory of *plane surfaces*, that is, considering that the footing's surface remains plane at all times.

Solution

For this purpose, CALRECOD offers the function $RealPressuresFoot$ which computes the soil pressure at each of the four corners of the rectangular isolated footing, depending on the type of isolated footing in question (either central standard footing - with the column at the center, border footing or a corner footing).

Let us consider for instance the following isolated footing element with the next geometry and materials:

```
%% Basic material and geometry parameters
hefoot=25; % initial footing's height dimensions (cm)
rec=5; % concrete cover in all directions
d=hefoot-rec; % initial effective footing's height
bc=30; hc=30;  % transversal dimensions of the column
dimCol=[bc,hc];

fc=250; % concrete's compressive strneght f'c
fy=4200; % Yield stress of steel reinforcement (Kg/cm2)
```

Additional parameters and variables must be set for a proper assessment, such as the type of isolated footing, admissible soil's load capacity, among others:

```
%% Additional parameters
typeFooting=1; % Type of footing (1, 2 or 3) - see documentation
qadm=1.25; % Admissible bearing load of soil (Kg/cm2)
FS=1.2; % Safety Factor for the contact pressures
qu=qadm/FS; % Reduced soil's bearing load capacity
```

The biaxial loads that the column transfers to the isolated footing must also be given, in format:

```
load_cols=[1 -10.95e3 -1.8e5 -0.85e5]; % [n-load, Pu, Mux, Muy] Kg,cm
pu=load_cols(1,2);
```

Now, given that transversal dimensions of the isolated footing are designed from the bearing load capacity, the **function** *designDimFootings* is first called for such purpose, so that the Safety Factor for such soil's bearing load capacity is considered:

```
%% Design of footing dimensions
% Note: in case it is required to design transversal dimensions
[be,le,contact_pressure]=designDimFootings(pu,qu,dimCol,hefoot,rec,...
                         typeFooting)

dim_zap=[be le];
```

Finally, it is possible to determine the soil's pressures distributions by calling the **function** *RealPressuresFoot*:

```
%% Distribution of soil pressures
[qu01,qu02,qu03,qu04,qprom]=RealPressuresFoot(load_cols,be,le,typeFooting,...
                             dimCol,1);
```

Additionally, in order to design an efficient thickness of the isolated footing, shear forces must be considered. For this purpose **function** *shearFootings* can be called as:

```
%% Shear design
[d,qmax]=shearFootings(be,le,qprom,dimCol,pu,d,fc,typeFooting)
hefoot=d+rec; % Modified footing's height: minimum required by shear
```

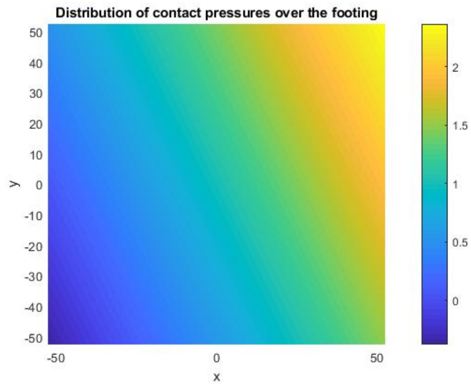
From which then, the design bending moments for reinforcement in both directions can be computed by calling the function *MomentDistributionFootings* twice (once for each axis direction). Thus, for standard isolated footings whose column is at their center:

```
%% Distribution of flexure over each footing's cross-section
dimpy=(be-hc-d)*0.5;
[mrL]=MomentDistributionFootings(qmax,dimpy,le)
dimpx=(le-bc-d)*0.5;
[mrB]=MomentDistributionFootings(qmax,dimpx,be)
```

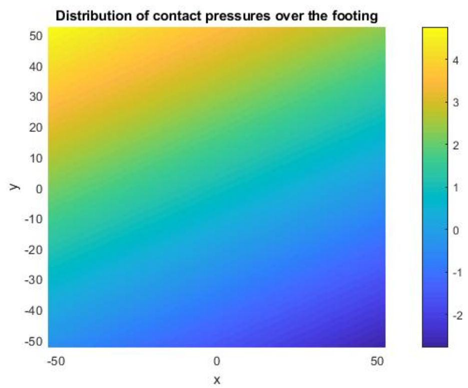
Results

For the results, the positive sign in the contact pressures indicates that the soil is in compression whereas a negative sign indicates that the soil is in tension (or that the pressures in those negative sign areas can be neglected), as shown below:

Note that the max soil's pressure is at the upper right corner, which is congruent with the sign of the given bending-compression loads.

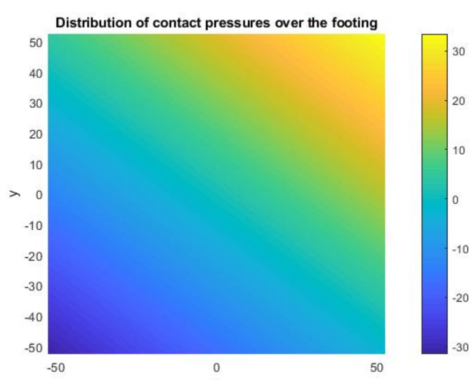


When changing the parameter's value $typeFooting = 3$ (corresponding to a corner footing) the results are:



Note that the max pressure is at the top left corner (which according to the sign of the bending compression loads does not make too much sense), however, given that it is considered indeed that the column is at such top left corner for this type of footing, then what such results actually tell us is that in order for the max soil pressure to move to the top right corner the bending moments must be way higher, as shown below for the increased loads:

```
load_cols=[1 -10.95e3 -11.8e5 -10.85e5]; % [n-load, Pu, Mux, Muy] Kg,cm
```



11.23 Ex23: Optimal design of rebar for a rectangular isolated footing

$Design\_Footings\_Ex01$

Problem

When designing rectangular isolated footings, both of its transversal cross-sections should be designed according to the distributed bending moments transferred from the soil's pressures. For this purpose it is common to establish min rebar separation criteria way higher than the established by code. Similar as for beams, the main goal is to determine the minimum rebar quantity in tension, which is placed at the bottom of the footing's cross-sections. The steel at the top, on the other hand, is usually designed with the min quantity established by code to withstand temperature forces.

This such process of optimization design can be easily computed through linear-search methods with great acceptable results, similar as for beams.

Solution

For this purpose, CALRECOD offers the function *isrFootings* with which it is possible to find an optimal rebar design for the both footing's cross-sections - if desired, or simply the optimal reinforcement quantity:

Let us consider for instance the following isolated footing element with the next geometry and material parameters:

```
%% Geometry data
rec=5; % concrete cover in all directions
hefoot=25; % initial footing height dimensions (cm)
d=hefoot-rec; % effective footing's height (cm)
bc=30; hc=30; % supporting column's dimensions
dimCol=[bc,hc];

%% Materials
fy=4200; % Yield stress of steel reinforcement (Kg/cm2)
fc=250; % concrete's compressive strength f'c (Kg/cm2)
wac=7.8e-3; % unit volume weight of the reinforcing steel (Kg/cm3)
```

Additional parameters and variables must be set for a proper assessment, such as the type of isolated footing, admissible soil's load capacity, among others:

```
%% Additional data
qadm=1.5; % Admissible bearing load of soil (Kg/cm2)
FS=1.3; % safety factor for the soil's bearing load capacity
qu=qadm/FS;
pu_steel_footings=26.75; % unit construction cost of reinforcement assembly
                         % (Per kilogram of reinforcing steel)

typeFoot=3; % Type of footing (1, 2 or 3) - see documentation
sepMinRebar=15; % Minimum rebar separation considered (cm) greater or
                % equal than the min absoulute give by code: 4/3*0.75 in
```

```
cols_sym_asym_isr="Rebar"; % ''ISR'' or anything else (''Symmetric'', ''Rebar'')
ductility=3; % to set the MAX reinforcement quantity (1, 2 or 3)

optimConv=1; % to visualize the optimization convergence plot
PlotRebarDesign=1; % To visualize the optimal rebar design layouts
```

The biaxial loads that the column transfers to the isolated footing must also be given, in format:

```
%% Loads
load_cols=[1 -14.59e3 -1.2e5 1.13e5]; % [n-load, Pu, Mux, Muy] Kg,cm
pu=load_cols(1,2);
```

The commercially available rebar diameters to choose from must be established. For this purpose, the following arrays is set:

```
%% Rebar data
% Commercially available rebar
                %type diam
RebarAvailable=[4 4/8.*2.54;
                5 5/8*2.54;
                6 6/8*2.54;
                8 8/8*2.54;
                9 9/8*2.54;
                10 10/8*2.54;
                12 12/8*2.54];
```

Now, given that transversal dimensions of the isolated footing are designed from the bearing load capacity, the **function** *designDimFootings* is first called for such purpose, so that the Safety Factor for such soil's bearing load capacity is considered:

```
%% Design of footing's transversal dimensions
% Note: in case it is required to design transversal dimensions
[be,le,contact_pressure]=designDimFootings(pu,qu,dimCol,hefoot,rec,...
                        typeFoot)

dim_zap=[be le];
```

Finally, the optimization function can be now called, thus:

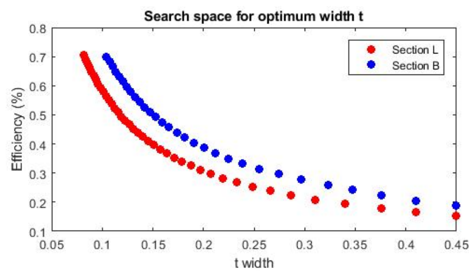
```
%% Optimization design process
[hmodif,m_max_eje,barDispositionFootings,arrangement_bar_footings,...
nbars_footings,AcBar,bestCost_elem,list_ef_footings,list_mr_footings]=...
isrFootings(pu_steel_footings,hefoot,dim_zap(1),dim_zap(2),...
rec,fc,fy,load_cols,dimCol,RebarAvailable,cols_sym_asym_isr,...
ductility,optimConv,PlotRebarDesign,typeFoot,sepMinRebar,wac);
```

Additionally, the design results can be exported as .csv files to be read by Visual CALRECOD's functions an visualize in a 3D space the structural design. For this purpose:

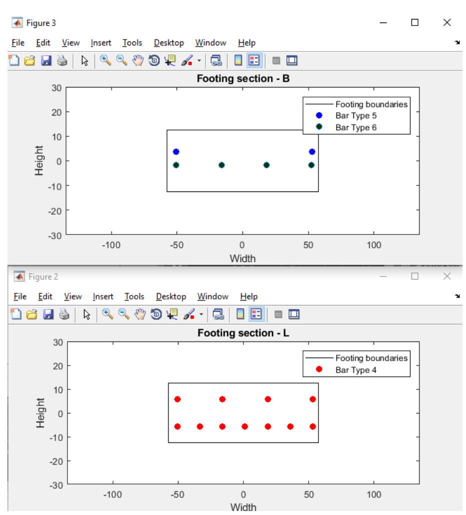
```
%% Exporting design results for visualization
dimensionFootingCollection=[be,le,hmodif,rec]
coordBaseFooting=[0,0,0]; % global position of the footing

directionData='C:\Users\luizv\OneDrive\DynamoRevit\Dynamo_visualization\';
ExportResultsIsolFootings(directionData,barDispositionFootings,...
    dimensionFootingCollection,nbars_footings,arrangement_bar_footings,...
    coordBaseFooting,cols_sym_asym_isr)
```

Results

Usually for isolated footings, the designed thickness to withstand shear forces provides a great resistance contribution against the pure flexure forces for each transversal cross-section, therefore, the rebar quantities tend to be very close to minimum ones established by code, as it can be observed in the following convergence plot for the optimal ISR's width in tension, where the initial reinforcing area quantity for both cross-sections was reduced almost 7 times to reach an acceptable efficiency of 0.7. This is one of the reasons of why the min separation restriction tends to be higher than for beams or columns.



The following results correspond to the cross-section layouts.



11.24 Ex24: Optimal design of rebar for a circular column

Design_Rebar_Circular_Columns_Ex01

Problem

When dealing with circular columns symmetrical rebar designs of only one rebar diameter are the most common ones. The main difficulty for these sort of elements when analysing their resistance is to determine the contribution that the concrete zone in compression provides for the overall resistance for a given neutral axis depth value.

Solution

With the functions that CALRECOD offers, however, not only the assessment of structural resistance of a column can be done but also to determine optimal rebar designs against a given set of load conditions.

Let us consider for instance the following circular column element with the next geometry and material parameters:

```
%% GEOMETRY
height=400; %column's length (cm)
diam=50; % cross-section diameter (cm)
rec=4; % concrete cover (cm)

%% MATERIAL
fy=4200; % yield stress of rebars
E=fy/0.0021; % Modulus of Elasticity of the reinforcing steel (Kg/cm2)
fc=280; % concrete's compressive strength (Kg/cm2)
wac=7.8e-3; % unit volume weight of reinforcing steel (Kg/cm2)
```

Additional parameters and variables must be set for a proper assessment, such as the unit cost of rebar assembly for each rebar diameter available and the ductility demand level:

```
%% ADITIONAL PARAMETERS
cols_rebar_isr="Rebar";
puColsISR=[28.93];

% symmetrical rebar
puColsRebar=[29.19, 29.06, 28.93, 28.93, 28.93, 28.93, 28.93];

% Plot options:
optimaConvPlot=1; % optima reinforcement area convergence plot
plotsISRdiagrams=1; % interaction diagrams

% Ductility demand
ductility=3;
```

The biaxial bending-compression loads for these elements is just set as shown below given the symmetry of the cross-section with respect to any of its axis:

```
%% Loads
load_conditions=[1 -15e3 32e5]; % [nload, Pu, Mu] (Kg-cm)
```

The commercially available rebar diameters to choose from must be established. For this purpose, the following arrays is set:

```
%% Rebar data
% Commerically available rebars
barsAvailable=[4 4/8*2.54;
                5 5/8*2.54;
                6 6/8*2.54;
                8 8/8*2.54;
                9 9/8*2.54;
                10 10/8*2.54;
                12 12/8*2.54];
```

Now, the ISR optimization process is first initiated to determine an approximate min reinforcement area through the ISR's width:

```
%% ISR OPTIMIZATION
[cost_elem_col,act,Ef_sec_col,MrtCol,t_value,c]=isrCircCols(puColsISR,...
height,wac,diam,rec,fy,fc,load_conditions,ductility,optimaConvPlot,...
plotsISRdiagrams);
```

Finally, the rebar optimization function can be now called, thus:

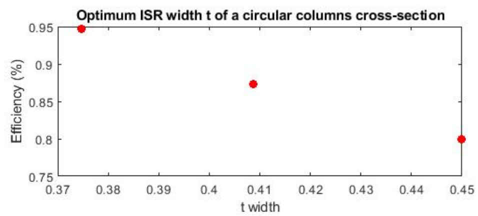
```
%% REBAR OPTIMIZATION (OPTIONAL)
if cols_rebar_isr=="Rebar"

    plotRebarDesign=1; % To visualize the rebar design layout plot
    npdiag=40;
    [MrColRebar,Inertia,bestArea,bestCost,bestdiagram,bestnv,bestobar,...
    bestEf,bestArrangement,bestDisposition,bestc]=optimalRebarCirc...
    (diam,rec,act,E,npdiag,0.85*fc,puColsRebar,load_conditions,...
    barsAvailable,wac,height,plotRebarDesign);
end
```

Additionally, the design results can be exported as .csv files to be read by Visual CALRECOD's functions an visualize in a 3D space the structural design. For this purpose:

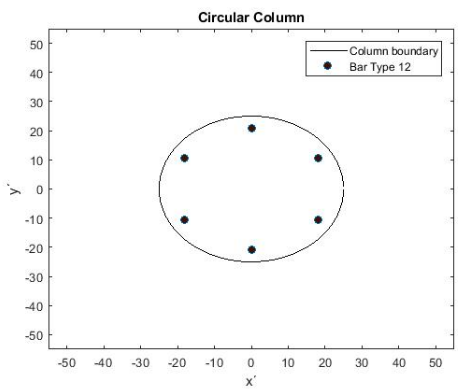
```
%% EXPORT RESULTS
coordBaseCols=[0 0 0]; % global location of the column base [x,y,z]
dimColumnsCollection=[diam height rec];
nbarColumnsCollection=length(bestDisposition(:,1));
directionData='C:\Users\luizv\OneDrive\DynamoRevit\Dynamo_visualization\';

ExportResultsColumnCirc(directionData,dimColumnsCollection,bestDisposition,...
    nbarColumnsCollection,bestArrangement,cols_rebar_isr,coordBaseCols)
```

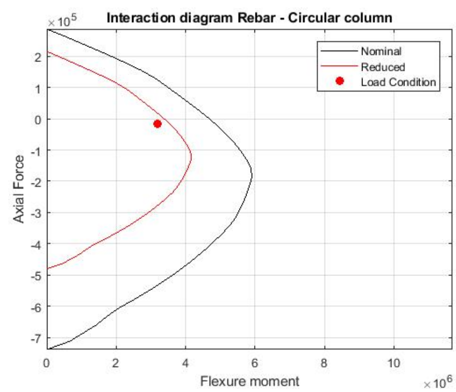
Results

As shown below only three iterations were enough for the ISR's width to reach an acceptable resistance efficiency from the given initial one by default of 0.5.



Once such ISR's min area was determined a linear transformation to rebar area took place, from which the following results were obtained:





11.25 Ex25: Modified cross-section inertia of a symmetrically reinforced column

Ecc_Inertia_CrackingColSym_Ex01

Problem

When performing structural analysis of reinforced concrete frames it is recommended by codes to consider cracking mechanisms for columns and beams, mainly, so that their gross cross-section momentum of inertia may be decreased. To simplify the assessment of the inertia reduction of reinforced concrete columns some codes recommend a reduction factor of between 0.6 to 0.7, however, such reduction factor may vary from geometry to geometry, material, amount of reinforcing steel and load eccentricities.

There is a load eccentricity limit that indicates when to consider a cross-section as cracked or non-cracked (see p. 28). A cracked cross-section takes on account the concrete zone in tension that does not work nor provides any contribution in resistance to the structural element. However, to compute such mechanism for columns is not that easy as for beams, and therefore, designers have a tendency to apply directly the recommended reduction factors by codes without making any further studies of how such reduction or modification of a column's inertias may vary as some parameters change. Nevertheless, to be able to perform those studies could be extremely important for some project so that better decisions can be made regarding the amount of steel reinforcement to provide for certain columns or max permissible bending-compression loads.

Solution

With the function *CrackingColumnsSym* and *CrackingColumnsAsym* that CALRECOD offers it is possible to determine relatively accurately the reduction or modification factor of inertia for symmetrically reinforced and asymmetrically reinforced rectangular concrete columns, respectively, subject to biaxial bending-compression forces. Such functions operate based on the *transformed section* method (see p. 28)and they turn out quite easy to deploy.

Let us consider for instance the following rectangular column element with the next geometry and material parameters:

```
%% Geometry
h=80; % cross-section's height dimension (cm)
b=60; % cross-section's width dimension (cm)
rec=[4,4]; % concrete cover for each cross-section direction (cm)
height=300; % column's length (cm)
dimensionsColumn=[b,h];

%% Materials
fc=300; % concrete's compressive strength (Kg/cm2)
fdpc=0.85*fc;
betac=1.05-fc/1400;
if betac<0.65
    betac=0.65;
elseif betac>0.85
    betac=0.85;
end
fy=4200; % yield stress of the reinforcing steel (Kg/cm2)
E=2.1e6; % Modulus of Elasticity of the reinforcing steel (Kg/cm2)
```

Additional parameters and variables must be set for a proper assessment, such as a string variable that indicates that a cracked cross-section mechanism should be consider if necessary:

```
%% Additional parameters
conditionCrack="Cracked";
```

The cracking mechanism computed in functions *CrackingColumnsSym* and *CrackingColumnsAsym* are based on an idealization of the reinforcing steel, therefore, the ISR's width should be provided to properly use those functions:

```
%% Reinforcement data
% ISR's width to propose a reinforcement area
twidthISR=0.45; % cm
```

From such given ISR's with, the amount of proposed reinforcement area can be then computed for extra information to compare it, for example, with the max and min amounts given by codes, as:

```
%% Additional computations
Aisr=(2*(b-rec(1))+2*(h-rec(2)))*twidthISR % proposed reinforcement area
Amin=0.01*b*h % Min reinforcement area by code
Amax=0.04*b*h % Max reinforcement area by code

Inertia0=[b*h^3/12,h*b^3/12]; % gross section's inertia
```

Note in the previous set of code that the gross cross-section inertia is computed, so that later the modification of such cracked or non-cracked cross-section inertia can be compared with this one.

A compression load should be given, so that along with the load eccentricities, the bending moments can be later computed:

```
%% Loads
Pu=-105.3e3; % Compression load (Kg)
```

Now, a loop will be created to compute the modified inertias as the load eccentricities increase, however, before calling the required functions to determine such modified inertias it is necessary to determine the neutral axis depth values corresponding to the given bending-compression loads, for both cross-section directions. For this purpose, the function *widthEfficiencyCols* is used. The following piece of code describes this explanation:

```
%% Main loop
IxyredVec=[];
eccxyVec=[];
maxEc=80; % Max load eccentricity for both section's axis
ecxy=0; % initial load eccentricity
while ecxy<=maxEc
    ecxy=ecxy+0.1;
    eccentricityXY=[ecxy,ecxy]; % Load eccentricities for each
```

```
                             % section's axis (cm)
    eccxyVec=[eccxyVec; eccentricityXY]; % collecting eccentricities
                                         % for plot
    Mux=abs(Pu*eccentricityXY(1)); % bending loads Mu = P * e
    Muy=abs(Pu*eccentricityXY(2)); % Kg-cm

    conditions=[1 Pu Mux Muy]; % Kg-cm

    %% Structural efficiency
    [Eft,diagramaInteraccion,tablaEficiencias,cxy]=widthEfficiencyCols...
       (twidthISR,dimensionsColumn,rec,fy,60,conditions,fdpc,Es,betac);

    %% Modified momentum of inertia for both axis directions
    % cracking mechanisms are considered through the
    % "transform section" method

    % Take parameter [cxy] from the "widthEfficiencyCols" function
    [InertiaXYmodif,Atransfxy,elimxy]=CrackingColumnsSym(h,b,fdpc,rec,...
             twidthISR,eccentricityXY,twidthISR,Pu,cxy,conditionCrack,E);

    %% Computation of the inertia reduction degree
    redIxy=InertiaXYmodif./Inertia0;
    IxyredVec=[IxyredVec; redIxy];
end
```

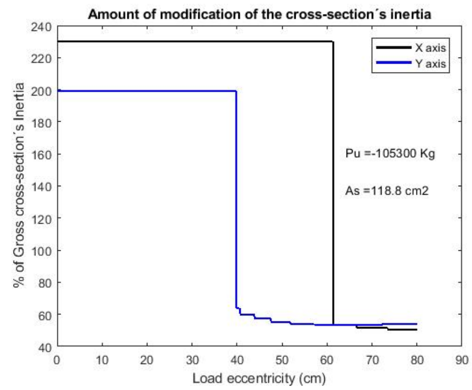
As it can be observed in the previous set of code, the max load eccentricity set is 80 cm starting from 0.0 cm. In every iteration the load eccentricity for both cross-section axis direction will be increased 0.1 cm.

Once the computations have been performed and collected in their respective vectors, the plots of such inertias variation can be computed as:

```
%% Plotting results
figure(1)
plot(eccxyVec(:,1), IxyredVec(:,1)*100,'k -','LineWidth',1.8)
title('Amount of modification of the cross-section´s inertia')
ylabel('% of Gross cross-section´s Inertia')
xlabel('Load eccentricity (cm)')
legend('X axis')
hold on
plot(eccxyVec(:,2), IxyredVec(:,2)*100,'b -','LineWidth',1.8,...
    'DisplayName','Y axis')
hold on
% Adding labels
PuText=num2str(Pu);
PText=strcat('Pu = ',PuText,' Kg');
text(maxEc*0.8,max(IxyredVec(:,1))*100*0.7,PText)
AsText1=num2str(Aisr);
AsText=strcat('As = ',AsText1,' cm2');
text(maxEc*0.8,max(IxyredVec(:,1))*100*0.6,AsText)
```

Results

Note in the following resulting plots that initially the cross-section inertia is approximately 2 times higher than the gross cross-section inertia for both axis directions, given that in this range of load eccentricities the cross-section is not yet cracked and the contributions of the reinforcing steel in the whole inertias are already being considered.



However, after their respective limits of load eccentricity for both axis directions in which the cross-sections are considered as cracked, their inertia are reduced in about a 60% of the gross cross-section inertia, and continue with a slight decreasing rate as the load eccentricities keep on increasing, until such decrements stabilize in about the max load eccentricity of 80cm. After seeing this plot, it is not that surprising that many design codes recommend such inertia reduction factors of between 0.6 to 0.7 for columns.

11.26 Ex26: Cost-Dim curve of rectangular concrete beams

CostDim_Curve_Design_Beams_Ex01

Problem

When designing a rectangular beam element it may be of interest to determine the optimal set of cross-section dimensions so that the min construction cost or min usage of materials can be reached.

Solution

For this purpose, the function *CostDimCurveOptimDesignBeam* that CALRECOD offers could be used. The output of such function provides a great insight of how the optimal construction costs (concrete costs, reinforcement costs and the sum of both) vary as the cross-section width dimension increases.

Let us consider for instance the following rectangular beam element with the next geometry and material parameters:

```
%% Geometry
span=500; % beam's length (cm)

h_rec=[4]; % vertical concrete cover
b_rec=3; % lateral concrete cover

%% Material
fc=280; % Concrete compressive strength Kg/cm2
fy=4200; % Yield stress of steel reinforcement (Kg/cm2)
wac=7.8e-3; % unit volume weight of the reinforcing steel (Kg/cm3)
```

When inserting the pure bending load conditions, the three main cross-sections must be considered for a more accurate assessment of the construction costs. Thus:

```
%% Load conditions
load_conditions=[1 -23.0e5 16.0e5 -20.0e5]; % Kg-cm
```

Additional parameters and variables must be set, such as the average unit construction cost of reinforcement assembly and ductility demand level. Also the concrete cost per unit volume for the required f'c must also be given. One option is to provide the whole database for better automation as the f'c varies, that is up to you:

```
%% Additional data
pu_beams_steel=38.6; % unit assembly cost of steel reinforcement
duct=1; % high ductility demand

             %f'c % Bombed %Direct Shot (per unit volume)
cost_concrete=[100 2281.22e-6;
               150 2401.22e-6;
               200 2532.14e-6;
               250 2777.96e-6;
```

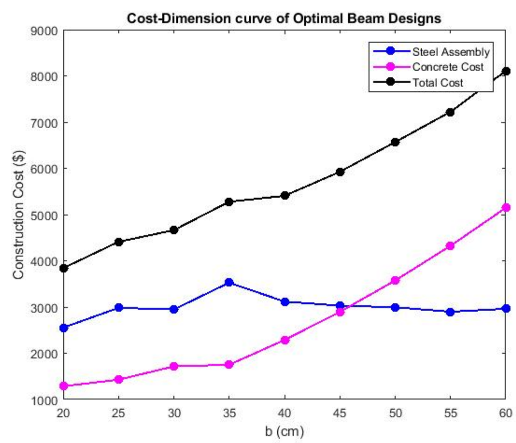
```
            300 2939.12e-6;
            350 3111.50e-6;
            400 3298.16e-6;
            450 3499.10e-6];
```

Finally, the main function is called:

```
%% Cost-Dim curve design
[collectionDimBeams,collectionISRareaBeams,collectionISRbeams,...
    collectionEffBeams]=CostDimCurveOptimDesignBeam(span,wac,fc,b_rec,...
    h_rec,duct,pu_beams_steel,unit_cost_conc_beams,load_conditions,fy);
```

Results

Note in the following resulting plots that the corresponding optimal cross-section height dimensions for each width dimension are not depicted as they can be seen in the function output vector called *collectionDimBeams*.



As it can be seen in the plot, the optimal cross-section width dimension for this given set of load conditions is the minimum one of $25cm$ with a corresponding optimal height dimension of $45cm$.

11.27 Ex27: Cost-Dim curve of rectangular concrete columns

CostDim_Curve_Design_Columns_Ex01

Problem

Similar as for rectangular beams, when designing a rectangular columns it may be of interest to determine the optimal set of cross-section dimensions so that the min construction cost or min usage of materials can be reached.

Solution

For this purpose, the function *CostDimCurveOptimDesignCols* that CALRECOD offers could be used. The output of such function provides a great insight of how the optimal construction costs (concrete costs, reinforcement costs and the sum of both) vary as the cross-section width dimension increases.

Let us consider for instance the following rectangular column element with the next geometry and material parameters:

```
%% GEOMETRY
height=400; %cm
rec=[4 4]; % [coverx covery] (cm)

%% MATERIALS
fc=280; % Kg/cm2
fy=4200; % Yield stress of steel reinforcement (Kg/cm2)
wac=7.8e-3; % unit volume weight of the reinforcing steel (Kg/cm3)
```

The following biaxial bending-compression loads are considered:

```
%% LOAD CONDITIONS
load_conditions=[1 15e3 32e5 23e5]; % [nload, Pu, Mx, My] (Kg-cm)
```

Additional parameters and variables must be set, such as the average unit construction cost of reinforcement assembly and ductility demand level. Also the concrete cost per unit volume for the required f'c must also be given. One option is to provide the whole database for better automation as the f'c varies, that is up to you:

```
%% Additional data
pu_cols=[28.93];

% Ductility demand
ductility=3;

              %f'c % Bombed %Direct Shot (per unit volume)
cost_concrete=[100 2281.22e-6 2266.98e-6;
                150 2401.22e-6 2390.88e-6;
                200 2532.14e-6 2525.28e-6;
                250 2777.96e-6 2845.00e-6;
                300 2939.12e-6 3010.90e-6;
```

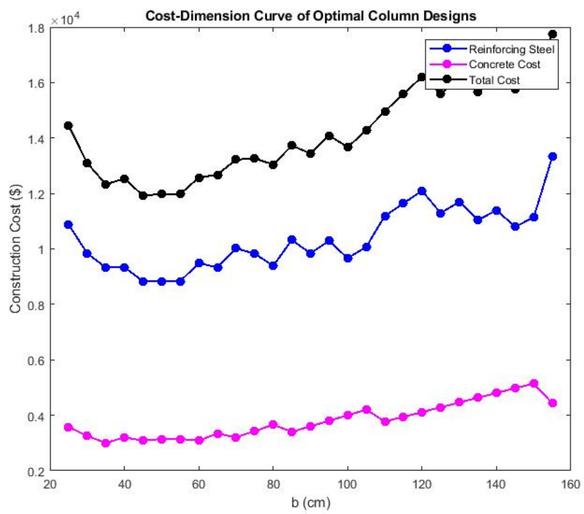
```
            350 3111.50e-6 3188.50e-6;
            400 3298.16e-6 3380.50e-6;
            450 3499.10e-6 3587.35e-6];
```

Finally, the main function is called:

```
%% Cost-dimension curves
[collectionDimCols,collectionISRareaCols,collectionISRcols,...
collectionEffCols]=CostDimCurveOptimDesignCols(height,wac,fc,rec,...
ductility,pu_cols,unit_cost_conc_cols,load_conditions,fy);
```

Results

Note in the following resulting plots that the corresponding optimal cross-section height dimensions for each width dimension are not depicted as they can be seen in the function output vector called *collectionDimCols*.



As it can be seen in the plot, the optimal cross-section width dimension for this given set of load conditions is the one of $45cm$ with a corresponding optimal height dimension of $60cm$, whose ratio b/h is very close to the bending loads ratio My/Mx.

11.28 Ex28: Optimal design of reinforcement of a concrete plane frame subject to seismic forces

$Design\_RCPlane\_Frames\_Ex01$

Problem

Similar as for individual beams, columns or footings, for reinforced concrete frame systems it is usually required to optimally design the reinforcement of each of its elements. Even though other design criteria should be considered for this purpose such as the *Strong Column - Weak Beam* mechanism or even performance-based design criteria, it might be very useful to be able to optimize the reinforcement for each and every one of its elements separately once their cross-section dimensions have been proposed. This output could provide with a great insight of the whole weight of the structure, the potential construction cost, material volumes or the more likely dimensions of each of its elements.

Solution

CALRECOD offers the **function** $DesignRCPlaneFrameBCI$ for this such purpose, which includes the optimization design functions $beamsISR$, $isrColumnsSymAsym$ and $isrFootings$, for rectangular beams, rectangular columns and rectangular isolated footings, respectively.

When dealing with whole structural systems to input the geometric and material data may not be a simple as for individual elements given that for this case the whole topology data must also be given. Let us consider for instance, the following plane frame data, starting with by labelling each structural element as a beam or column for further reference in the design:

```
nnodes=8; % number of nodes of the plane frame
nbars=8; % number of elements of the plane frame

%% Materials and mechanical properties
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"];

% f'c for each element type:
fcbeams=250;
fccols=250;
fc_footing=250;

% To detect which 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

for i=1:nbars
    if type_elem(i,2)=="Col"
        fpc(i,1)=fccols;
    elseif type_elem(i,2)=="Beam"
        fpc(i,1)=fcbeams;
    end
end

% 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
```

Additional geometrical parameters must be given, such as the cross-section dimensions of each plane frame's element and the node coordinates:

```
%% Geometry
dimensions=[30 30; % cross-section dimensions of each element
            30 30;
            25 50;
            30 45;
            30 45;
            30 60;
            25 50;
            30 30];

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];

% Length of each element
for i=1:nbars
    lenElem(i,1)=((coordxy(nf(i),1)-coordxy(ni(i),1))^2+(coordxy(nf(i),2)...
        -coordxy(ni(i),2))^2)^0.5;
end
```

Now, given that the design results will be exported as .csv files to be read by Visual CALRECOD the location coordinates of all the elements must be input, including the location of the isolated footings:

```
%% Location of elements in the 3D system of reference of Visual-CALRECOD
%% Plane Y-Z
% coordinates of the columns base' centroids:
coordBaseCols=[0 0 -100;
              0 0 400;
              0 600 400;
              0 600 -100;
              0 1200 -100];

% coordinates of the footing base' centroids:

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

% coordinates of the beams' starting nodes:
coordEndBeams=[0 0 800;
               0 0 400;
               0 600 400];
```

In order to be able to perform structural analysis the topology data must be also set:

```
%% Topology
% Initial-final node of each bar
ni=[1;2;3;4;5;2;5;7];
nf=[2;3;4;5;6;5;7;8];

% Topology conectivity matrix
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
```

As for any structural analysis, the boundary conditions for any prescribed DOF should be given too. This is done through an array of two columns called *bc* in which in the first column the DOF are listed and in the second column the prescribed values for those DOF:

```
%% Boundary conditions: prescribed dof
bc=[1 0;
    2 0;
    3 0;
    16 0;
    17 0;
    18 0;
    22 0;
    23 0;
    24 0];
```

Now, moving further more into the design process, the following parameters must be set:

```
%% Additional design and analysis parameters
np=7; % number of points of analysis for the computation of mechanic
       % elements for each structural element with the FEM

qadm=2.5; % Admissible bearing load of soil (Kg/cm2)
FS=1.5; % Design Safety Factor for isolated footings
nodes_support_column=[1 6 8; % support (dof)
                      1 5 8]; % column element number

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

ductility=3; % required ductility on the element's cross-sections for the
             % design of reinforcement

recxy_cols=[4 4]; % concrete cover

%% Rebar data
% Commerically available rebars
RebarAvailable=[4 4/8*2.54;
                5 5/8*2.54;
                6 6/8*2.54;
                8 8/8*2.54;
                9 9/8*2.54;
                10 10/8*2.54;
                12 12/8*2.54];
```

Coming back again the structural analysis, the uniformly distributed loads on the beams and the unit volume weight of the concrete must be given. Also, in order to consider the lateral seismic loads and the self-weight of the columns as external loads, the DOF data for these loads must be given:

```
%% Loads
beams_LL=[1 75; % Live and Dead Loads on beams as distributed forces
          2 77;
          3 71];

% Unit weight of concrete - To consider self weight
wco=0.0024;
unit_weight_elem=zeros(nbars,2);
for i=1:nbars
    unit_weight_elem(i,2)=wco; % kg/cm3
end

% To consider the self weight load as a distributed load in beams
qbary=zeros(nbars,2);
for i=1:nbeams
    qbary(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
dofWeightElemCols=[2 5 14 17 23; % dof
                   1 2 4  5  8]; % cols

fglobal=zeros(3*nnodes,1);
for i=1:length(dofWeightElemCols(1,:))
    fglobal(dofWeightElemCols(1,i))=-1.1*areaElem(dofWeightElemCols(2,i))*...
    unit_weight_elem(dofWeightElemCols(2,i),2)*lenElem(dofWeightElemCols(2,i));
end

% To consider lateral forces (seismic or impacto, etc.)
dof_seismic_forces=[4 7]; % in case there are equivalent sesimic loads
                          % applied
```

Now, regarding the seismic loads, these must be acting in both directions of the in-plane system of reference (left and right) so that a structural analysis is executed for each direction to extract design mechanical elements for each column and beam. For this purpose, the **function** *PlaneFrameStaticLinearAnalysis* is used:

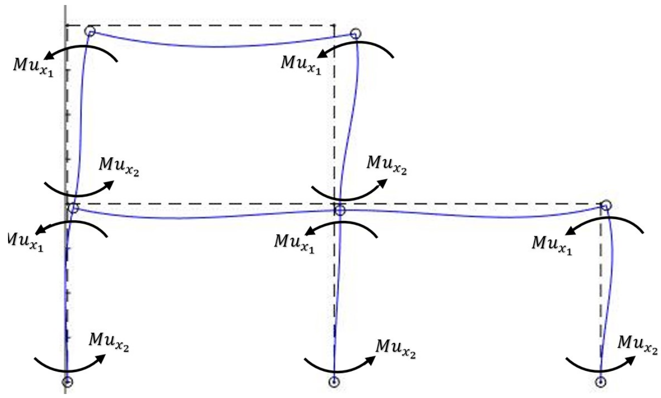
```
%% Structural Analysis taking all loads in consideration.
% Lateral forces in the right direction (F-Right):
Vx1=3500;
Vx2=4500;
fglobal(dof_seismic_forces)=[Vx1;Vx2];

eRefN=5;eRefV=3;eRefM=3; % these are the element's number to be
                         % taken for reference when ploting the mechanical
                         % elements for each bar
[displacements_right,r_right,Ex_right,Ey_right,es_bars_normal_right,...
es_bars_shear_right,es_bars_moment_right]=PlaneFrameStaticLinearAnalysis(nnodes,...
nbars,Eelem,areaElem,inertiaElem,bc,fglobal,ni,nf,qbary,Edof,np,coordxy,0,...
```

```
eRefN,eRefV,eRefM);

% Lateral forces in the left direction (F-Left):
Vx1=-3500;
Vx2=-4500;
fglobal(dof_seismic_forces)=[Vx1;Vx2];

[displacements_left,r_left,Ex_left,Ey_left,es_bars_normal_left,...
es_bars_shear_left,es_bars_moment_left]=PlaneFrameStaticLinearAnalysis(nnodes,...
nbars,Eelem,areaElem,inertiaElem,bc,fglobal,ni,nf,qbary,Edof,np,coordxy,0,...
eRefN,eRefV,eRefM);
```

Whereas for the extraction of design mechanical elements, let us consider first those for the beams. For these elements the max absolute bending moments for each beams' cross-sections from both structural analysis (left and right should be considered). Thus:

```
%% Extracting design forces for each element
% -----------------------------------------------------------------------
% Design loads on beams from the structural analysis
% -----------------------------------------------------------------------
for j=1:nbeams
    nelem=elem_beams(j);
    for i=1:np
        if abs(es_bars_normal_right(i,nelem))<abs(es_bars_normal_left(i,nelem))
            es_bars_normal_right(i,nelem)=es_bars_normal_left(i,nelem);

        end
        if abs(es_bars_shear_right(i,nelem))<abs(es_bars_shear_left(i,nelem))
            es_bars_shear_right(i,nelem)=es_bars_shear_left(i,nelem);

        end
        if abs(es_bars_moment_right(i,nelem))<abs(es_bars_moment_left(i,nelem))
            es_bars_moment_right(i,nelem)=es_bars_moment_left(i,nelem);

        end
    end

    shear_beams(j,:)=es_bars_shear_right(:,nelem);
    load_conditions_beams(j,1)=es_bars_normal_right(1,nelem);
    load_conditions_beams(j,2)=es_bars_moment_right(1,nelem);
    load_conditions_beams(j,3)=max(abs(es_bars_moment_right(2:6,nelem)));
    load_conditions_beams(j,4)=es_bars_moment_right(7,nelem);
end
```

Whereas for the columns both structural analysis' results should be considered simultaneously, that is, for each column element four different bending moments are taken for design along with their respective axial compression force. Note in the following piece of code that no bending moment is considered in the out-of-plane direction Mu_y, however, given the deformation shape of each column the acting bending moments at both their ends are considered given that they have opposite acting directions over the column cross-sections. The following figure depicts this phenomenon:




```
% ----------------------------------------------------------------------
% Design loads on columns from the structural analysis
% ----------------------------------------------------------------------
emin=0.000001; % no load is considered on the out-of-plane direction

for i=1:ncols
    nelem=elem_cols(i);

    % Design loads of seismic forces to the right ---------------------
    % Load at one column's end
    load_conditions_columns(4*i-3,1)=4*i-3;

    axial=es_bars_normal_right(1,nelem);
    load_conditions_columns(4*i-3,2)=axial;

    [Mux1]=es_bars_moment_right(1,elem_cols(i));
    Mminy1=axial*emin;

    load_conditions_columns(4*i-3,3)=Mux1; % Mux (in-plane)
    load_conditions_columns(4*i-3,4)=Mminy1; % Muy (out-of-plane)

    % Load at the other column's end
    load_conditions_columns(4*i-2,1)=4*i-2;

    axial=es_bars_normal_right(np,nelem);
    load_conditions_columns(4*i-2,2)=axial;

    [Mux2]=es_bars_moment_right(np,elem_cols(i));
    Mminy2=axial*emin;

    load_conditions_columns(4*i-2,3)=Mux2; % Mux (in-plane)
    load_conditions_columns(4*i-2,4)=Mminy2; % Muy (out-of-plane)

    % Design loads of seismic forces to the left ----------------------
    % Load at one column's end
    load_conditions_columns(4*i-1,1)=4*i-1;
```

```
    axial=es_bars_normal_left(1,nelem);
    load_conditions_columns(4*i-1,2)=axial;

    [Mux3]=es_bars_moment_left(1,elem_cols(i));
    Mminy3=axial*emin;

    load_conditions_columns(4*i-1,3)=Mux3; % Mux (in-plane)
    load_conditions_columns(4*i-1,4)=Mminy3; % Muy (out-of-plane)

    % Load at the other column's end
    load_conditions_columns(4*i,1)=4*i;

    axial=es_bars_normal_left(np,nelem);
    load_conditions_columns(4*i,2)=axial;

    [Mux4]=es_bars_moment_left(np,elem_cols(i));
    Mminy4=axial*emin;

    load_conditions_columns(4*i,3)=Mux4; % Mux (in-plane)
    load_conditions_columns(4*i,4)=Mminy4; % Muy (out-of-plane)
end
```

To extract the design loads of the isolated footings, on the other hand, the max absolute reactions of the DOF stated previously in the array *nodes_support_column* are considered. Therefore:

```
% ------------------------------------------------------------------
% Design loads on Footings from the structural analysis
% ------------------------------------------------------------------

for j=1:length(r_right)
    if abs(r_right(j))<abs(r_left(j))
        r_right(j)=r_left(j);
    end

end
reactions=r_right;
```

Finally, before calling the main optimization function it must be stated which type of reinforcement is desired to be designed in the columns, either symmetrical or asymmetrical. For each case the unit cost of rebar assembly should be given, as well as for beams and footings. In this example, the optimization design process with symmetrical rebar in the columns takes place, Thus:

```
%% Unit construction cost of rebar assembly for beams, columns and footings
pu_beams=38.85; % unit construction cost of reinforcement assembly
pu_steel_footings=26.75;

cols_sym_asym_isr="Symmetric"; % To choose which type of rebar design
                               % is required
if cols_sym_asym_isr=="Symmetric" || cols_sym_asym_isr=="Asymmetric"
```

```
    pu_cols=[29.19, 29.06, 28.93, 28.93, 28.93, 28.93, 28.93];

elseif  cols_sym_asym_isr=="ISR"
    pu_cols=[29.1];
end

%% Directory route to save the design results (if required)
% Note: if not required just set: directionData=[];
directionData=[];

%% Optimal design with symmetrical reinforcement in columns
[totalWeightStruc,wsteelColsTotal,pacColsElem,Mp,final_dimensions,...
unit_weight_elem,wsteelConcBeamsElem,wsteelConcColsElem,...
wsteelConcFootingsElem,hefootings,dimFoot,totalCostStruc,inertiaElem,...
wsteelStructure]=DesignRCPlaneFrameBCI(pu_beams,pu_cols,lenElem,fpc,...
inertiaElem,qadm,FS,nodes_support_column,pu_steel_footings,dimensions,...
fcbeams,fccols,fc_footing,areaElem,cols_sym_asym_isr,RebarAvailable,...
condition_cracking,ductility,elem_cols,elem_beams,recxy_cols,...
load_conditions_beams,load_conditions_columns,reactions,shear_beams,...
coordBaseCols,coordEndBeams,coordBaseFooting,directionData);
```

Note in the previous piece of code that the directionData variable that indicates the route in which the design data will be saved is empty, this would mean that for now such results are not to be saved:

Now, for the second optimization design process with asymmetrical rebar in the columns the following is done:

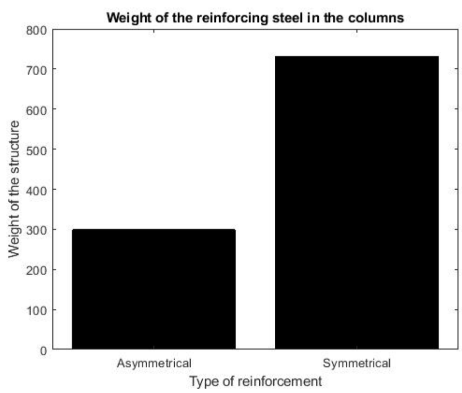
```
cols_sym_asym_isr="Asymmetric";
if cols_sym_asym_isr=="Symmetric"
    pu_cols=[29.19, 29.06, 28.93, 28.93, 28.93, 28.93, 28.93];
elseif cols_sym_asym_isr=="Asymmetric"
    pu_cols=[29.19, 29.06, 28.93, 28.93, 28.93, 28.93, 28.93];
elseif  cols_sym_asym_isr=="ISR"
    pu_cols=[28.93];
end

%% Optimal design with asymmetrical reinforcement in columns
directionData=[];
[totalWeightStruc,wsteelColsTotal,pacColsElem,...
Mp,final_dimensions,unit_weight_elem,wsteelConcBeamsElem,...
wsteelConcColsElem,wsteelConcFootingsElem,hefootings,dimFoot,...
totalCostStruc,inertiaElem,wsteelStructure]=DesignRCPlaneFrameBCI...
(pu_beams,pu_cols,lenElem,fpc,inertiaElem,qadm,FS,nodes_support_column,...
pu_steel_footings,dimensions,fcbeams,fccols,fc_footing,areaElem,...
cols_sym_asym_isr,RebarAvailable,condition_cracking,ductility,...
elem_cols,elem_beams,recxy_cols,load_conditions_beams,...
load_conditions_columns,reactions,shear_beams,coordBaseCols,coordEndBeams,coordBaseFooting,directionData);
```

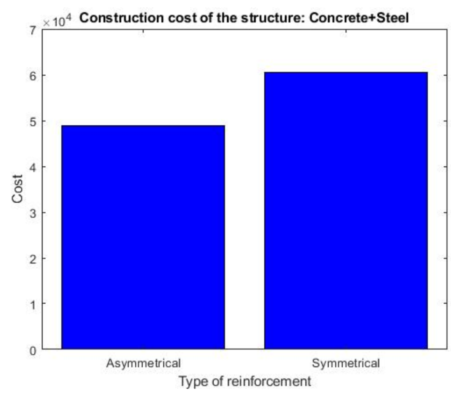
At the end, after both optimization design processes were finished, the total construction costs and the weight of steel rebar in the columns is saved to assess the savings that could be generated when designing asymmetrical rebar in the columns:

```
%% Bar diagrams to compare results
x=categorical({'Symmetrical','Asymmetrical'});
figure(10)
bar(x,y1,'black')
hold on
title('Weight of the reinforcing steel in the columns')
xlabel('Type of reinforcement')
ylabel('Weight of the structure')

figure(11)
bar(x,y2,'blue')
hold on
title('Construction cost of the structure: Concrete+Steel')
xlabel('Type of reinforcement')
ylabel('Cost')
```

Results

As it can be observed in the following plots, the amounts of rebar volumes that can be saved with asymmetrical reinforcement are huge, that so that the construction costs tend to be lower as well even though asymmetrical reinforcement is far more expensive per unit weight than symmetrical conventional designs.





12 References

References

[1] 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

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

[3] 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

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

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

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

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

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