COMPUTATIONAL PERFORMANCE OF AN EMBEDDED REINFORCEMENT MESH GENERATION METHOD FOR LARGE-SCALE RC SIMULATIONS

George Markou

Alhosn University, Department of Civil Engineering, P.O.Box 38772, Abu Dhabi, UAE.
E-mail: g.markou@alhosnu.ae

Keywords: Embedded Reinforcement, Mesh Generation, Large-Scale Models.

Abstract. In this paper, a numerical investigation on the limits of an automatic procedure for the generation of embedded steel reinforcement inside hexahedral finite elements is presented. In 3D detailed reinforced concrete simulations, mapping the reinforcement grid inside the concrete hexahedral finite elements is performed using the end-point coordinates of the rebar reinforcement macro-elements. This procedure is computationally demanding while in cases of large-scale models, the required computational time for the reinforcement mesh generation is excessive. This research work scopes to study and present the limitations of the embedded mesh generation method that was proposed by Markou and Papadrakakis, through the use of a 64-bit operating system. The embedded mesh generation method is integrated with a filtering algorithm in order to allocate and discard relatively short embedded rebar elements that result from the arbitrary positioning of the embedded rebar macro elements and the non-prismatic geometry of the hexahedral mesh. The computational robustness and efficiency of the integrated embedded mesh generation method are demonstrated through the analysis of three numerical models. The first two numerical models are a full-scale 2-storey and a 7-storey RC structures while the third model deals with a full-scale RC bridge with a trapezoidal section and a total span of 100m. Through the third numerical implementation, the computational capacity of the integrated embedded rebar mesh generation method is investigated.

1 Introduction

Modeling of reinforced concrete (RC) structures with the use of 3D detailed models is usually performed by research teams [1-5] or by large consultancy companies [6-7] that foresee the thorough investigation of the mechanical behavior of geometrically complicated RC structures. Researchers have been using this modeling type so as to verify experimental results or develop new constitutive models in an attempt to derive a numerically objective modeling method that will eventually provide the ability of performing assessment analysis for any type of RC structure design.

As it was described in [8], when modeling three-dimensional RC structures with the finite element (FE) method, three main approaches are available for the simulation of reinforcement: smeared, discrete, and embedded [9-11]. The smeared and discrete formulations have been found to be unsuitable for complicated reinforcement grid geometries thus undergo several restrictions when implemented. On the other hand, the embedded reinforcement formulation provides the ability of representing the grid's geometry in an exact manner without the need of modifying the actual arrangement of rebars to conform with the concrete FE mesh [12-13].

The approach of allocating and generating the embedded rebars inside hexahedral elements that was proposed by Barzegar and Maddipudi [9], which is an extension of the work of Elwi and Hrudey [10], has the advantage of allowing arbitrary positions for the rebars inside the concrete elements and a free geometry for each hexahedral element. In order to optimize the performance of the above embedded mesh generation method, Markou and Papadrakakis [8] proposed the introduction of a geometrical constraint in order to decrease the computational effort for generating the input data of the embedded rebar elements, particularly when dealing with relatively large-scale reinforced concrete models. The proposed method (Fig. 1) was incorporated in ReConAn FEA [5] which was developed and built to run in a 32-bit environment.

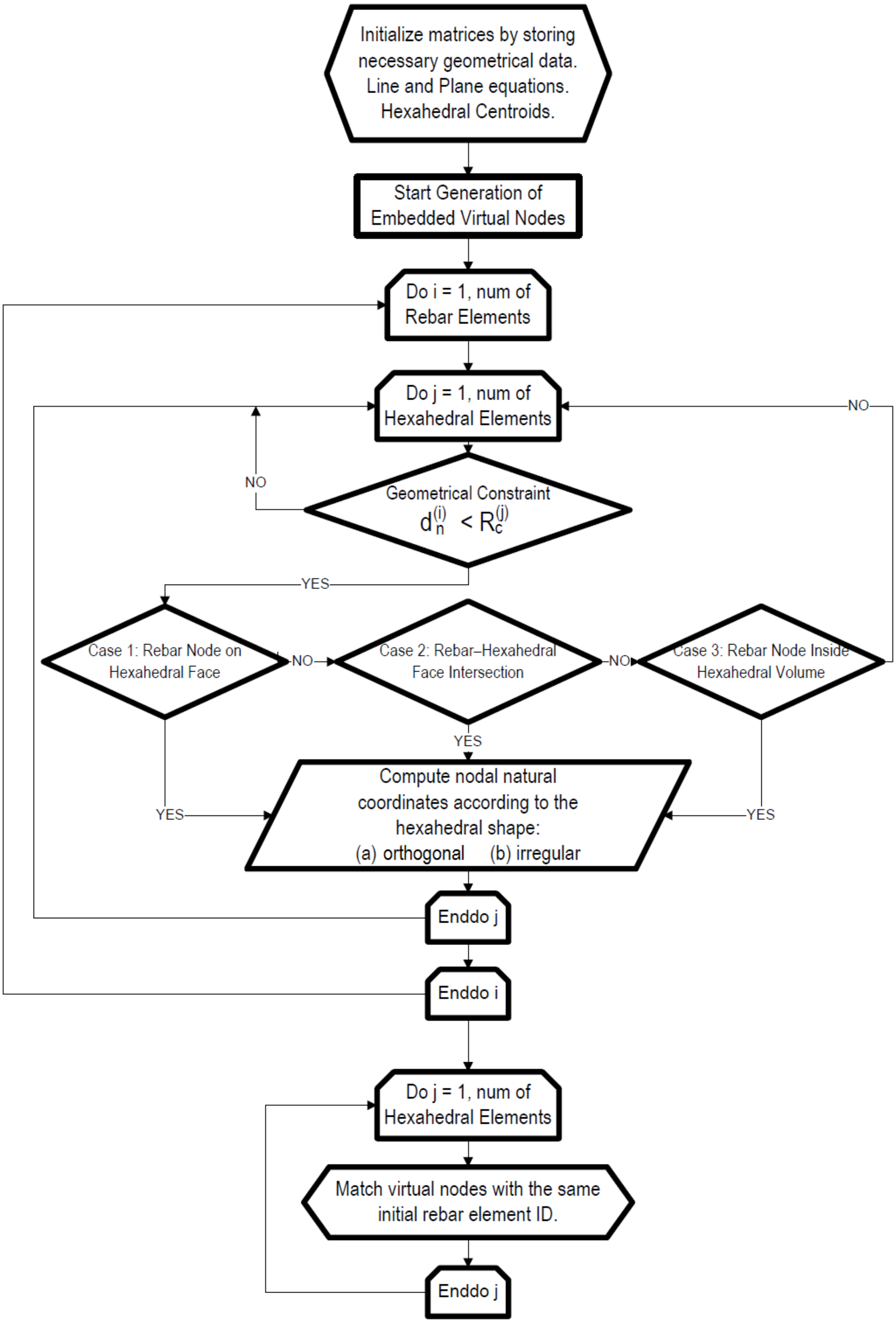

Initialize matrices by storing necessary geometrical data. Line and Plane equations. Hexahedral Centroids.
Start Generation of Embedded Virtual Nodes
Do i = 1, num of Rebar Elements
Do j = 1, num of Hexahedral Elements
NO
NO
Geometrical Constraint $d_n^{(i)} < R_c^{(j)}$
YES
Case 1: Rebar Node on Hexahedral Face
NO
Case 2: Rebar–Hexahedral Face Intersection
NO
Case 3: Rebar Node Inside Hexahedral Volume
YES
YES
YES
Compute nodal natural coordinates according to the hexahedral shape: (a) orthogonal (b) irregular
Enddo j
Enddo i
Do j = 1, num of Hexahedral Elements
Match virtual nodes with the same initial rebar element ID.
Enddo j

Figure 1. Flow chart of the embedded rebar element mesh generation method [8].

The purpose of this research work, is to investigate the numerical performance of the embedded mesh generation method proposed by Markou and Papadrakakis [8] for the case when dealing with large-scale numerical models, while derive the limitations of this method. So as to achieve this task, the Markou and Papadrakakis method was intergraded in a windows based 64-bit operating system with an Intel(R) Core i5-2500K (CPU 3.30 GHz), and was used to generate the rebar mesh of three different numerical models.

2 Generating reinforcement inside hexahedral elements

The method applied in this work, considers arbitrary positioning of the rebars inside the concrete elements, as shown in Fig. 2, which avoids a nonlinear search procedure for calculating the natural coordinates of the embedded reinforcement nodes inside the prismatic hexahedral elements. By implementing this separation scheme to the generation algorithm, the geometry of each hexahedral element is categorized (prismatic or non-prismatic) and accordingly treated in order to compute the natural coordinates of its containing embedded rebar elements [8].

Through a pre-processing software code, the initial mesh generation is performed consisting of the concrete elements and the rebar macro-elements defined by their end nodes. Then the coordinates of the macro-elements’ end-nodes are used to generate the numerical model of the embedded rebar elements inside each concrete hexahedral finite element. The generation of embedded rebar elements is performed for each rebar macro-element separately. This requires an independent search for each rebar macro-element, which is performed in order to detect all intersections of its straight part with the surrounding solid elements. In order to decrease the required computations of the embedded rebar generation procedure, a geometrical constraint is introduced, aiming at locating faster the embedded rebar elements inside the corresponding concrete hexahedral elements. More details about the generation of embedded rebar elements can be found in Appendix A.

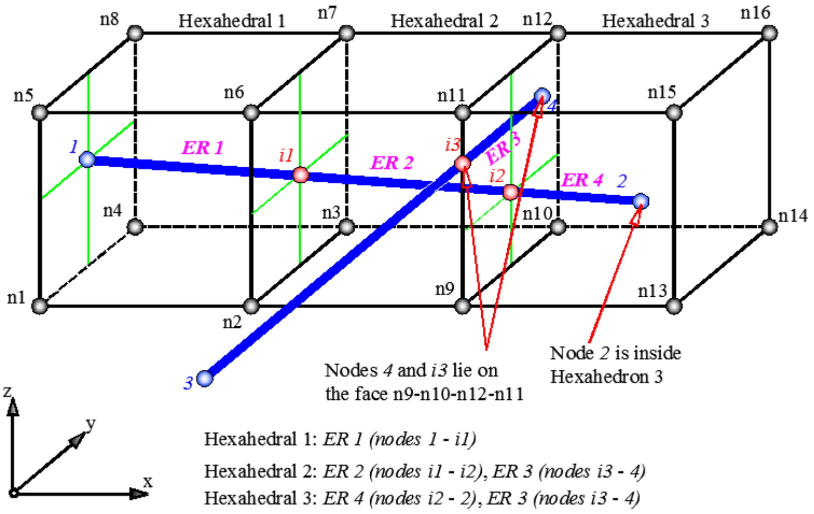

n8
Hexahedral 1
n7
Hexahedral 2
n12
Hexahedral 3
n16
n5
n6
n11
n15
4
1
ER 1
i1
ER 2
i3
ER 3
i2
ER 4
2
n4
n3
n10
n14
n1
n9
n13
n2
3
Nodes 4 and i3 lie on the face n9-n10-n12-n11
Node 2 is inside Hexahedron 3
z
y
x
Hexahedral 1: ER 1 (nodes 1 - i1)
Hexahedral 2: ER 2 (nodes i1 - i2), ER 3 (nodes i3 - 4)
Hexahedral 3: ER 4 (nodes i2 - 2), ER 3 (nodes i3 - 4)

Figure 2. Embedded rebar macro-elements inside hexahedral elements [8].

3 The 64-bit operating system environment

When dealing with large-scale numerical simulations the first problem that rises is the physical memory allocation which in the case of 32-bit architectures is limited to 2 Gb. This limitation of the 32-bit operating system restrains all software from allocating matrices (dynamic memory allocation) that have sizes larger than 2 Gb, while the maximum allowable RAM is usually 4 Gb. There are IA-32 processors that allow the use of a 36-bit physical memory address but are not commonly used.

The above limitation has as a result the use of parallel processing algorithms (cluster computing systems) or 64-bit operating systems that allow the software developers to build applications that can allocate files with sizes beyond 4 Gb. The major drawback of cluster computing systems is their size and eventually their cost given that for the case of the Civil Engineering profession, large-scale numerical computations are not very common.

In this research work, ReConAn FEA was integrated and built in a 64-bit operating system so as to acquire the ability of analyzing through the use of 3D detailed modeling [5] the mechanical behavior of RC structures in limit state loading conditions.

4 Numerical Implementation

In this section, a parametric investigation and a numerical verification of the efficiency of the integrated embedded reinforcement mesh generation method is presented. In addition to that, a large-scale numerical implementation will be generated and solved so as to illustrate the performance of the method and reveal its limitations. Achieving the above tasks, three different FE models have been used: a 2-storey RC structure, a 7-storey RC structure and a RC bridge with a total span of 100 m. All numerical tests were performed with a 64-bit operating system (3.3GHz processor).

4.1 RC 2-Storey Building

This numerical test was presented by Markou and Papadrakakis [8] and it was created in order to investigate the computational effort required by their proposed mesh generation method for the case of a 32-bit operating system. The geometry of the under-study RC building and the reinforcement details that were implemented in this numerical test are given in Figs. 3 & 4 and Tables 1 & 2. Fig. 5 shows the FE mesh of the RC building which consists of 4,382 hexahedral elements and 26,959 rebar macro-elements.

As it was presented in this research work [8], after the completion of the embedded rebar mesh generation procedure, the proposed method managed to allocate 51,064 embedded rebar elements in 69 seconds. The total required memory, so as for the software to generate the rebar elements, was 103 Mb.

a/a	*Dimensios (cm)*	*Rebars*		*Additional Left*		*Additional Right*		*Rebars for Torsion*	*Stirrups*		
		Up	*Down*	*Up*	*Down*	*Up*	*Down*		*Left*	*Middle*	*Right*
1	25x50	2 Ø14	3Ø14	2Ø18	-	-	-	-	Ø8/10	Ø8/10	Ø8/10
2	25x50	2 Ø14	3Ø14	-	-	2Ø18	-	-	Ø8/10	Ø8/10	Ø8/10
3	25x50	2 Ø14	3Ø16	2Ø18	-	-	-	2Ø12	Ø8/10	Ø8/10	Ø8/10
4	25x50	2 Ø14	3Ø14	-	-	-	-	2Ø12	Ø8/10	Ø8/10	Ø8/10
5	25x50	2 Ø14	3Ø14	-	-	2Ø18	-	2Ø12	Ø8/10	Ø8/10	Ø8/10
6	25x50	2 Ø14	3Ø14	2Ø18	-	2Ø18	-	-	Ø8/10	Ø8/10	Ø8/10
7	25x50	2 Ø14	3Ø14	2Ø20	-	2Ø20	-	2Ø12	Ø8/10	Ø8/10	Ø8/10
8	25x50	2 Ø14	3Ø14	2Ø18	-	-	-	2Ø12	Ø8/10	Ø8/10	Ø8/10
9	25x50	2 Ø14	3Ø14	-	-	2Ø18	-	-	Ø8/10	Ø8/10	Ø8/10
10	25x50	2 Ø14	3Ø14	2Ø18	-	2Ø18	-	2Ø12	Ø8/10	Ø8/10	Ø8/10
11	25x50	2 Ø14	3Ø14	2Ø18	-	2Ø18	-	2Ø12	Ø8/10	Ø8/10	Ø8/10
12	25x50	2 Ø16	3Ø16	3Ø20	-	-	-	-	Ø8/10	Ø8/10	Ø8/10
13	25x50	2 Ø14	3Ø14	-	-	2Ø18	-	2Ø12	Ø8/10	Ø8/10	Ø8/10
14	25x50	2 Ø14	3Ø14	2Ø18	-	-	-	2Ø12	Ø8/10	Ø8/10	Ø8/10
15	25x50	2 Ø14	3Ø14	-	-	2Ø18	-	2Ø12	Ø8/10	Ø8/10	Ø8/10
16	25x50	2 Ø14	3Ø14	2Ø18	-	2Ø20	1Ø14	2Ø12	Ø8/10	Ø8/10	Ø8/10

Table 1. 2-storey RC Building. Reinforcement details of the beams. Stories 1 and 2.

Footing ID	*Dimensions (cm)*	*Height (cm)*	*Rebars X-axis*	*Rebars Y-axis*
1	125x150	50	Ø12/15	Ø12/15
2	125x150	50	Ø12/15	Ø12/15
3	125x150	50	Ø12/15	Ø12/15
4	125x200	50	Ø12/15	Ø12/15
5	175x125	50	Ø12/15	Ø12/15
6	125x175	50	Ø12/15	Ø12/15
7	125x200	50	Ø12/15	Ø12/15
8	200x125	50	Ø12/15	Ø12/15
9	300x275	50	Ø12/15	Ø12/15
10	125x150	50	Ø12/15	Ø12/15
11	225x125	50	Ø12/15	Ø12/15

Table 2. 2-storey RC Building. Reinforcement details of the isolated footings.

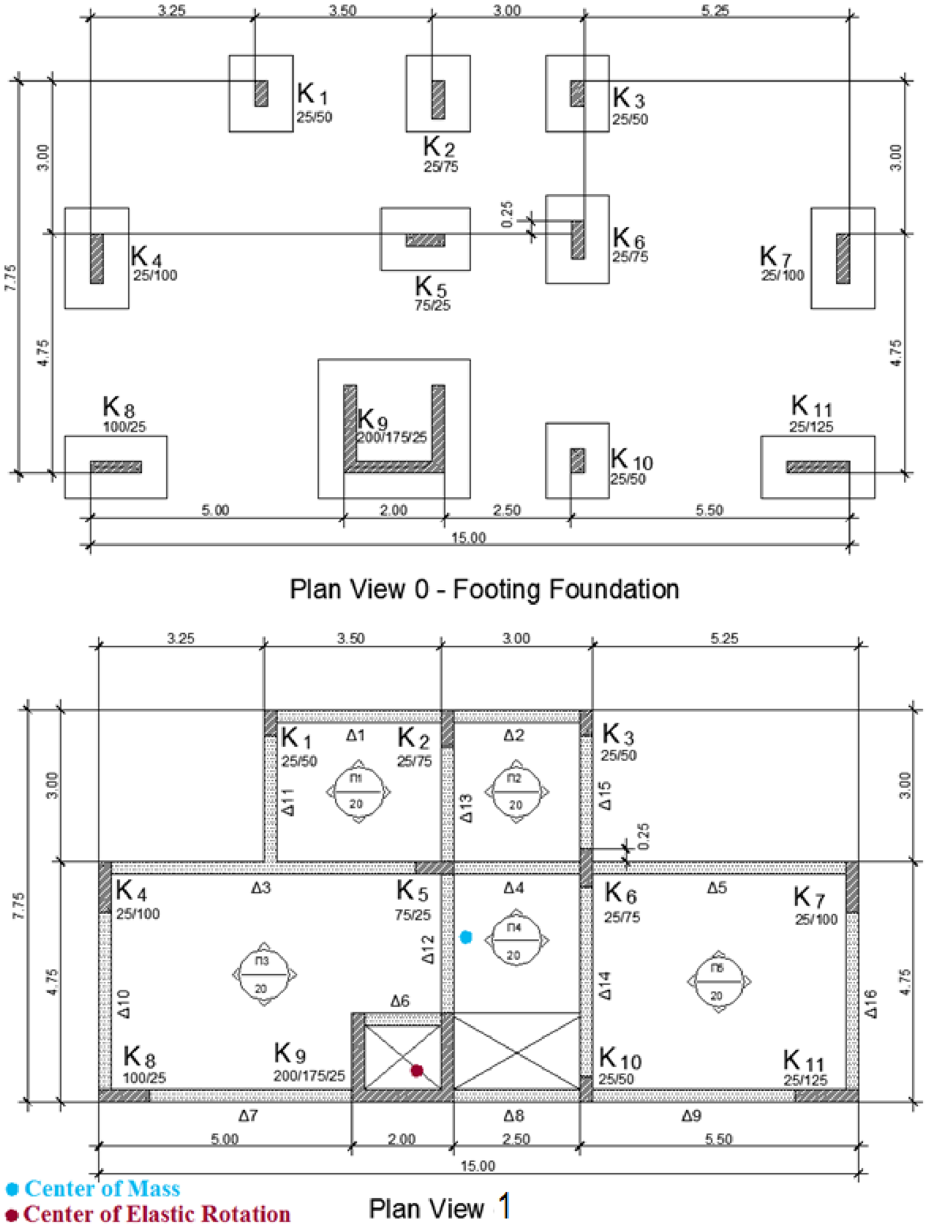


Figure 3. 2-storey RC Building. Plan view of the frame [8].

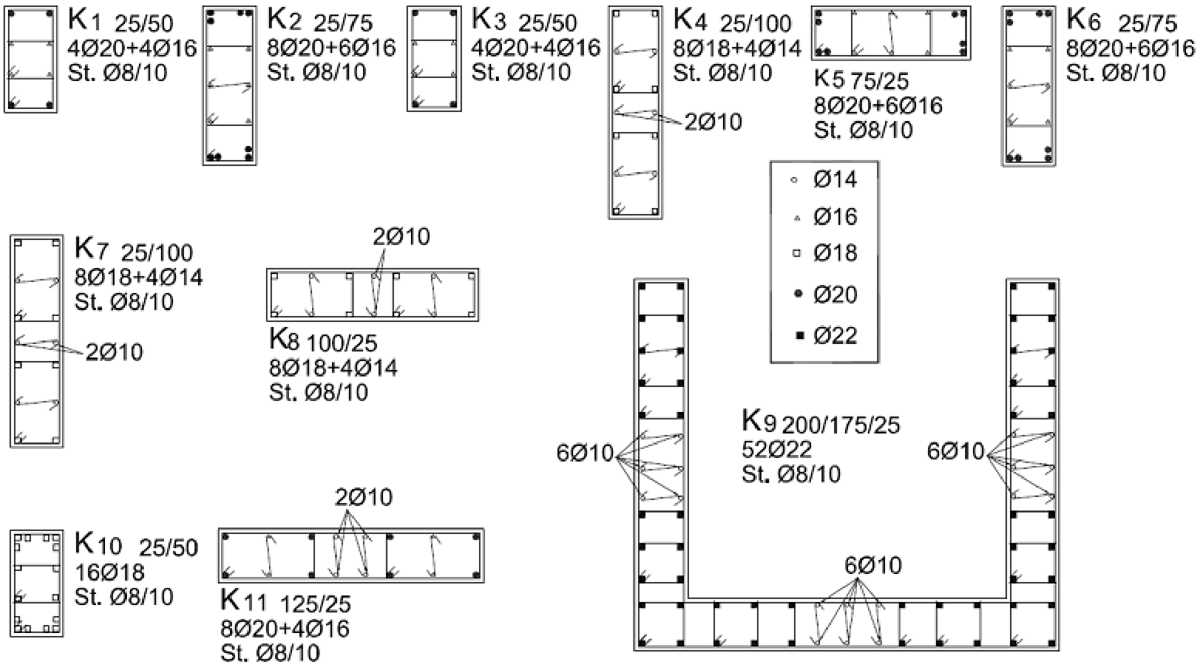
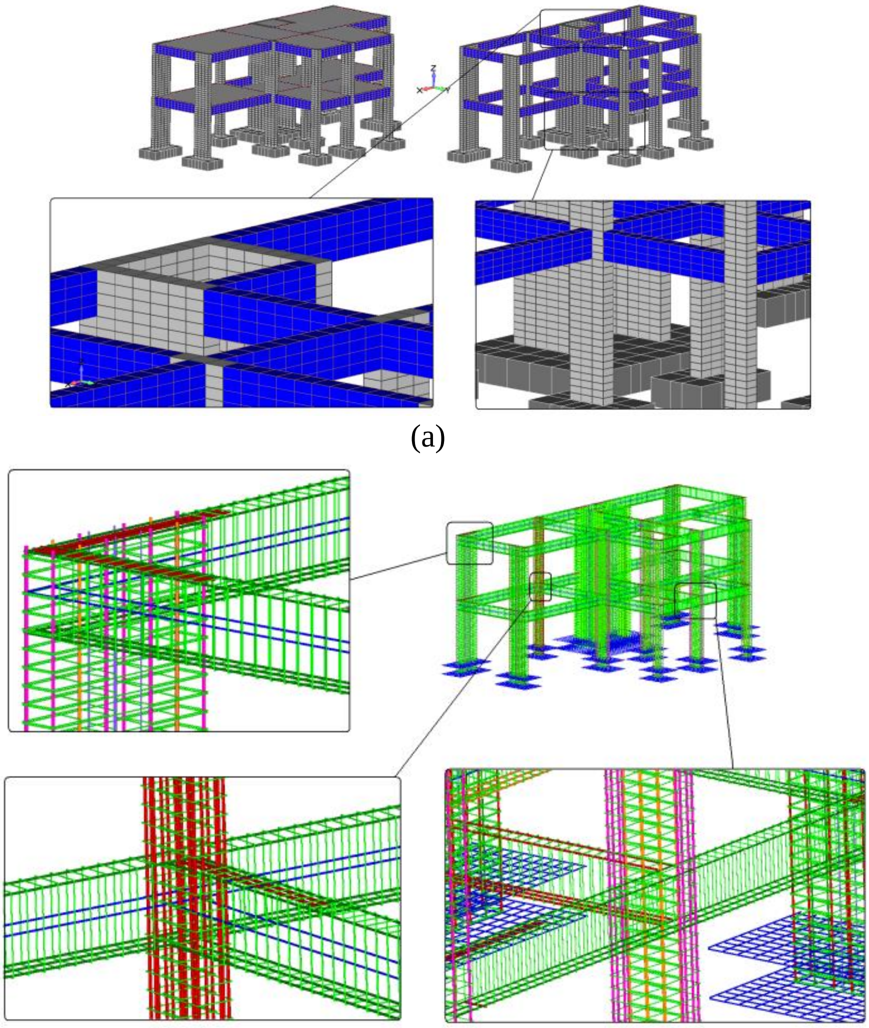


Figure 4. Reinforcement details of the columns and shear walls [8].



(a)

(b)

Figure 5. (a) Hexahedral and (b) rebar macro-element mesh [8].

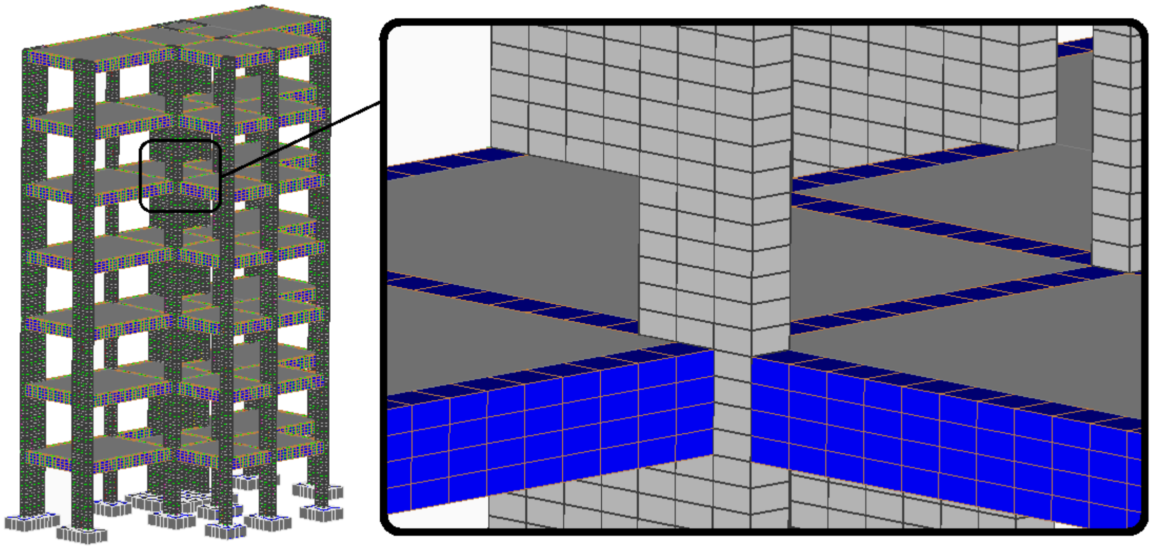
The same numerical problem was used in order to investigate the performance of the integrated mesh generation method when using a 64-bit operating system, where the corresponding embedded rebar mesh generation procedure was completed in 12.8 seconds. By accounting the difference of the computational power of the two CPUs used in these numerical analyses (32-bit: 1.90 GHz and 64-bit: 3.30 GHz), the resulted difference in the computational times is approximately equal to 3 times. The fact that the 64-bit software is 3 times faster than the 32-bit software is not entirely attributed to the type of the operating system, given that the compilers used to build the two applications were different. In addition to that, new CPU technology outperforms the older processing performance thus the comparison between the two processing units cannot be performed in a linear manner. Finally, the DDR3 technology that usually is installed in the new 64-bit motherboards improves significantly the performance of the software, while memory procedures are executed more efficiently.

4.2 RC 7-Storey Building

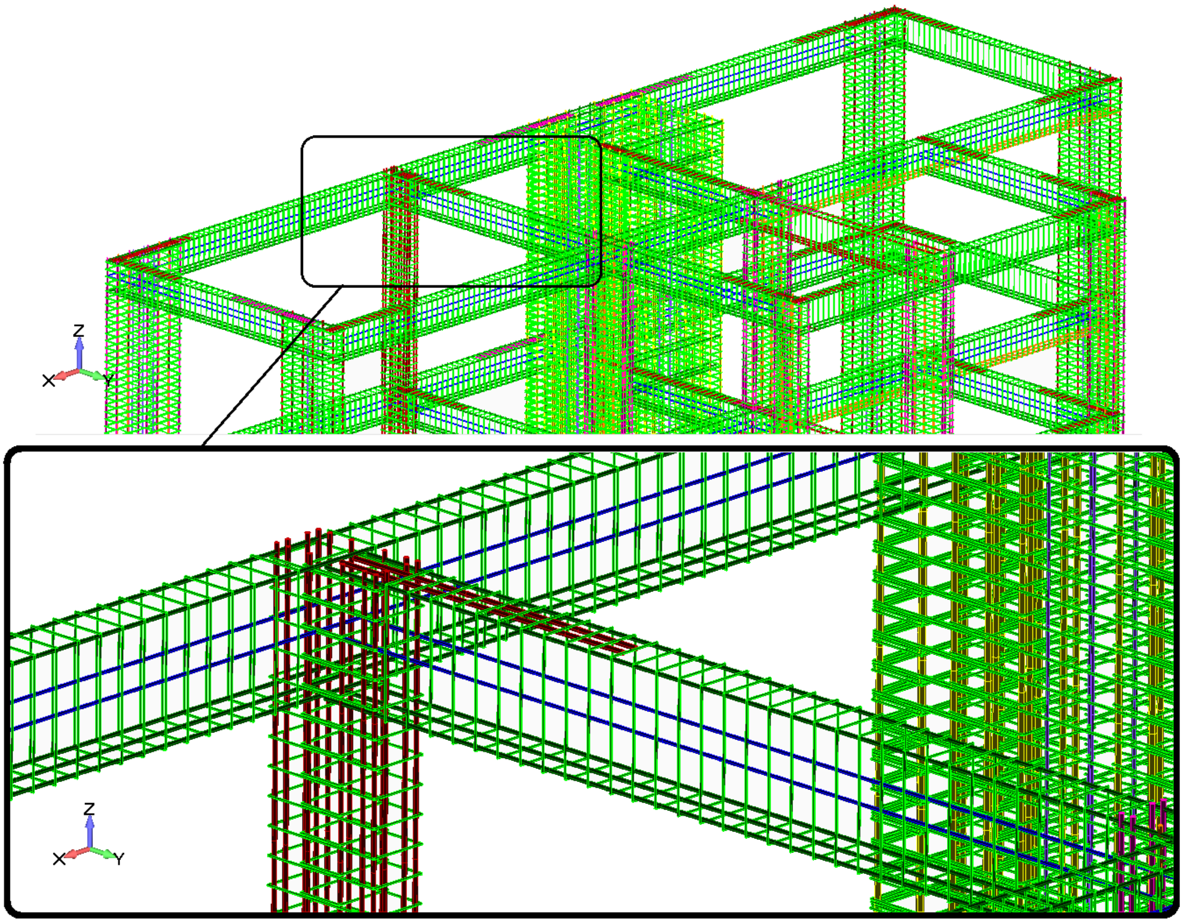
In order to investigate further the required computational time for the allocation of the embedded rebar elements in the case of RC buildings, the previous numerical example is enlarged by increasing the corresponding stories from 2 to 7. The increase of stories that was applied in the above numerical model derived a FE mesh which consists of 14,893 hexahedral elements and 85,479 rebar macro-elements. Fig. 6 shows the resulted hexahedral and rebar macro-element meshes.

It is important to note at this point, that this model was created only for investigational reasons, given that the columns and beams sections are rather small in comparison to the structure's geometry. Furthermore, the foundation type (isolated footings) is not ideal for such a structure (7-storey RC building), therefore this frame design should not be considered as a recommendation.

The next step, after exporting the input file, is to run the mesh generation procedure by executing the corresponding algorithmic procedure. The total allocated memory during the embedded mesh generation procedure was 400 Mb, while the total number of embedded rebar elements allocated was 167,824. The required time so as to generate the 167,824 embedded rebar elements was 183 seconds.



(a)

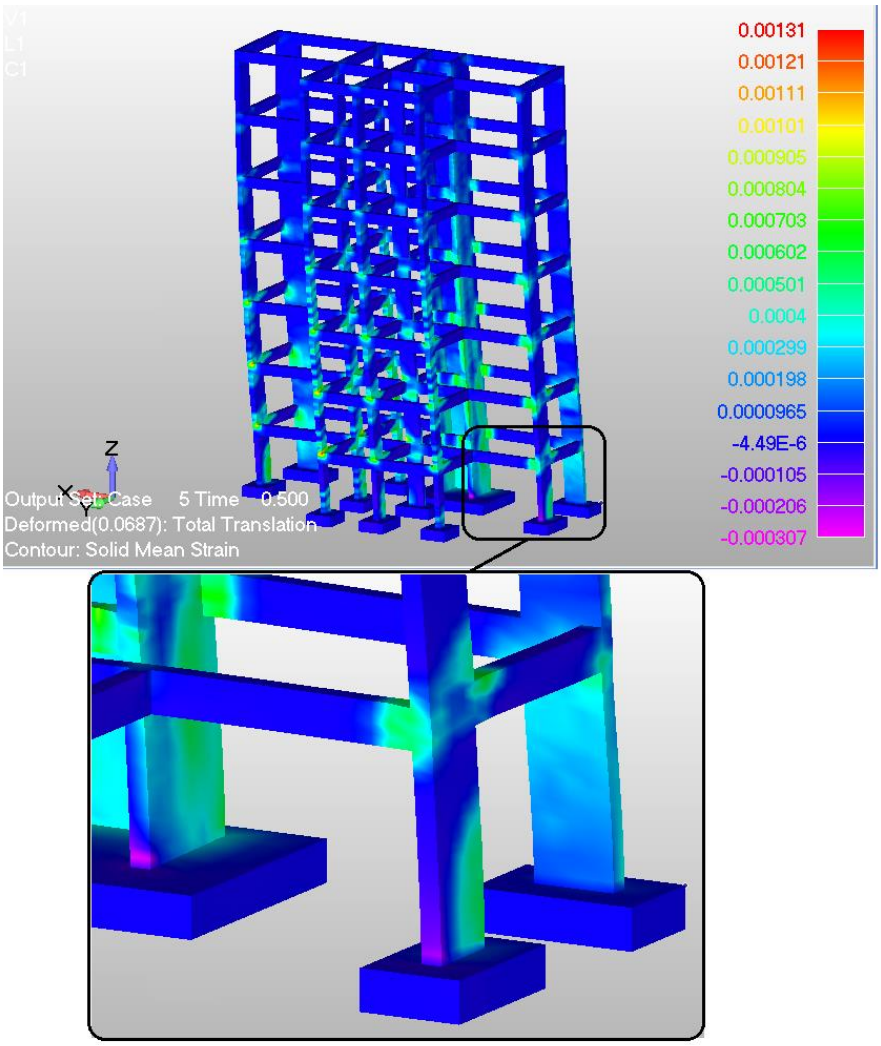


(b)

Figure 6. 7-Storey RC building. (a) Hexahedral and (b) rebar macro-element mesh.

The fact that the required physical memory, allocated by the software so as to generate the embedded rebar elements, was 400 Mb underlines the significant memory demand that rises when generating the embedded rebar elements. Even though the 400 Mb in physical memory might seem relatively large, this is not the case given that the main numerical issue is located in the total required memory allocation during the nonlinear analysis procedure where the stiffness matrix is being assembled. For the at hand numerical model, the total required memory during the solution procedure was 2.8 Gb, the total number of the equations was 104,820 and the number of stiffness coefficients stored in the stiffness matrix was 96,317,751 (770 Mb). As it can be seen, the computational demand that derives from handling numerically the stiffness matrix is much larger than that of the embedded mesh generation procedure. Nevertheless, for this numerical model the embedded rebar mesh approximately required 15% of the total memory demand.

Fig. 7 illustrates the deformed shape of the RC frame and the mean strain distribution in the hexahedral elements as it resulted from the nonlinear analysis. As it can be seen the main strain concentrations are located at the joints and the shear walls. It must be noted that the quasi static horizontal load was applied along the positive direction of the Y global axis. Fig. 3 shows the corresponding cracked pattern of the RC frame for the 2nd load increment (20% of the total horizontal load applied), where it can be seen that the core of the elevator develops a large number of cracks. This is attributed to its large stiffness in comparison to the other vertical structural members that have much smaller sections, thus the core attracts large shear deformations during the push-over analysis. The above mechanical behavior can be depicted in both Figs. 7 & 8.



Output Set: Case 5 Time 0.500
Deformed(0.0687): Total Translation
Contour: Solid Mean Strain

Figure 7. 7-Storey RC building. Mean strain contour at load increment 2 (20% of the total applied load).

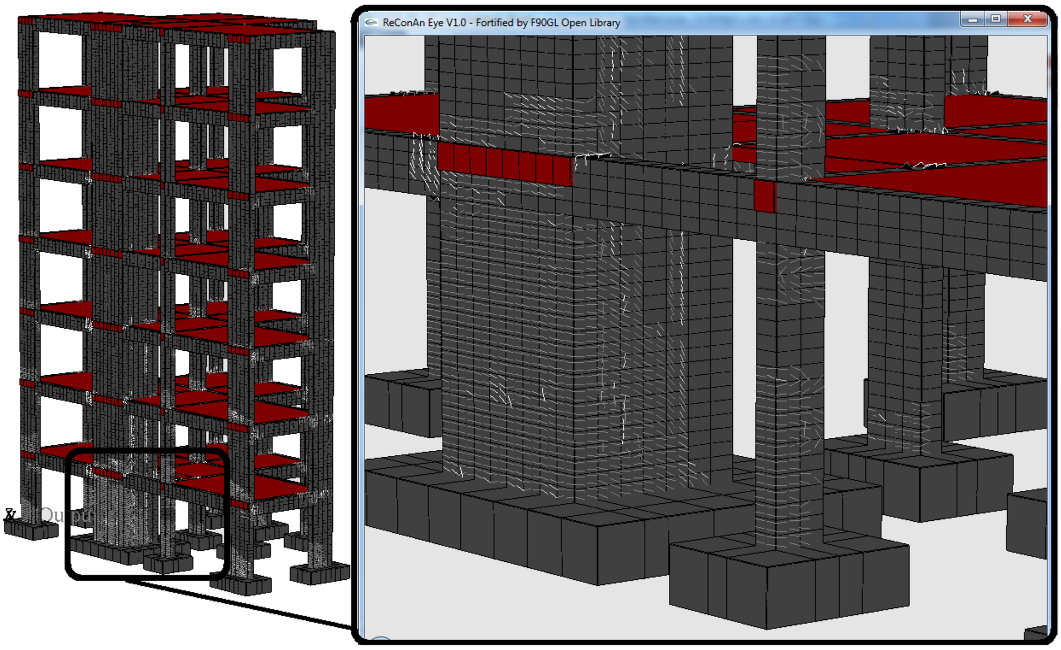


Figure 8. 7-Storey RC building. Crack pattern at load increment 2 (20% of the total applied load).

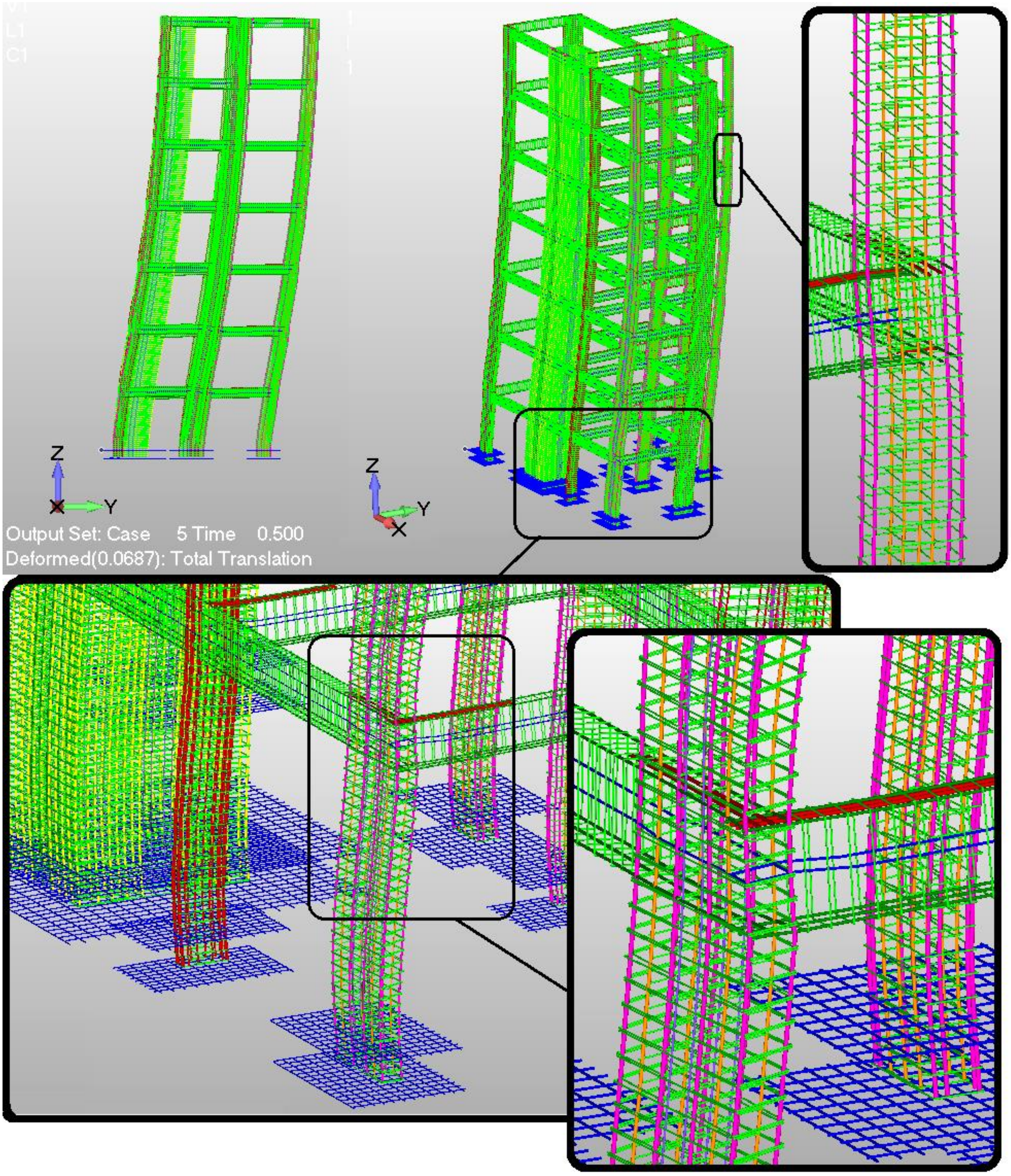

V1
L1
C1
Z
Y
X
Output Set: Case 5 Time 0.500
Deformed(0.0687): Total Translation
Z
Y
X

Figure 9. 7-Storey RC building. Deform shape of the embedded rebar mesh at load increment 5 (50% of the total applied load). Displacement magnification scale factor used: 5.

Fig. 9 shows the deform shape of the embedded rebar mesh as it derived from the 5th load increment of the nonlinear analysis. In order to magnify the reinforcement deformations and present the resulted deformed shape inside the concrete volume in an optimum way, the displacements were increased graphically 5 times. The deform shape of the reinforcement grid is controlled by the deformation shape of the concrete elements (hexahedrons). Fig. 9 illustrates the ability of the method to handle efficiently and with numerical robustness relatively large-scale numerical models.

4.3 RC Arc-Shaped Bridge

This numerical test was chosen in order to illustrate the actual limitations of the embedded mesh generation method when dealing with large-scale implementations. In order to describe the procedure followed for constructing the FE mesh, the geometry of the bridge will be presented followed by the hexahedral mesh construction and verification. Finally, the results from the embedded mesh generation procedure will be presented and the numerical results that derived from the solution of the model will be discussed.

4.3.1 Geometrical Features and Reinforcement Details of the RC Bridge

The geometry of the under study RC bridge is shown in Figs. 10-12. As it can be seen, the bridge has an effective span of 99.1 m, of which 51.55 m is the left span's length and 47.55 m is the right span's length. The total height of the two pylons is 5.1 m and the spacing between them is 5 m (Fig. 11). A typical section of the bridge is given in Fig. 12, where the geometrical features can be depicted. The technical drawings show that the total width of the deck is 10.4 m and it has a height of 2.3 m. The different thicknesses of each structural component of the bridge's deck are given in Table. 3.

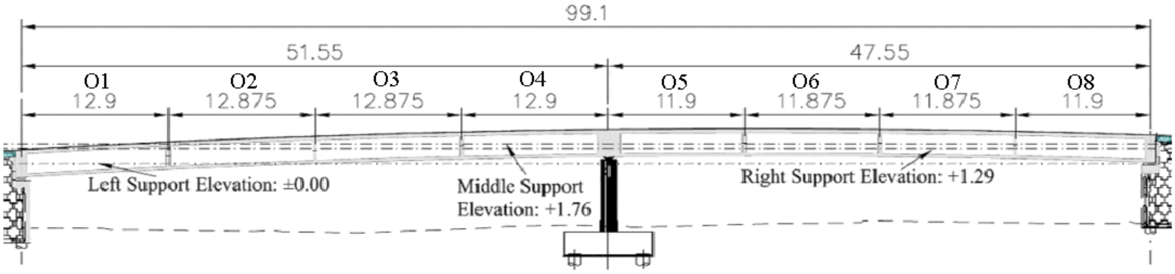


Figure 10. RC Bridge. View of elevations.

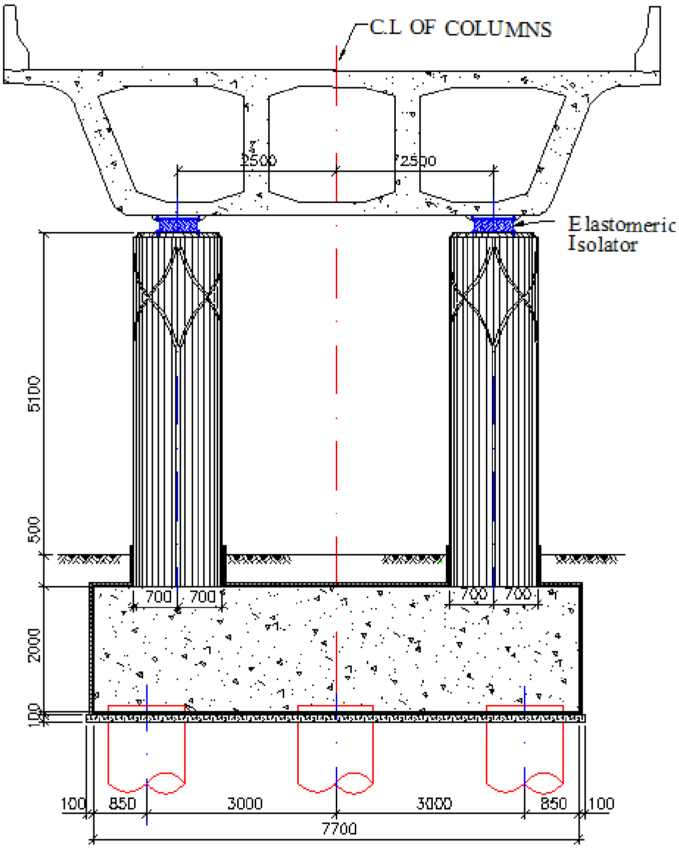


Figure 11. RC Bridge. Geometry of the Pylons (dimensions in mm).

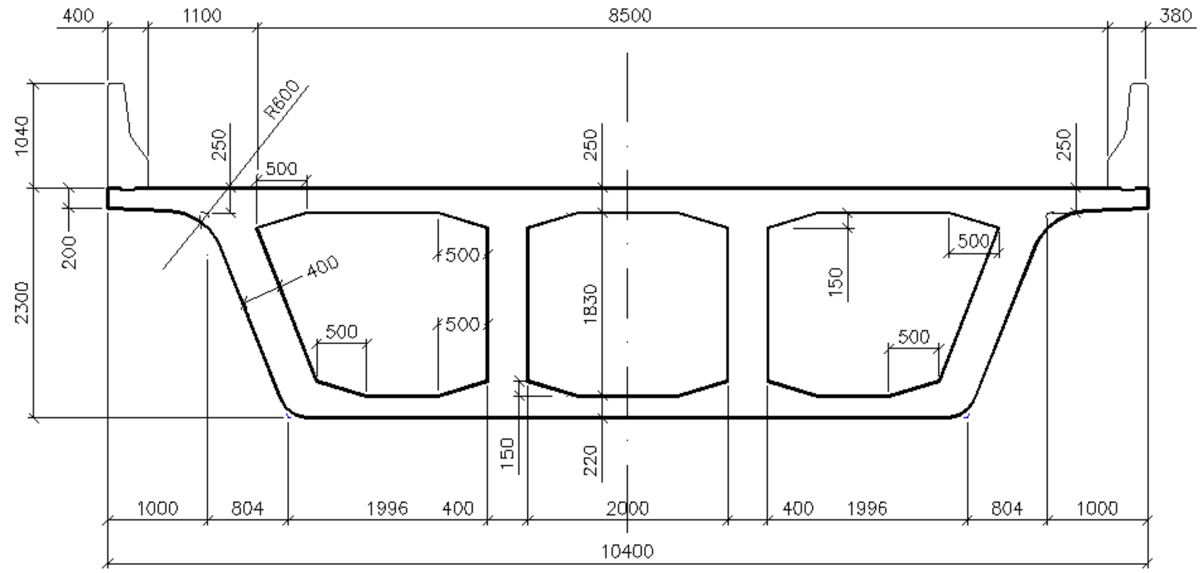


Figure 12. RC Bridge. Geometry of the section of the deck (dimensions in mm).

a/a	Structural component	Thickness in cm
1	Upper Deck	25
2	Lower Deck	22
3	Overhangs	20
4	Vertical Walls	40
5	Vertical Diaphragms	30
6	Vertical Diaphragms at Supports	100
7	Vertical Diaphragm above Pylons	200

Table 3. RC Bridge. Thicknesses of different structural components of the deck.

The reinforcement details of the Pylons are shown in Fig. 13, where it can be seen that 26 rebars of Ø32 mm are used and the stirrups have a diameter of Ø12 mm. Fig. 14 shows the reinforcement details of the foundation (pile cap). It is important to note that the model will assume a fixed base at the level of the piles, thus the 6 piles will not be included in the FE mesh.

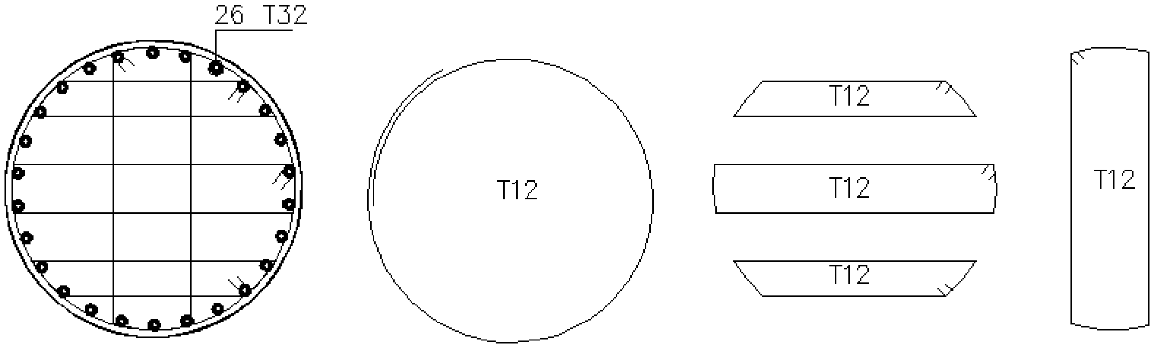


Figure 13. RC Bridge. Reinforcement details of the Pylon’s section.

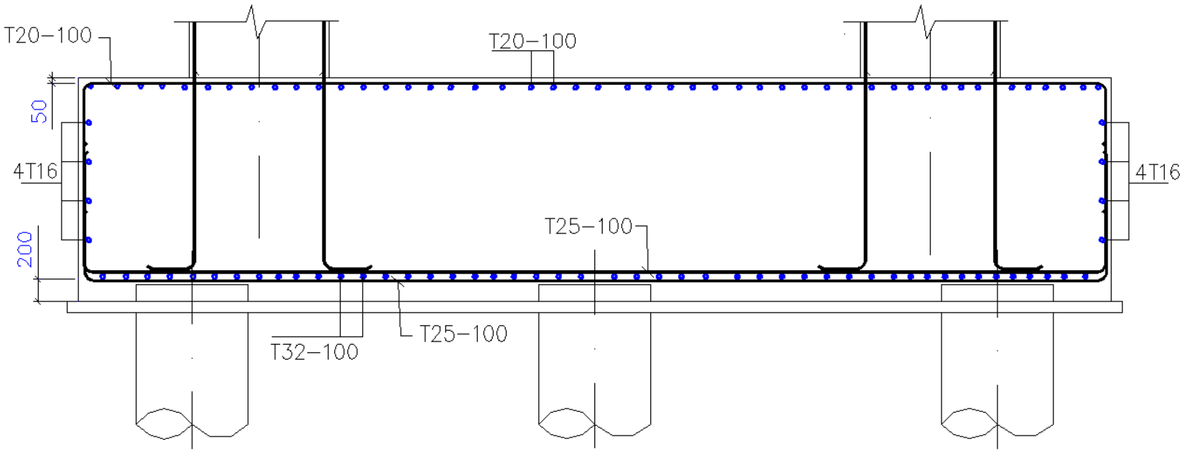


Figure 14. RC Bridge. Reinforcement details of the foundation.

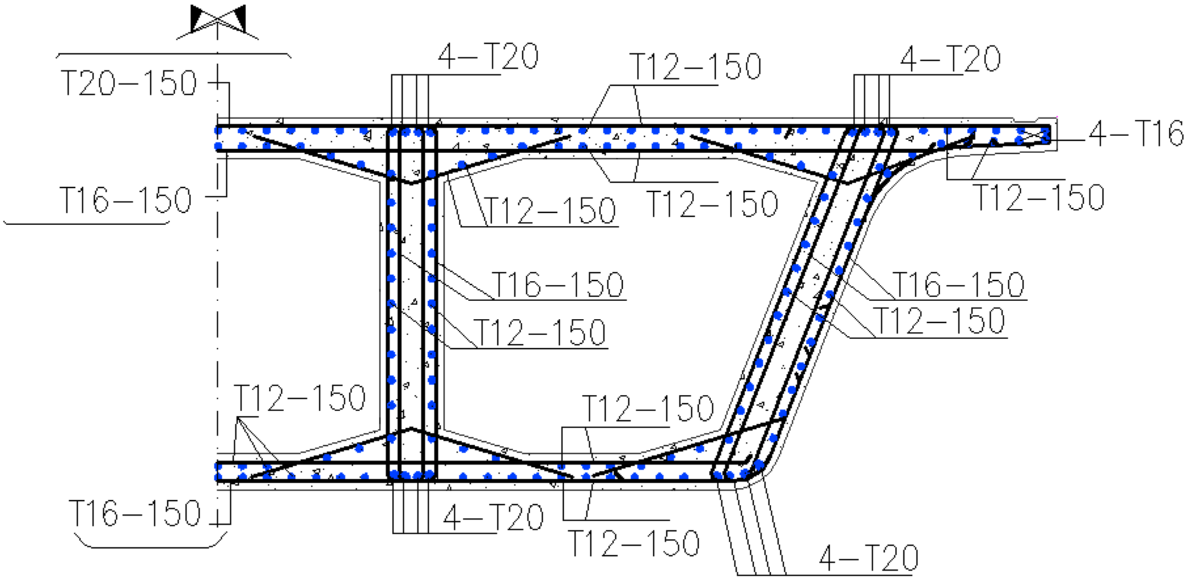


Figure 15. RC Bridge. Reinforcement details of a typical deck section.

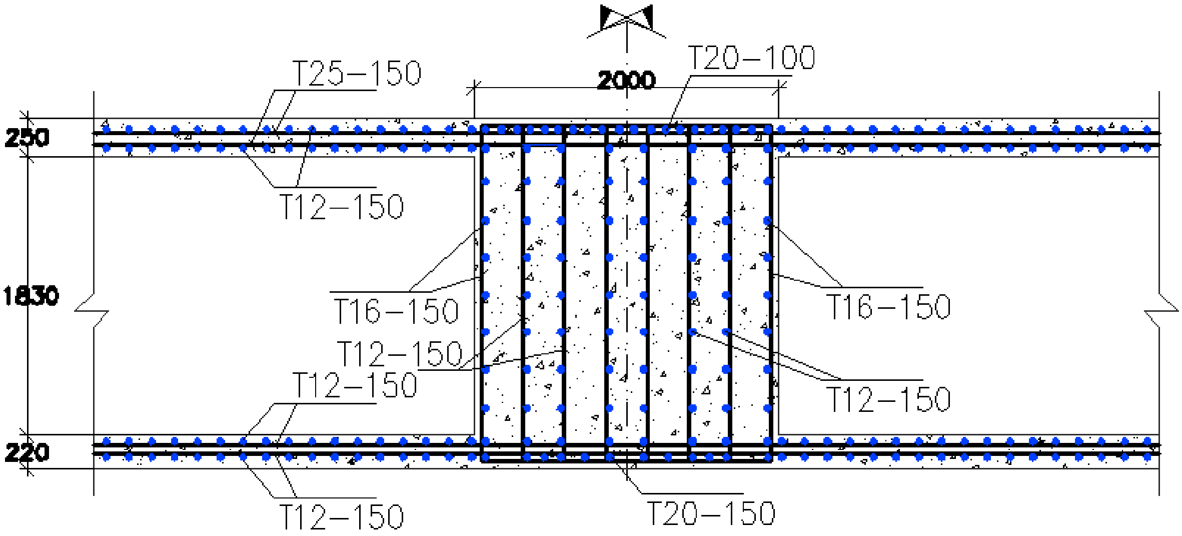


Figure 16. RC Bridge. Reinforcement details of the middle diaphragm.

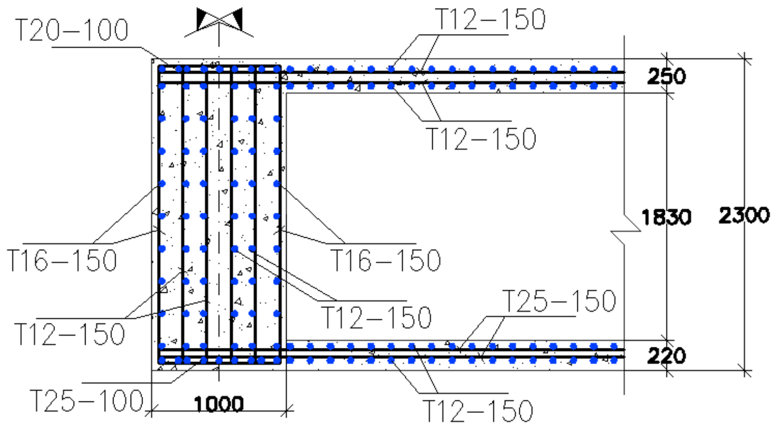


Figure 17. RC Bridge. Reinforcement details of the diaphragms at the left and right supports.

Fig. 15 shows a typical deck section reinforcement detailing which foresees the use of 12, 16 and 20 mm rebars, while the geometry of the reinforcement arrangement of the support diaphragms can be depicted in Figs. 16 & 17. It must be noted that this is the preliminary design (AASHTO code) of a RC bridge in Dubai that has not been constructed yet. Given that the objective of this research work is to investigate the embedded mesh generation procedure, the ultimate limit state of the bridge that derives from the analysis will not be presented in this paper. In addition to the above, for the same reason the bridge's post-tension cables will not be modeled.

4.3.2 Constructing, Managing and Verifying the Hexahedral FE Mesh

ReConAn FEA uses Femap [14] in order to construct the numerical model which is eventually exported into a text file (.neu → neutral file) that is used to generate the FE numerical model during the analysis of the problem. For the needs of this numerical implementation, the use of different mesh control techniques were adopted so as to avoid the construction of a FE mesh that would have derived mesh inconsistencies and numerical problems related to meshing issues. Fig. 18 shows the final mesh of the 102,934 hexahedral elements (8-noded). The details related to the mesh are given in Table 4 and as it can be seen, the total number of concrete elements is 102,622, while the total number of nodes (excluding the embedded rebar macro-elements) is 168,400. The average hexahedral edge size used to derive the FE mesh of the RC bridge was 20 cm.

One of the mesh control methodologies adopted in this work, was the division of the FE mesh into Layers. Specifically, 56 Layers were created only for controlling and organizing the hexahedral elements. Femap has the ability of assigning each object or FE into a specific Layer that can be handled the same way that Layers are used in a CAD graphical environment. By adopting this approach, the mesh can be divided into groups (i.e. Upper Deck, Lower Deck, Vertical Walls, etc.) according to the structural member's location.

As it can be seen in Fig. 10, the bridge's deck has 6 vertical diaphragms and the diaphragm in the middle that eventually divide the deck into 8 openings. The proposed logic behind the topological sorting of the Layers is based on the assumption that the hexahedral mesh of the bridge will be divided into four main groups of Layers (a. Deck Span, b. Bearing Supports, c. Support Diaphragms and d. Pylons/foundation), while in each main group of Layers the mesh of all structural members (Upper Deck, Vertical Walls, etc.), which belong to a specific opening of the bridge, will be assigned into the designated Layer. The graphical illustration of the 56 Layers used in order to manage the hexahedral elements' mesh is shown in Fig. 19 and the use of Layers is demonstrated in Fig. 20 for the case of the deck opening O1.

An additional mesh control procedure that was implemented in order to verify that each part of the under construction hexahedral FE mesh was consistent, thus avoiding inducing numerical problems during the embedded mesh generation procedure and the analysis procedure of the structure, was the “Convergence Analysis by Parts” procedure. The proposed methodology foresees the analysis of each part of the numerical model of the hexahedral mesh (i.e. 1st opening, 2nd opening, pylons with pile cap, etc.), by applying only the self-weight loads and solving the static linear numerical problem.

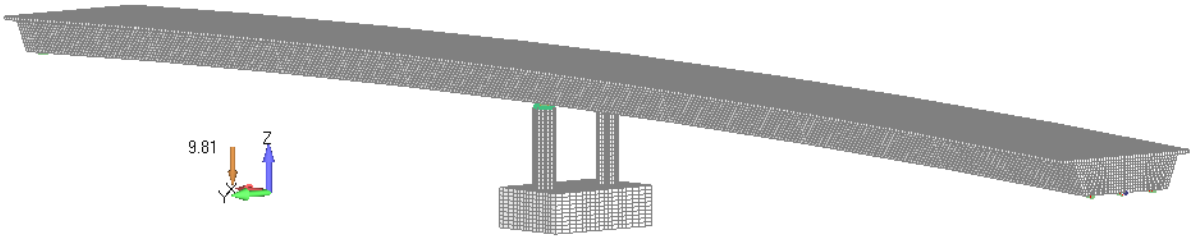


Figure 18. RC Bridge. FE mesh of 8-noded hexahedral elements.

a/a	Description	Value
1	Hexa8 total number of Elastomeric elements	120
2	Hexa8 total number of Steel elements	192
3	Hexa8 total number of Concrete elements	102,622
4	Total number of Hexa8 elements	102,934
5	Total number of Hexa8 nodes	168,400

Table 4. RC Bridge. FE mesh data related to the 8-noded hexahedral mesh.

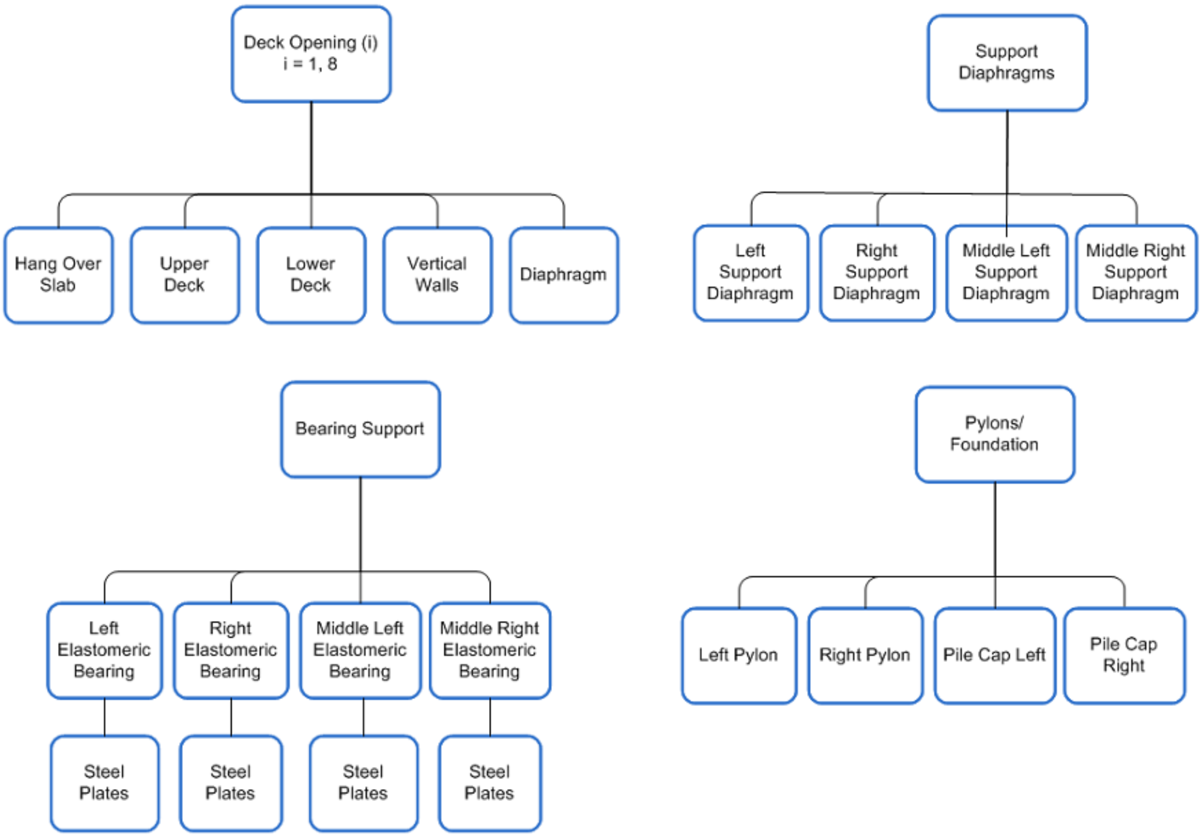


Figure 19. RC Bridge. Hexahedral mesh Layer organization chart.

For the at hand model, the total number of mesh convergence analysis performed was 7 and the corresponding models used are shown in Fig. 21. The hexahedral FE meshes for each main Layer group are shown in Figs. 22-32, while the distribution of the hexahedral elements, according to the assumed Layers for the management of the hexahedral elements' mesh, are given in Table 5. It is evident that the largest number of hexahedral elements is located at the deck of the RC bridge. After the completion of the mesh convergence investigation, the resulted FE mesh consisted of 102,934 from which 102,622 are concrete hexahedral elements.

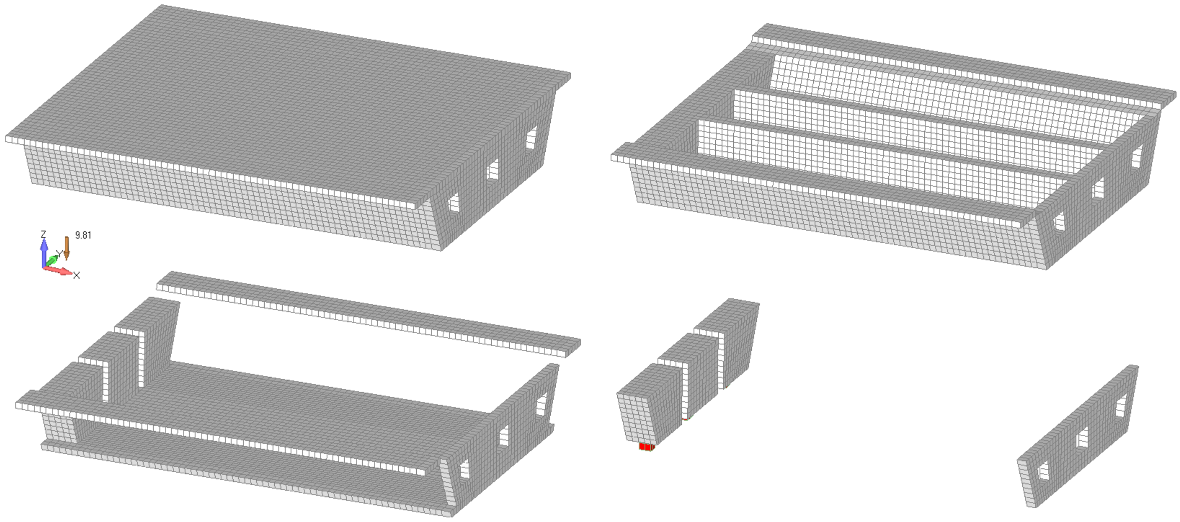


Figure 20. RC Bridge. Hexahedral mesh of the 1st Deck Opening O1. Graphical activation and deactivation of Layers.

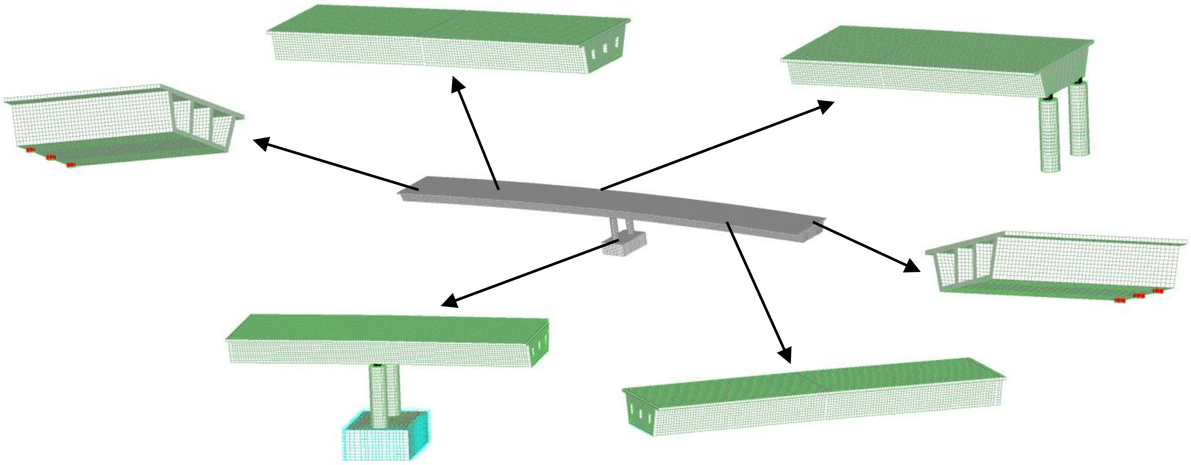


Figure 21. RC Bridge. Models used for the mesh convergence analysis procedure ("Convergence Analysis by Parts").

Fig. 33 shows the deformed shape and the von Mises Stress contour for the case of the 4th and 5th Deck Openings (O4 & O5), while the middle elastomeric bearings and the two Pylons that are attached to the pile cap are included in this Convergence Analysis by Part check. As it can be depicted in this figure, the deck behaves as a sway mechanism which rotates counterclockwise. Given that the only loads accounted for are the loads that result from the self-weight of the hexahedral elements and given the fact that the geometry of the bridge foresees that the 4th opening's length (Fig. 10) to be equal to 12.9 m while the corresponding 5th opening's length to be equal to 11.9 m, the resulted deformation (counterclockwise rotation Fig. 33) was the expected one.

a/a	Structural Member	Hexa8
1	Opening O1 Deck + Left Support Diaphragm (Fig. 18)	13,824
2	Opening O2 Deck (Fig. 19)	11,254
3	Opening O3 Deck (Fig. 20)	11,254
4	Opening O4 Deck + Half of the Middle Diaphragm (Fig. 21)	12,662
5	Opening O5 Deck + Half of the Middle Diaphragm (Fig. 22)	12,432
6	Opening O6 Deck (Fig. 23)	10,558
7	Opening O7 Deck (Fig. 24)	10,558
8	Opening O8 Deck + Right Support Diaphragm (Fig. 25)	11,800
9	Two Pylons (Fig. 26)	3,080
10	Pile Cap (Fig. 26)	5,200
11	Elastomeric Bearings + Steel Plates (Fig. 27)	312
	Total	102,934

Table 5. RC Bridge. Hexahedral element distribution.

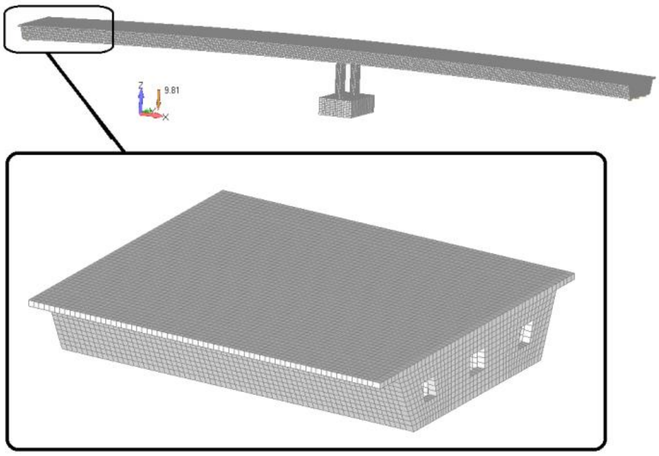


Figure 22. RC Bridge. FE mesh of opening O1 and left support’s diaphragm. (13,824 hexa8)

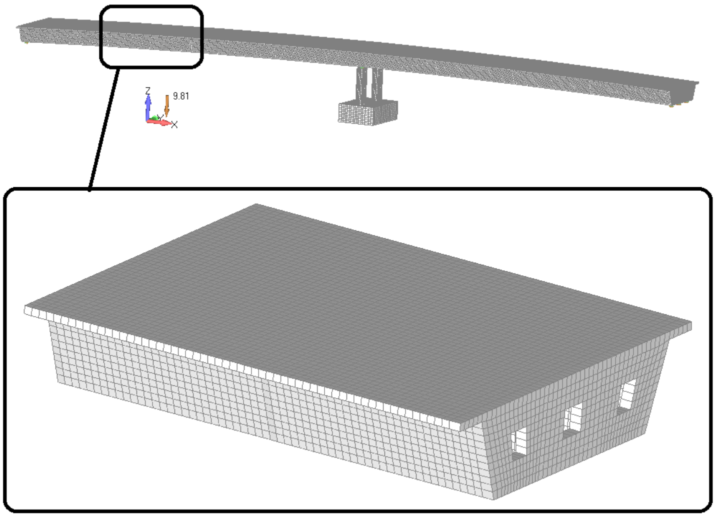


Figure 23. RC Bridge. FE mesh of opening O2. (11,254 hexa8)

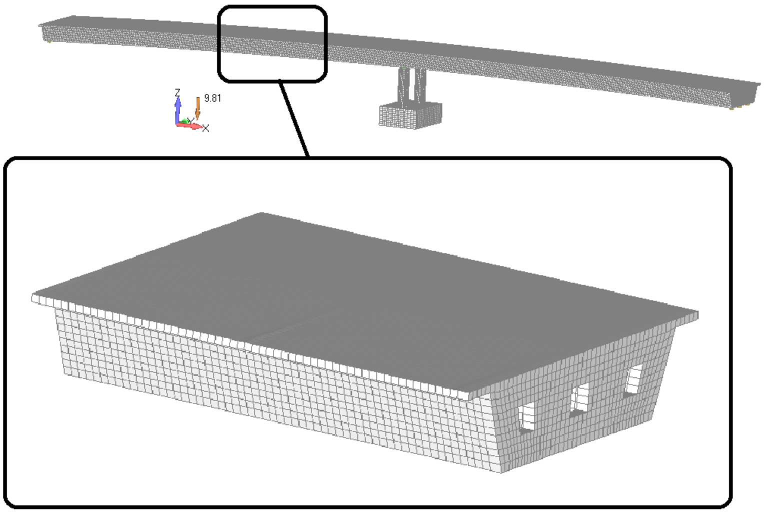


Figure 24. RC Bridge. FE mesh of opening O3. (11,254 hexa8)

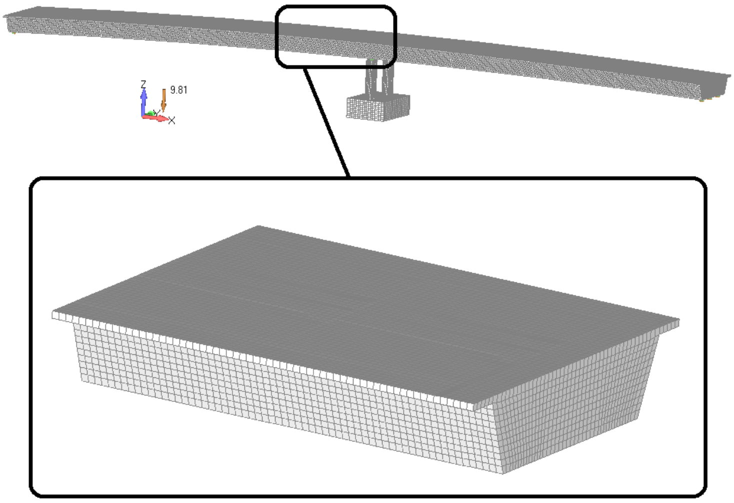


Figure 25. RC Bridge. FE mesh of opening O4 + half of the middle diaphragm. (12,662 hexa8)

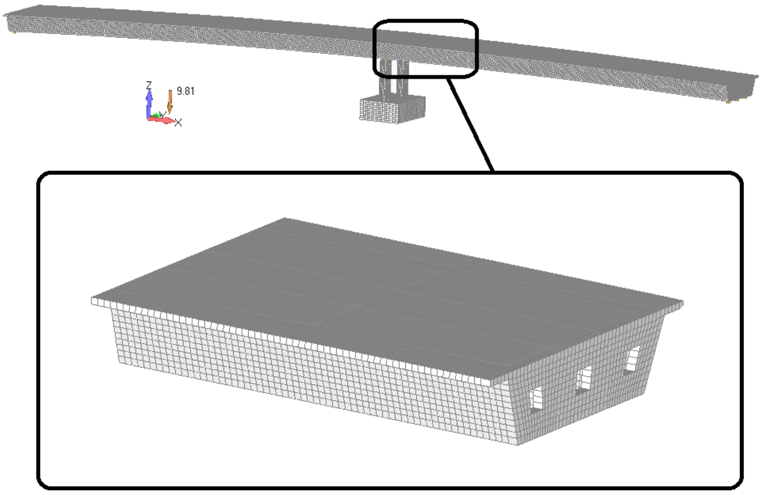


Figure 26. RC Bridge. FE mesh of opening O5 + half of the middle diaphragm. (12,432 hexa8)

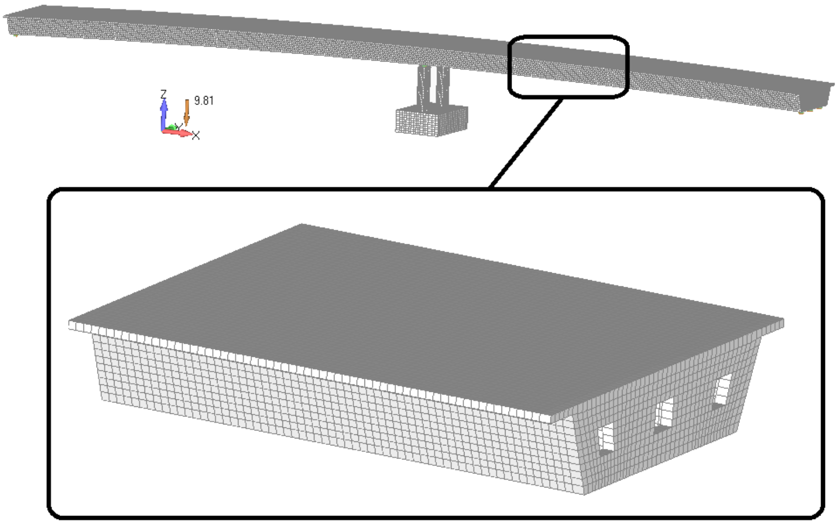


Figure 27. RC Bridge. FE mesh of opening O6. (10,558 hexa8)

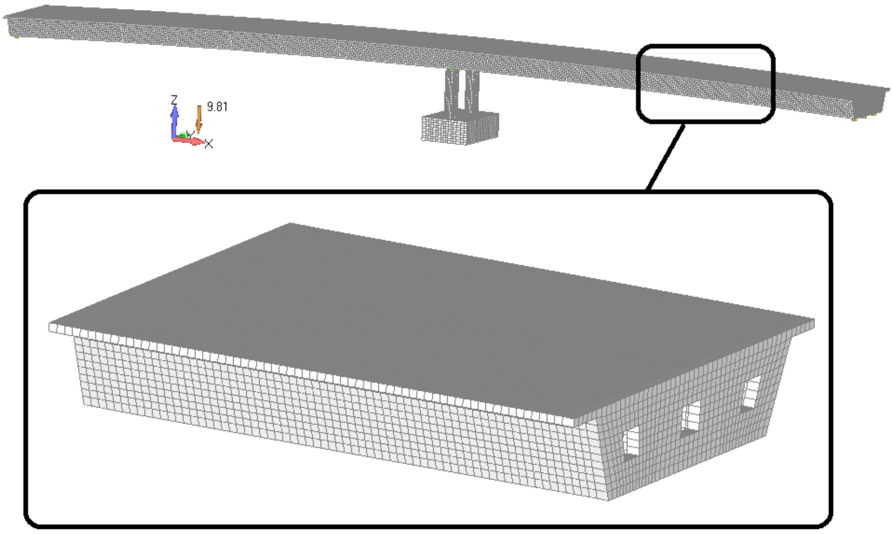


Figure 28. RC Bridge. FE mesh of opening O7. (10,558 hexa8)

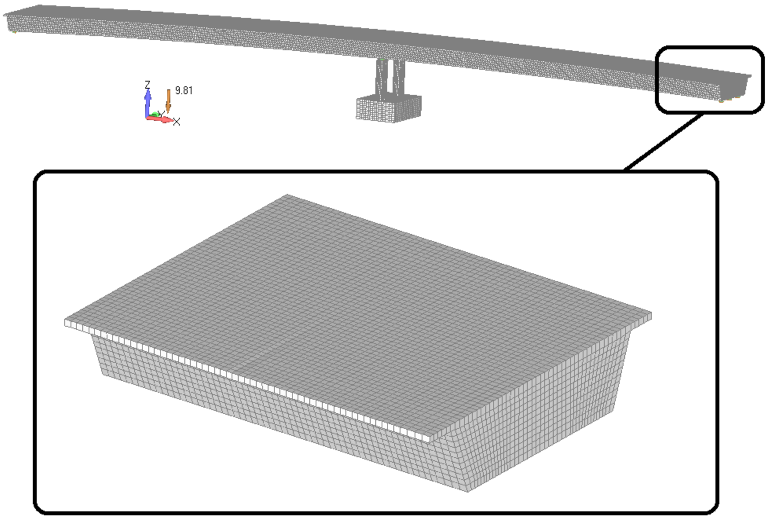


Figure 29. RC Bridge. FE mesh of opening O8 and right support’s diaphragm. (11,800 hexa8)

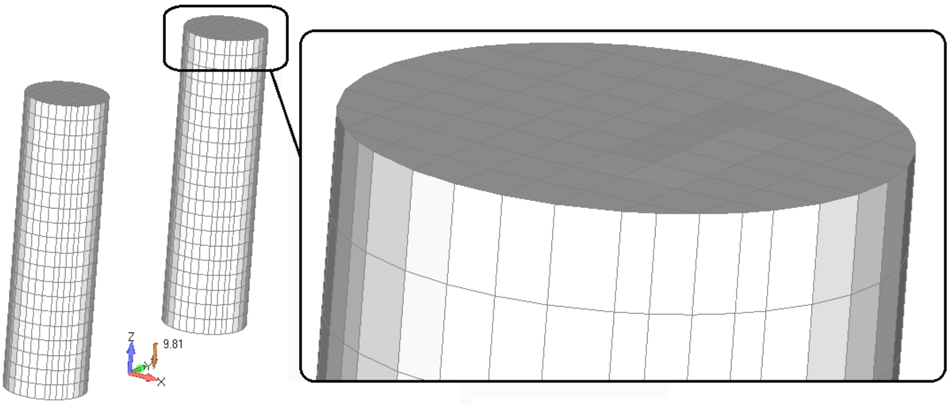


Figure 30. RC Bridge. FE mesh of the two pylons. (3,080 hexa8)

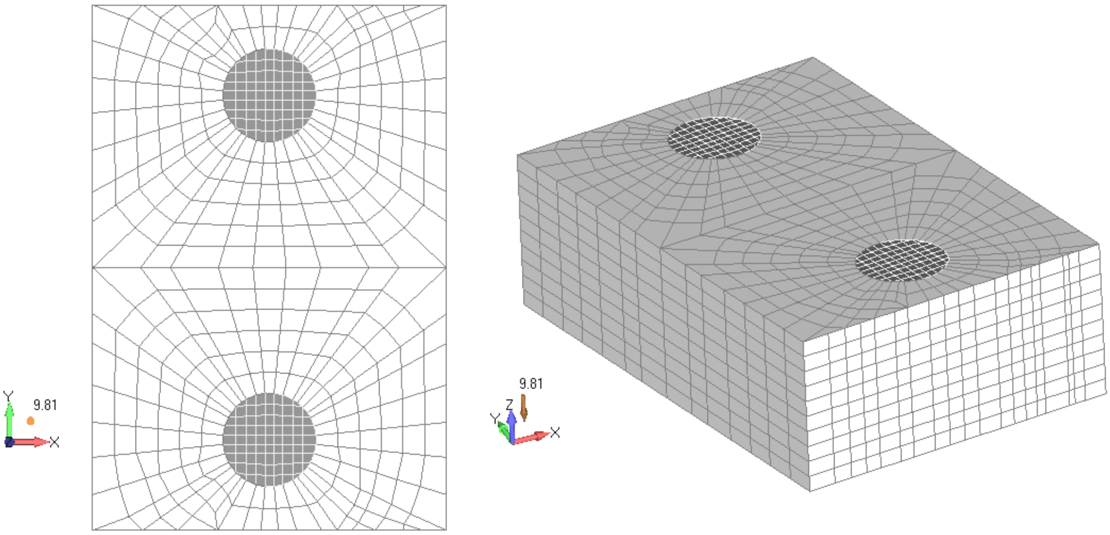


Figure 31. RC Bridge. (Left) Plan view and (Right) 3D view of the pile cap’s mesh. (5,200 hexa8)

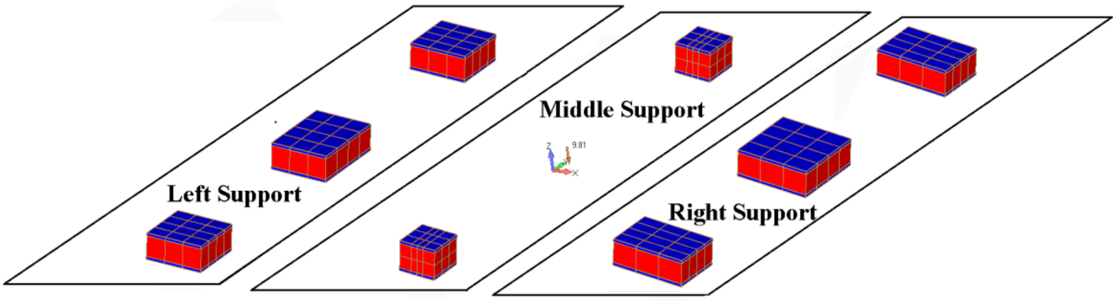


Figure 32. RC Bridge. FE Mesh of the elastomeric bearings and steel plates (312 hexa8).

The hexahedral mesh construction was finalized by performing the analysis of the complete hexa8 FE mesh by applying the self-weight of the structure. The deformed shape and the von Mises Stress distribution as derived from the analysis are shown in Figs. 34-36. It must be noted here that the total number of unknowns for the case of the final hexahedral FE mesh was 502,478, the total number of the stiffness matrix elements was 657,655,263 and the required physical memory for solving the numerical model without the embedded reinforcement was 7.6 Gb.

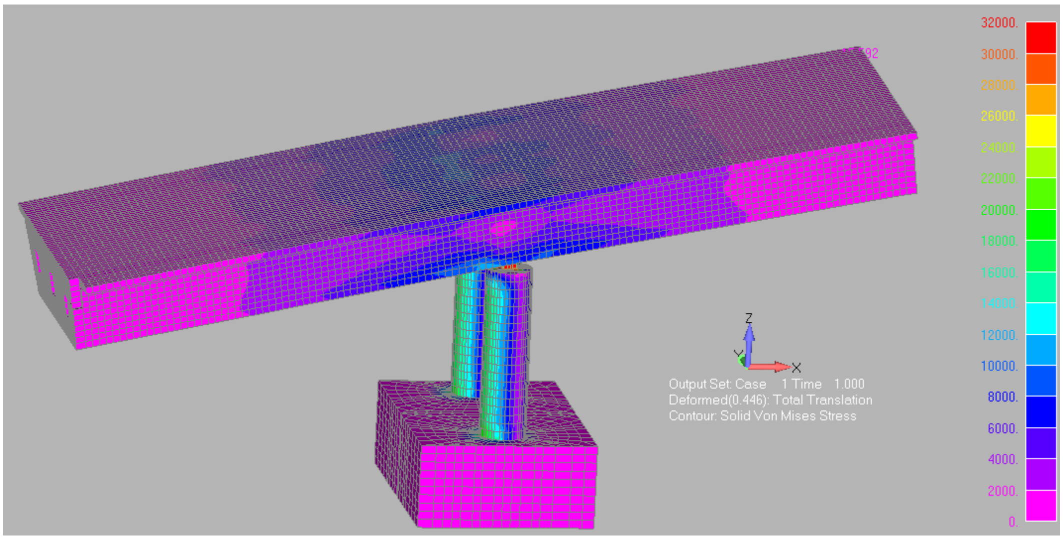


Figure 33. RC Bridge. Deformed shape and solid von Mises Stress contour (kPa) due to the self-weight of the structure (displacement magnification factor x5).

It must be noted at this point that the material model used for the linear elastic analyses that were performed so as to check the numerical response of the constructed hexahedral mesh, was the elastic von Mises material model. This material model is used to simulate the mechanical behavior of steel material thus uses the steel nominal weight to generate the self-weight loads. This is the reason why the stress values are increased within the presented von Mises stress contours.

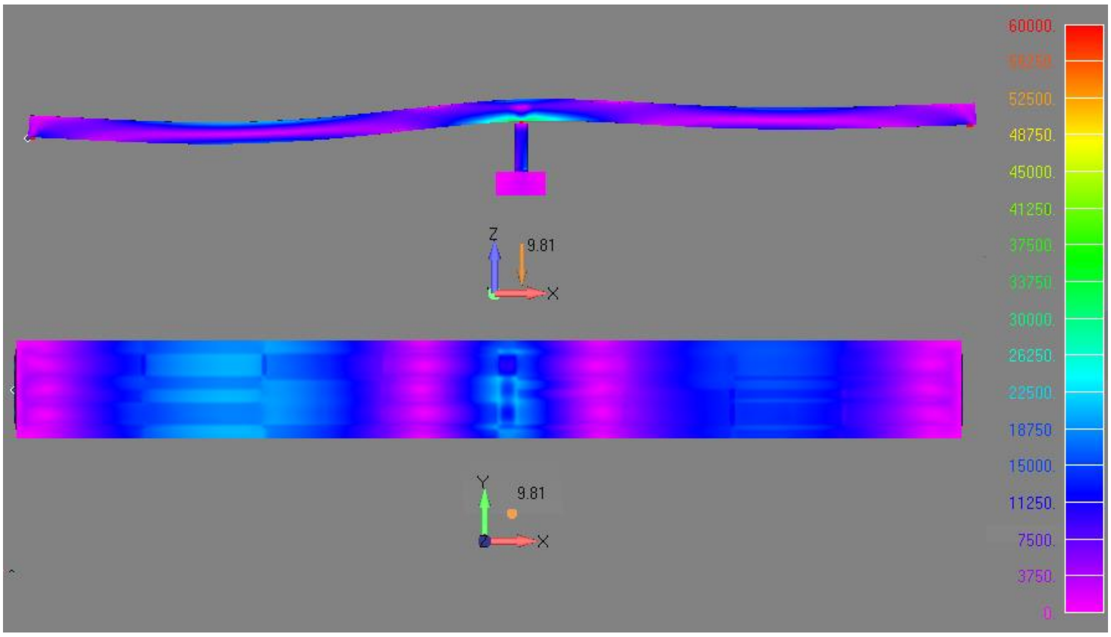


Figure 34. RC Bridge. XZ side and XY top view of the deformed shape and solid von Mises Stress contour (kPa), of the final hexa mesh, due to the self-weight of the structure.

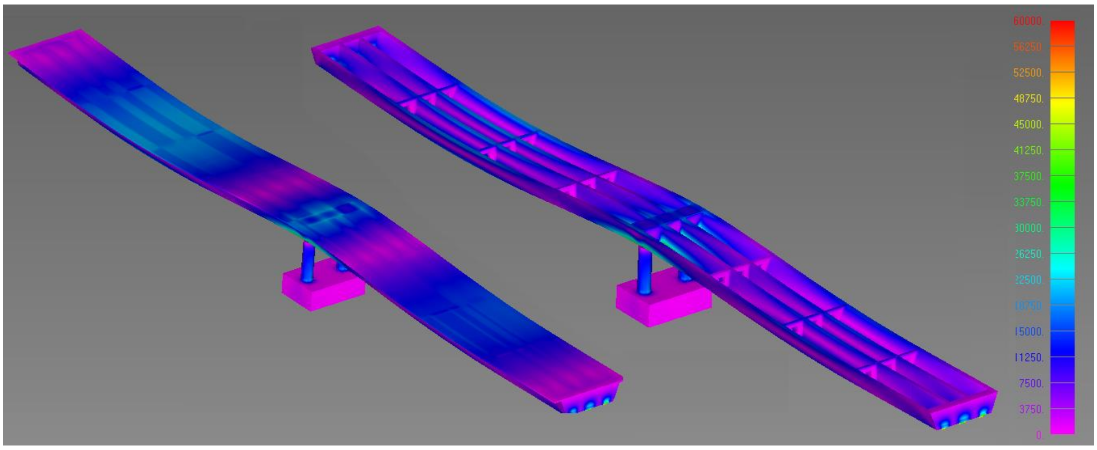


Figure 35. RC Bridge. Deformed shape and von Mises Stress contour. (Left) View of the full model and (Right) Internal view of the vertical and diaphragmatic walls.

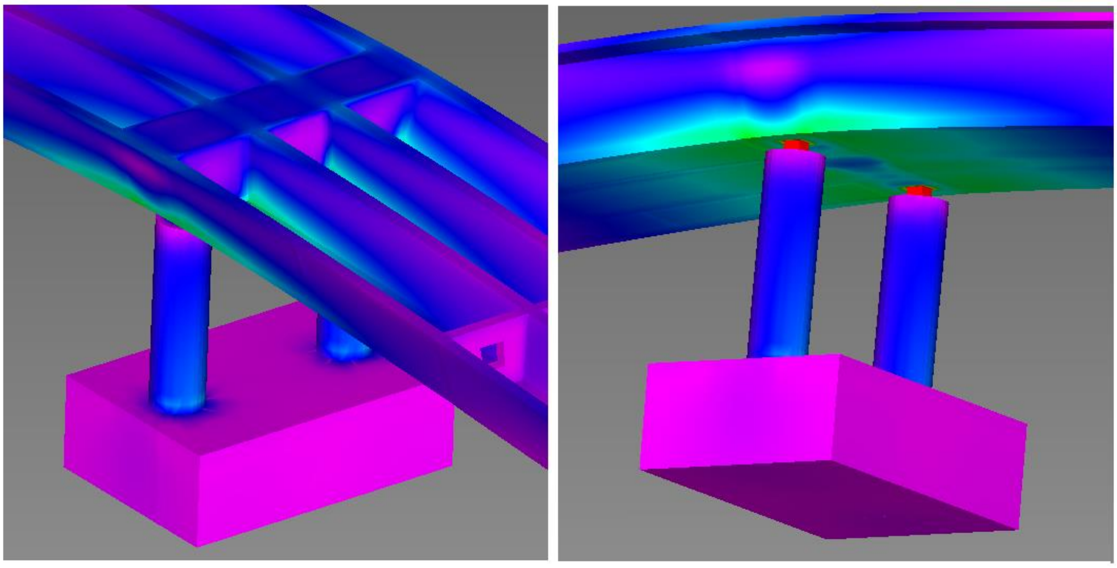


Figure 36. RC Bridge. Deformed shape and von Mises contour of the area where the deck and the pylons connect through the elastomeric bearings.

4.3.3 Constructing, Managing and Verifying the Embedded Reinforcement FE Mesh

When constructing the embedded rebar macro-element grid of a large-scale model, it is required to consider similar mesh control techniques as described in the previous section in order to achieve the construction of an error-free mesh. Given the fact that the required computational time for the analysis of this model will be significant, any mesh irregularities will induce numerical instabilities forcing the nonlinear analysis to terminate, while the size of the mesh will make it time-consuming if any mesh modification issues arise. For this reason the same mesh managing approach illustrated in the previous section was adopted in order to control the resulted embedded rebar mesh in a step-by-step logic (Layers and Convergence Analysis by Parts).

4.3.3.1 Managing the Embedded Rebar Mesh

As it is shown in Fig. 32, the embedded rebar mesh was divided into 48 Layers according to the RC bridge’s geometry. The 48 Layers were divided into 3 main groups so as to optimize the viewing procedure and the allocation of each reinforcement arrangement according to its positioning inside the structure. Furthermore, after the completion of the construction of the embedded rebar macro-element mesh for each structural member (see Table 6) of the bridge, a convergence analysis was performed so as to assess the derived FE model for each structural part of the bridge’s model.

4.3.3.2 Convergence Analysis by Parts

The embedded rebar mesh construction began by constructing and testing the Pylon/Pile Cap mesh (Figs. 38 & 39). The geometry of the hexahedral mesh of these structural members is irregular and mainly non-prismatic, while the embedded rebar mesh construction of the embedded rebar macro-elements was performed by using long macro-elements that intersect more than 15 hexahedral elements. In some areas of the Pile Cap mesh, rebar macro-elements penetrate up to 20 hexahedral elements.

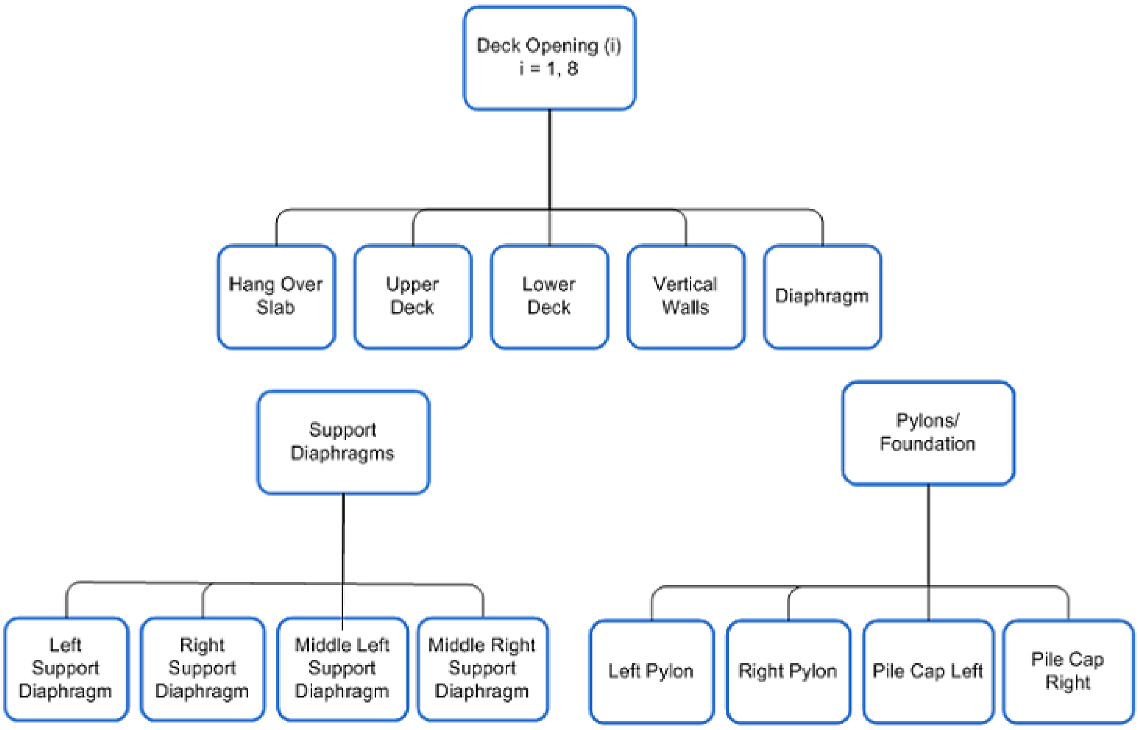


Figure 37. RC Bridge. Embedded rebar mesh Layer organization chart.

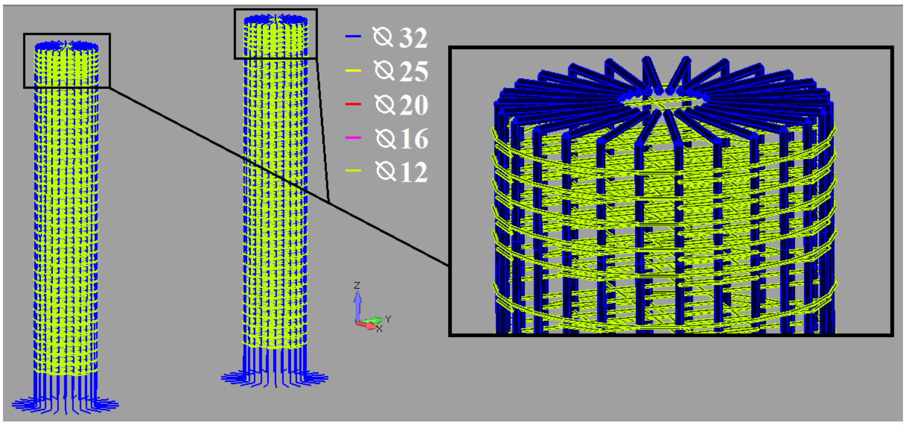


Figure 38. RC Bridge. Embedded rebar mesh of the Pylons (4,432 embedded rebar macro-elements).

As it was described in [8], embedded rebar macro-elements are the initial line-objects that are used from the embedded mesh generation algorithm so as to generate the embedded rebar elements in each hexahedral element. The geometrical constraint method as it was presented in [8], provides the ability of using any required macro-element length by adjusting the incremental constraint parameter *c*. In the under study structural part (Pile Cap), the selected incremental parameter c was set equal to 15 so as to utilize us with the ability of using long embedded rebar macro-elements that will initially discretize each reinforcement bar located in the Pile Cap by the use of one or maximum two embedded rebar macro-elements.

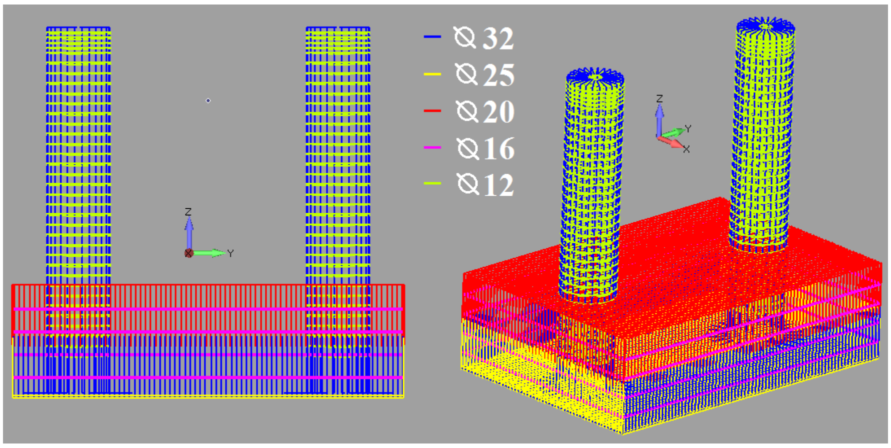


Figure 39. RC Bridge. Embedded rebar mesh of the Pylons and Pile Cap (5,454 embedded rebar macro-elements).

Figs. 38 & 39 show the embedded rebar macro-elements for the case of the reinforcement grid of the two Pylons and the Pile Cap. The total number of embedded rebar macro-elements that were used so as to construct their grid was 5,454 from which the 4,432 are located inside the two Pylons. After the completion of the embedded rebar macro-element mesh of the Pylons and the Pile Cap, the model was analyzed in order to assess the resulted FE mesh. During the embedded mesh generation procedure, for the case of the Pile Cap, the incremental parameter *c* was set equal to 15 and for the case of the Pylons, equal to 5. The total number of the embedded rebar elements that were generated was 21,721.

Fig. 40 shows the deformed shape of the embedded rebar mesh as it derived from the analysis of the model for the self-weight loads. As it can be seen, the mesh generation procedure managed to allocate the embedded rebar elements without any numerical instabilities, while the irregular geometry of the hexahedral elements did not result any numerical issues during the mesh generation procedure. The deformed shape and stress contour shown in Fig. 40 resulted by applying the self-weight of the structure and a distributed load of 1 kN/node at the tip of each Pylon.

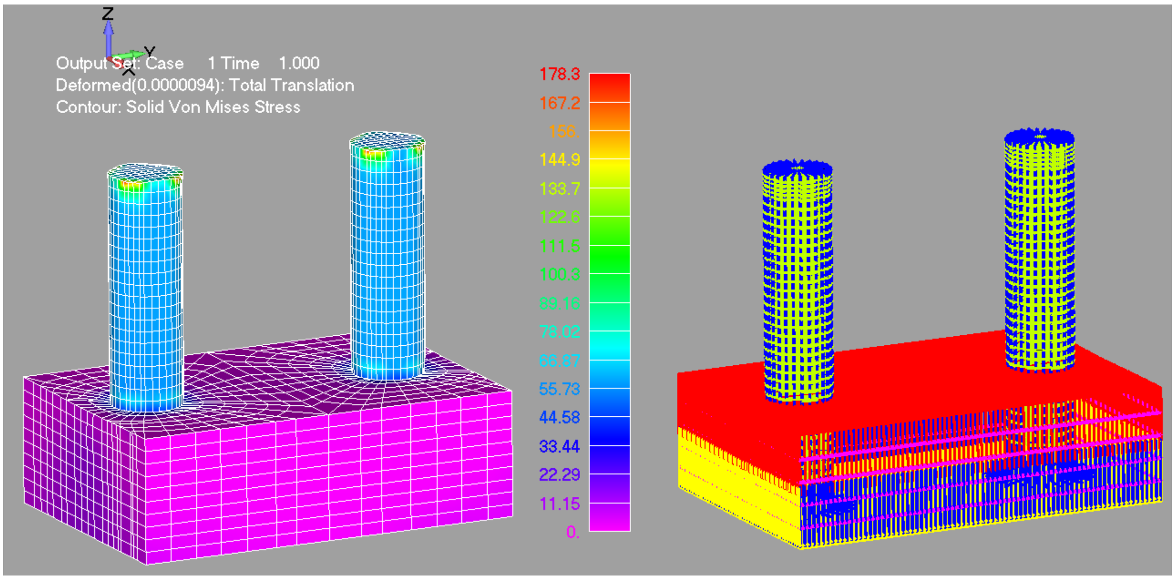

Z
Y
X
Output Set: Case 1 Time 1.000
Deformed(0.0000094): Total Translation
Contour: Solid Von Mises Stress
178.3
167.2
156.
144.9
133.7
122.6
111.5
100.3
89.16
78.02
66.87
55.73
44.58
33.44
22.29
11.15
0.

Figure 40. RC Bridge. (Left) von Mises Stress contour and (Right) Deformed shape of the embedded rebar elements of the two Pylons and the Pile Cap (21,721 embedded rebar elements generated).

Figs. 41-49 show the embedded rebar macro-element meshes that were constructed according to the technical drawings for each structural part of the bridge, while Table 6 shows the distribution of the total number of the embedded rebar macro-elements used to construct the reinforcement grid of the RC bridge. As it can be depicted, the total number of embedded rebar macro-elements used was 47,839 and the total number of embedded rebars that were generated through the Convergence Analaysis by Parts procedure was 520,624. It must be noted here that the parameter *c* was set between 5-30 (Table 6) given that most of the macro-elements were constructed so as to penetrate between 1-30 hexahedral elements.

a/a	Structural Member	*c*	Macro-Elements	Embedded Rebar FEs
1	Span O1 Deck + Left Support Diaphragm (Fig. 36)	15	6,721	70,614
2	Span O2 Deck (Fig. 37)	15	5,092	61,476
3	Span O3 Deck (Fig. 38)	15	5,100	62,954
4	Span O4 Deck (Fig. 39)	15	4,046	50,401
5	Middle Diaphragm (Fig. 40)	30	1,656	25,172
6	Span O5 Deck (Fig. 41)	15	4,601	52,673
7	Span O6 Deck (Fig. 42)	15	4,805	56,416
8	Span O7 Deck (Fig. 43)	15	4,818	55,693
9	Span O8 Deck + Right Support Diaphragm (Fig. 44)	15	5,546	63,504
10	Two Pylons and Pile Cap (Figs. 33 & 34)	5 15	5,454	21,721
		Total	47,839	520,624

Table 6. RC Bridge. Embedded rebar macro-elements and resulted embedded rebar FEs that derived from the procedure of the convergence analysis by parts.

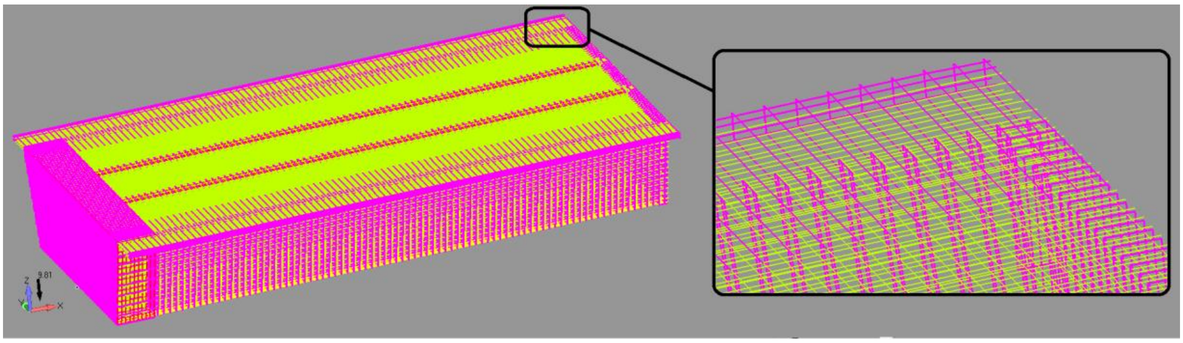


Figure 41. RC Bridge. Macro-element rebar mesh of the left support diaphragm and the opening O1.

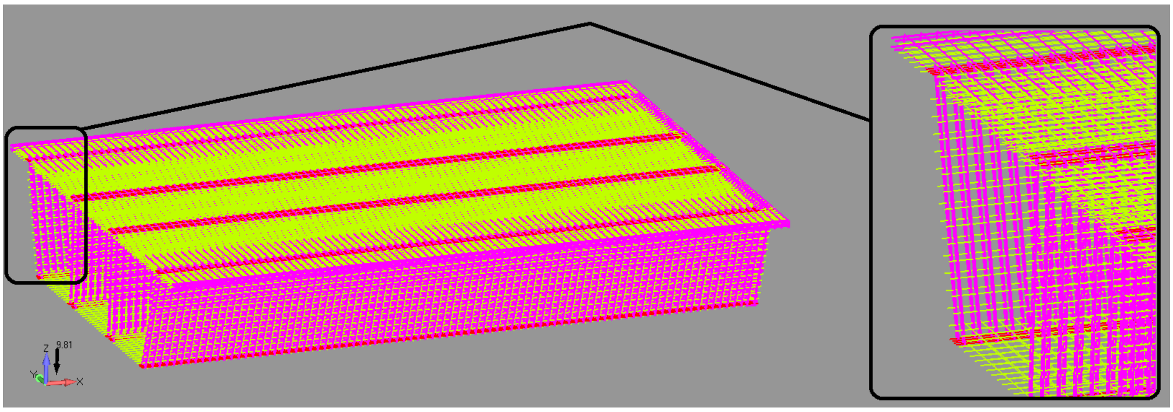


Figure 42. RC Bridge. Macro-element rebar mesh of the opening O2.

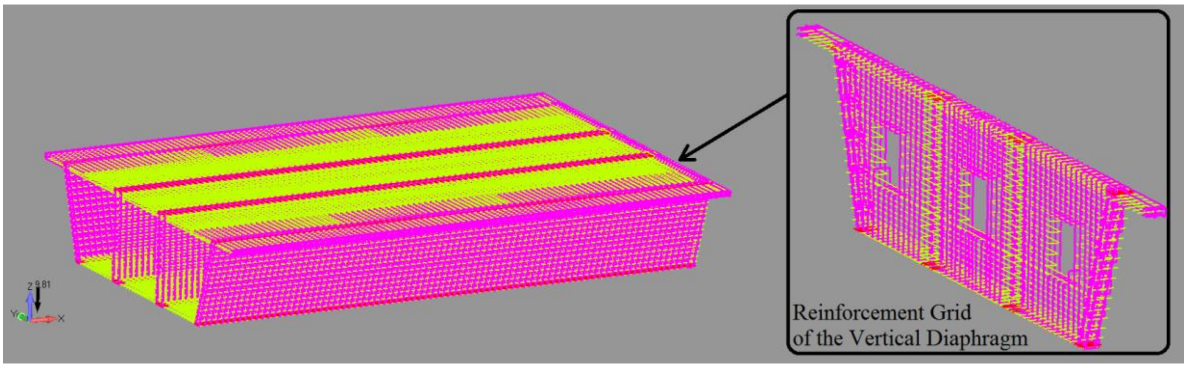


Figure 43. RC Bridge. Macro-element rebar mesh of the opening O3.

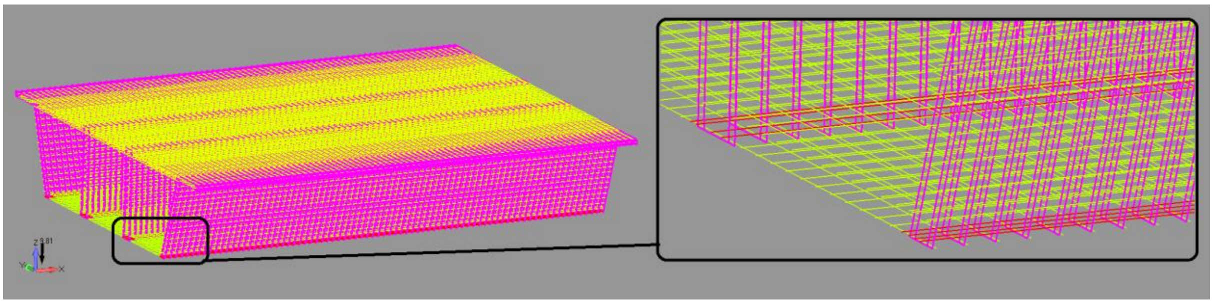


Figure 44. RC Bridge. Macro-element rebar mesh of the opening O4.

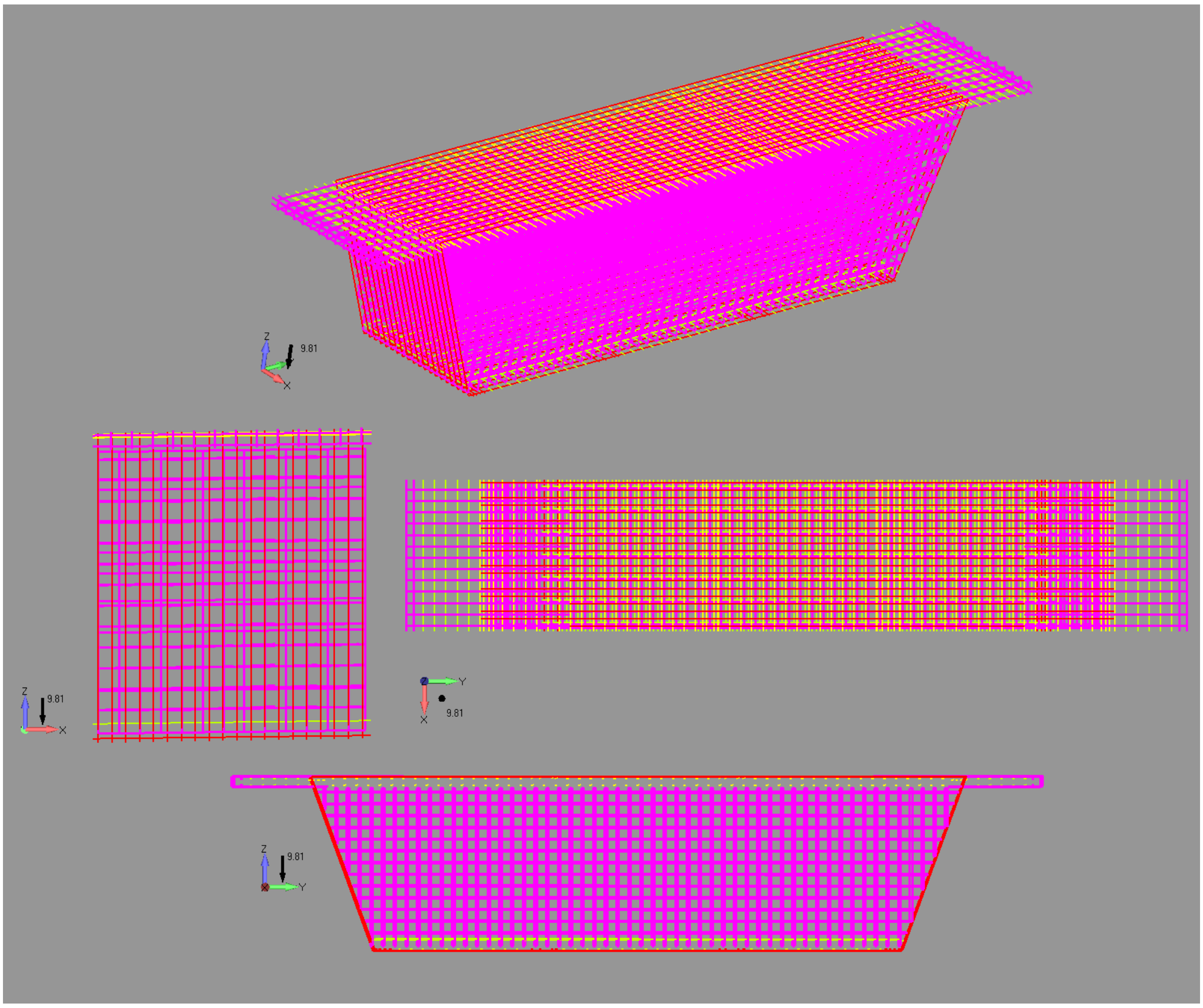


Figure 45. RC Bridge. Macro-element rebar mesh of the middle diaphragm.

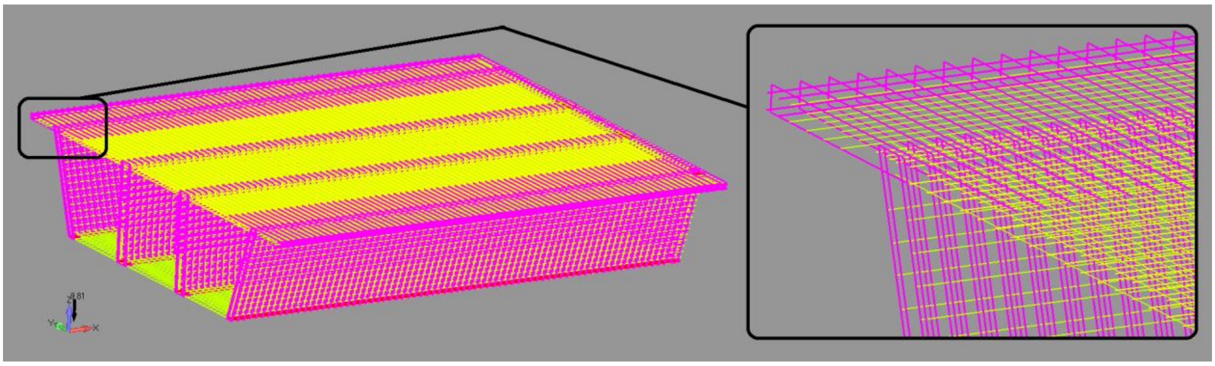


Figure 46. RC Bridge. Macro-element rebar mesh of the opening O5.

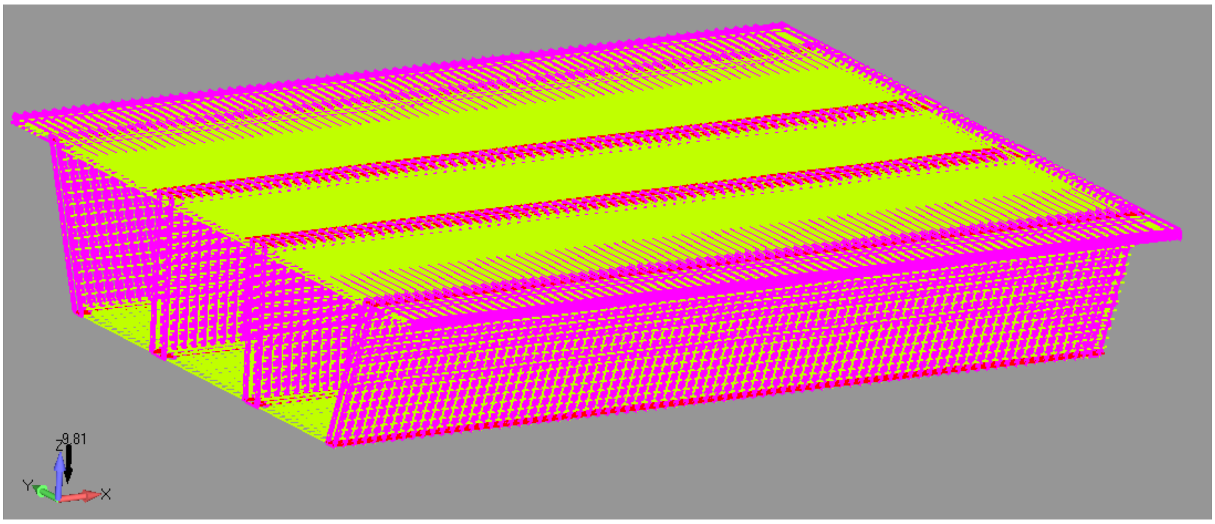


Figure 47. RC Bridge. Macro-element rebar mesh of the opening O6.

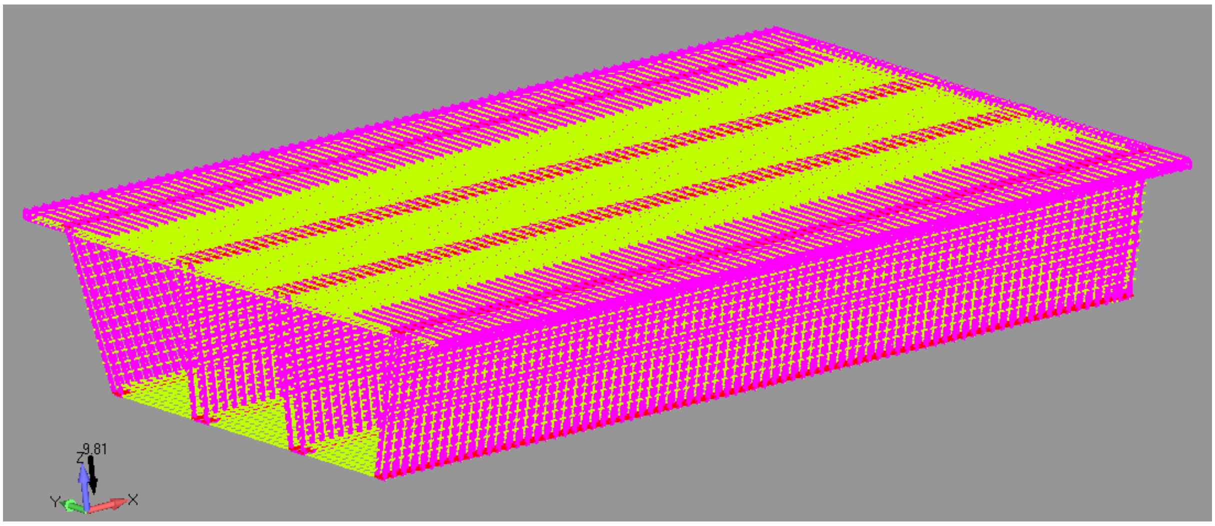


Figure 48. RC Bridge. Macro-element rebar mesh of the opening O7.

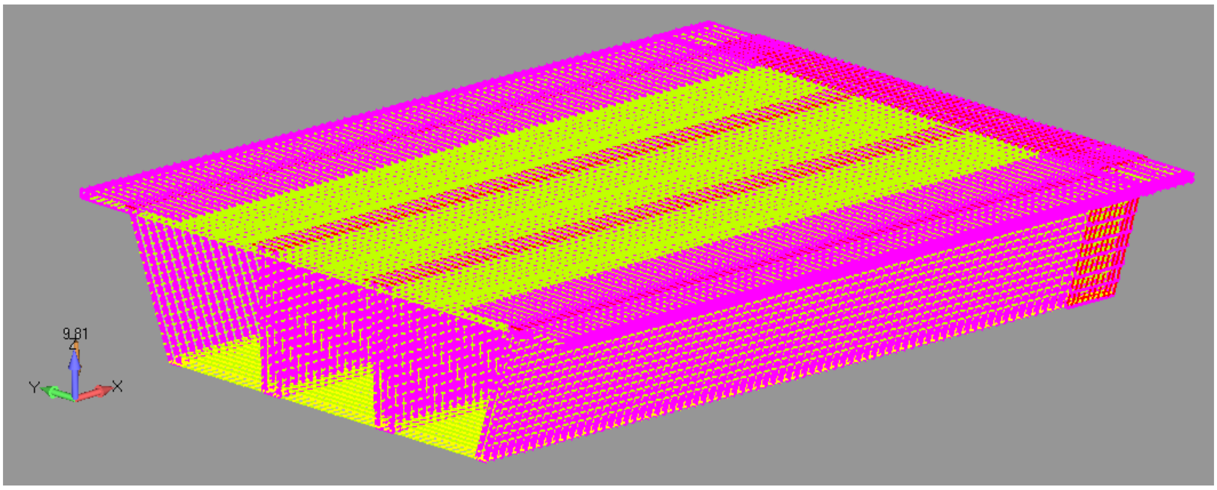


Figure 49. RC Bridge. Macro-element rebar mesh of the right support diaphragm and the opening O8.

4.3.3.3 Short embedded rebar elements

The creation of the reinforcement grid inside the hexahedral elements is a procedure that can induce numerical instabilities that result from the geometry of the solid elements and the arbitrary positioning of the embedded rebar macro-elements. When using an irregular hexahedral concrete mesh as the one shown in Figs. 30 & 31, embedded rebar marco-elements may intersect the hexahedral element in an arbitrary manner. A very interesting numerical phenomenon that results when constructing the embedded macro-element grid for such cases, is the creation of embedded rebar elements with very small lengths. The case of a short embedded rebar element can be depicted in Fig. 50, where the vertical embedded rebar macro-element intersects with the hexahedral element near its tip, resulting this way a very short embedded rebar element. In some cases the length of such embedded rebar elements can be less than 5 mm, while this numerical phenomenon can create the shear locking phenomenon in cases where the rebars are model with the beam FE.

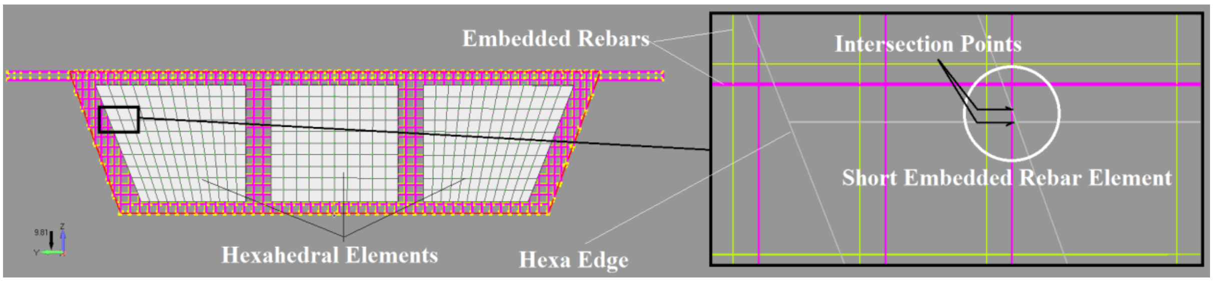
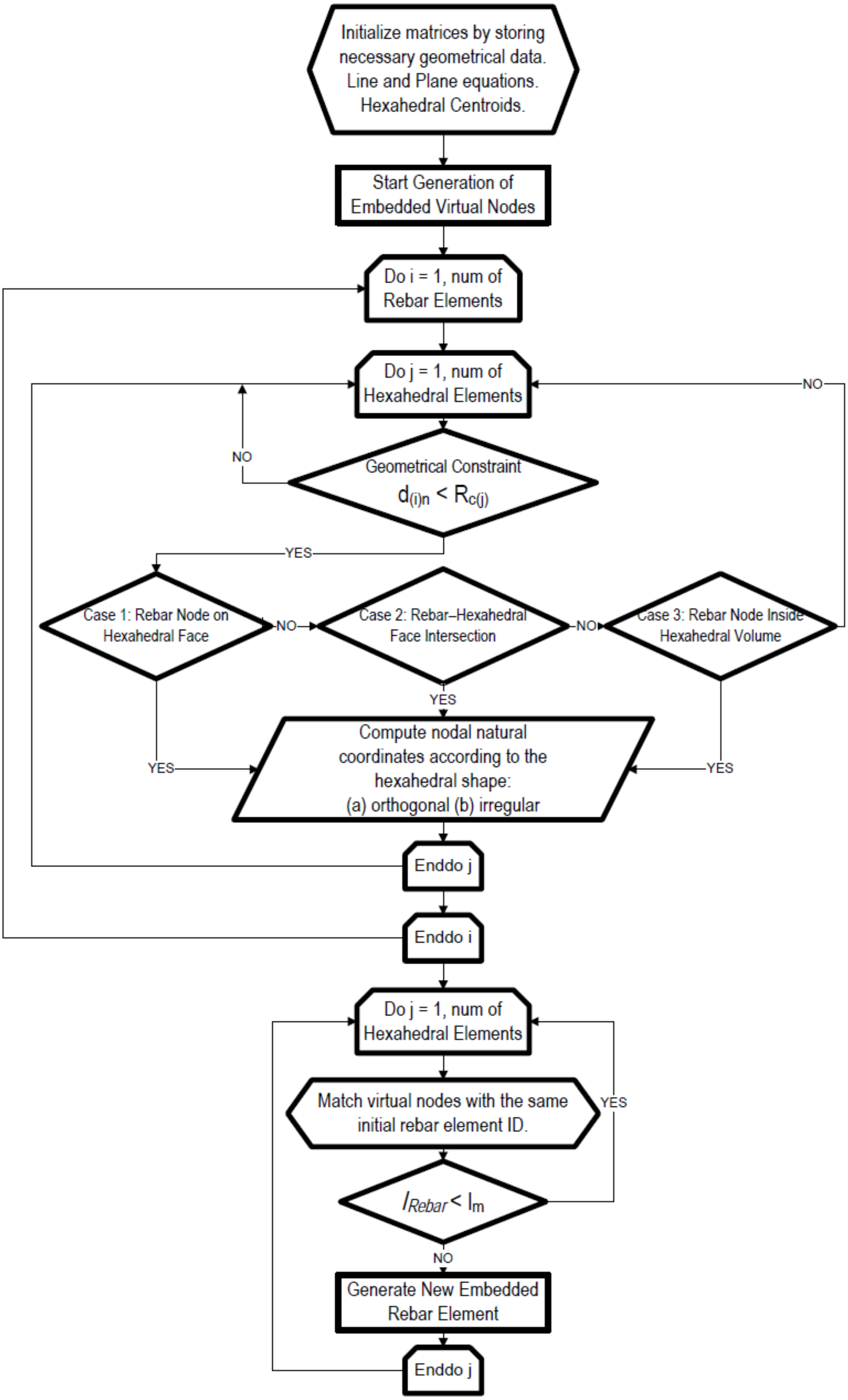


Figure 50. RC Bridge. Hexahedral and embedded rebar elements of the left support diaphragm. Short embedded rebar element case.

Given the fact that perfect bond is assumed during the solution procedure of the model and in an attempt to address the above numerical phenomenon, a filtering algorithm was introduced so as to allocate and control this type of embedded rebars. The proposed filter uses a length criterion which is implemented during the mesh generation procedure and does not allow the creation of embedded rebar elements with a length smaller than l_m. The filtering algorithm was implemented at the final stage of the embedded rebar mesh generation procedure, as it is shown in Fig. 51. For the needs of the under study numerical implementation the geometrical constraint l_m was set to 5 mm, whereas the total number of short embedded rebar elements that were found during the mesh generation procedure of the Convergence Analysis by Parts process was 1,439. The total number of short embedded rebar elements represents 0.28% of the total embedded rebar elements. It must be noted here that Table 6 shows the derived embedded rebar elements excluded the short rebars, thus the total number of embedded rebars that were allocated was 522,063 from which 1,439 were discarded due to their short length.



Initialize matrices by storing necessary geometrical data. Line and Plane equations. Hexahedral Centroids.
Start Generation of Embedded Virtual Nodes
Do i = 1, num of Rebar Elements
Do j = 1, num of Hexahedral Elements
NO
Geometrical Constraint $d_{(i)n} < R_{c(j)}$
NO
YES
Case 1: Rebar Node on Hexahedral Face
NO
Case 2: Rebar–Hexahedral Face Intersection
NO
Case 3: Rebar Node Inside Hexahedral Volume
YES
YES
YES
Compute nodal natural coordinates according to the hexahedral shape: (a) orthogonal (b) irregular
Enddo j
Enddo i
Do j = 1, num of Hexahedral Elements
Match virtual nodes with the same initial rebar element ID.
YES
$l_{Rebar} < l_m$
NO
Generate New Embedded Rebar Element
Enddo j

Figure 51. Flow chart of the updated embedded rebar element mesh generation method.

Practically, the filtering algorithm is implemented by defining the geometric constraint l_m that is assigned to each hexahedral property. The finite element mesh of the hexahedral elements is divided into groups and each group has a solid property. Each solid property can be assigned with a different geometric constraint l_m which is used during the filtering process. Therefore, when defining the solid properties the user can select to activate or deactivate the filtering command by defining $l_m = 0$ and $l_m > 0$, respectively. The filtering algorithm can be used to control the quality of the mesh by scanning for short embedded rebar elements. This way the derived embedded rebar element mesh is checked for any irregularities related to the lengths of the resulted embedded rebars.

At this stage the FE mesh is ready to be solved as a whole. Therefore, the next step is to use the complete finite element mesh and generate the embedded rebar elements through the use of the integrated mesh generation algorithm. The analysis results that will be presented in the next section, assume that the embedded rebar elements are modeled with the rod finite element. In addition to that, for illustrational reasons, the filtering algorithm will be activated and the geometric constraint l_m will be set to 5 mm throughout the hexahedral mesh.

4.3.4 Solution of the Complete Model

At this stage the complete model (Fig. 52) can be used so as to allocate and generate the final embedded rebar mesh through the use of the proposed embedded mesh generation method. Table 7 shows the details of the resulted FE mesh, the total required time for generating the embedded rebar elements and numerical details related to the solution of the FE model.

a/a	Description	Value
1	Number of Hexahedral Elements	102,934
2	Number of Nodes (hexa8 only)	168,400
3	Number of Macro-Elements	47,839
4	Total Number of Embedded Rebar FEs Generated	520,624
5	Total Number of Short Embedded Rebar FEs that were Discarded by the Filter Algorithm	1,439
6	Required Embedded Mesh Generation Time	42 m 22 s
7	Required RAM for the Stiffness Matrix	5.225 Gb
8	Max Required RAM Allocated by the Software	11.5 Gb
9	Computational Time for Solving 1 Load Increment	18 m
10	Computational Time for Writing the Output Data	53 m
11	Total Computational Time: i. Read/Initialize Problem ii. Generate Embedded Mesh iii. Solve the System of Equations for 1 Load Increment / 1 Internal Iteration iv. Write Output Data (out.txt file size: 475 Mb)	118 m

Table 7. RC Bridge. General numerical details that derived after the solution of the complete FE model.

As it can be seen from Table 7 and Fig. 53, the total required time for the embedded mesh generation procedure was 42 minutes 22 seconds. The total number of generated embedded rebar FEs was 520,624 while the total number of short embedded rebar FEs that were discarded due to their small length was 1,439. The deformed shape of the embedded rebar mesh (Fig. 54), resulted by applying only the self-weight of the structure. The solution of the complete model required a total of 11.5 Gb RAM from which the 5.225 Gb was required for the stiffness matrix allocation. So as to solve the model for a single load increment, ReConAn required 18 minutes, while the total computational time for reading, initializing, generating the embedded mesh, solving the system of equations (for 1 load step and 1 internal iteration) and writing the output data, required 118 minutes. The required embedded mesh generation time represents 36% of the total operation time. This ratio is relatively small given that the numerical problem foresees the solution of 1 load increment and 1 internal iteration (linear elastic system of equations) thus decreases the actual computational time that a nonlinear solution procedure would have required. In addition to that, the numerical results illustrate the significance of having a computationally robust and efficient embedded mesh generation procedure.

As it can be seen in Fig. 54, the deformed shape of the embedded rebar mesh was graphically illustrated by increasing the derived displacements 500 times. The deformed shape shows that the embedded rebar mesh follows the sinusoidal deformed shape of the bridge’s deck as it resulted from the numerical solution procedure.

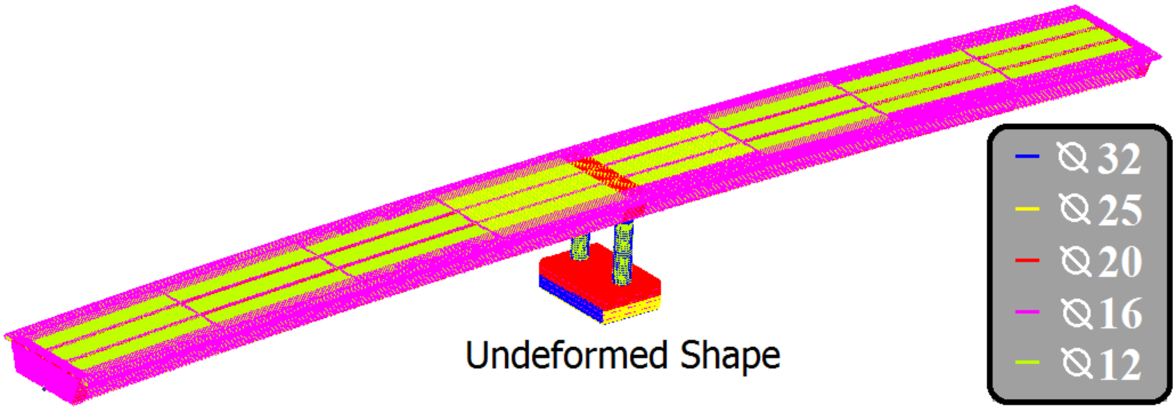
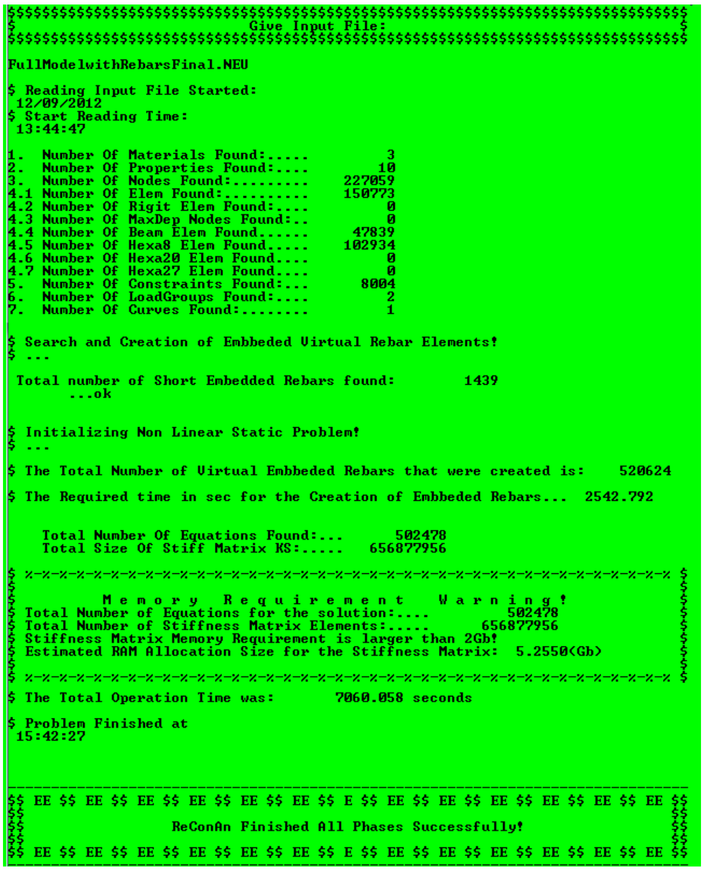


Figure 52. RC Bridge. Macro-element rebar mesh.



```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$                            Give Input File:                                  $
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

FullModelwithRebarsFinal.NEU

$ Reading Input File Started:
 12/09/2012
$ Start Reading Time:
 13:44:47

1.   Number Of Materials Found:.....        3
2.   Number Of Properties Found:....       10
3.   Number Of Nodes Found:.........   227059
4.1  Number Of Elem Found:..........   150773
4.2  Number Of Rigit Elem Found:....        0
4.3  Number Of MaxDep Nodes Found:..        0
4.4  Number Of Beam Elem Found......    47839
4.5  Number Of Hexa8 Elem Found.....   102934
4.6  Number Of Hexa20 Elem Found....        0
4.7  Number Of Hexa27 Elem Found....        0
5.   Number Of Constraints Found:...     8004
6.   Number Of LoadGroups Found:....        2
7.   Number Of Curves Found:........        1

$ Search and Creation of Embbeded Virtual Rebar Elements!
$ ...

 Total number of Short Embedded Rebars found:        1439
       ...ok


$ Initializing Non Linear Static Problem!
$ ...

$ The Total Number of Virtual Embbeded Rebars that were created is:    520624

$ The Required time in sec for the Creation of Embbeded Rebars...  2542.792


    Total Number Of Equations Found:...      502478
    Total Size Of Stiff Matrix KS:.....   656877956

$ %-%-%-%-%-%-%-%-%-%-%-%-%-%-%-%-%-%-%-%-%-%-%-%-%-%-%-%-%-%-%-%-%-%-%-%-%-% $
$                                                                             $
$          M e m o r y   R e q u i r e m e n t   W a r n i n g !             $
$ Total Number of Equations for the solution:....        502478               $
$ Total Number of Stiffness Matrix Elements:.....     656877956               $
$ Stiffness Matrix Memory Requirement is larger than 2Gb!                     $
$ Estimated RAM Allocation Size for the Stiffness Matrix:  5.2550(Gb)         $
$                                                                             $
$ %-%-%-%-%-%-%-%-%-%-%-%-%-%-%-%-%-%-%-%-%-%-%-%-%-%-%-%-%-%-%-%-%-%-%-%-%-% $

$ The Total Operation Time was:       7060.058 seconds

$ Problem Finished at
 15:42:27



--------------------------------------------------------------------------------
$$ EE $$ EE $$ EE $$ EE $$ EE $$ EE $$ E $$ EE $$ EE $$ EE $$ EE $$ EE $$ EE $$
$$                                                                           $$
$$                  ReConAn Finished All Phases Successfully!                $$
$$                                                                           $$
$$ EE $$ EE $$ EE $$ EE $$ EE $$ EE $$ E $$ EE $$ EE $$ EE $$ EE $$ EE $$ EE $$
--------------------------------------------------------------------------------
```

Figure 53. RC Bridge. Information window of ReConAn software.

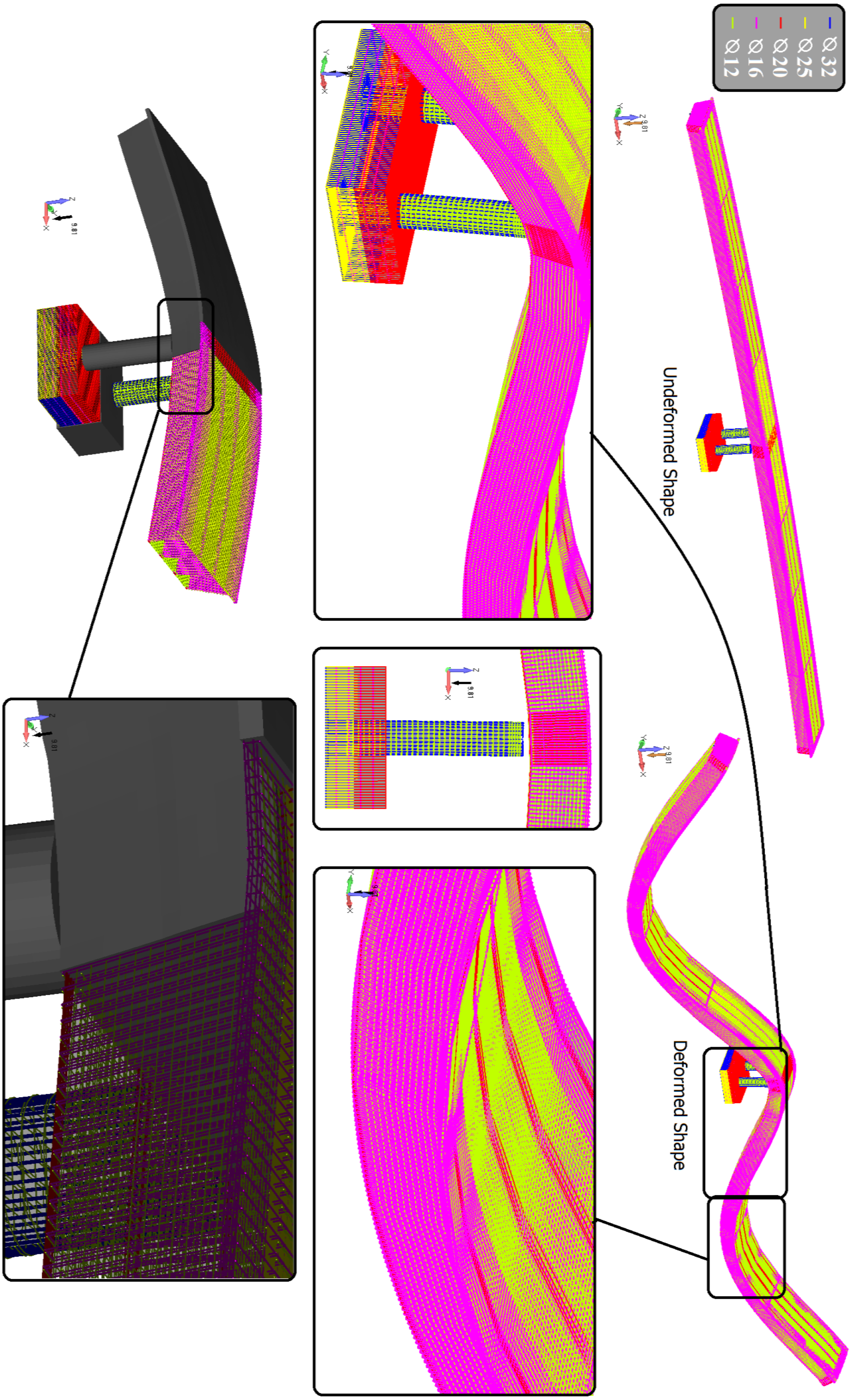


Figure 54. RC Bridge. Deformed shape of the embedded rebar FE mesh.

4.3.5 Double Deck Model

In an attempt to increase the previous FE mesh so as to further investigate the numerical efficiency of the under study mesh generation method, the numerical model that was presented in Fig. 52 is increased by replicating the bridge one time. Fig. 55 shows the new mesh, while in Table 8 the numerical details related to the mesh can be depicted.

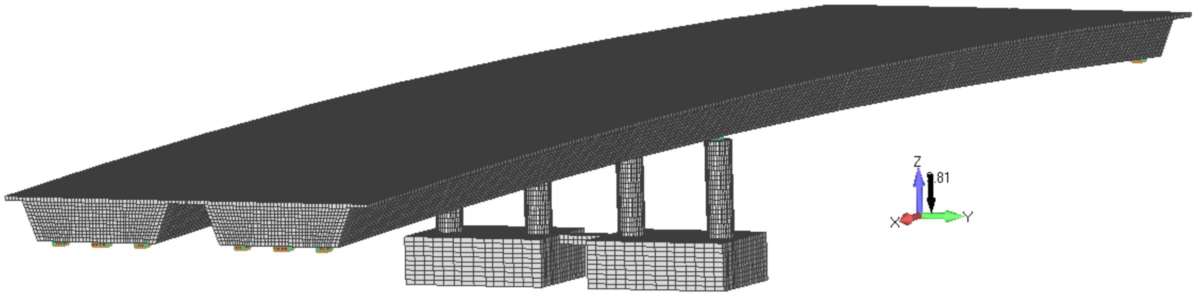


Figure 55. Double Deck RC Bridge. FE mesh of 8-noded hexahedral elements.

a/a	Description	Value
1	Number of Hexahedral Elements	205,928
2	Number of Nodes (hexa8 only)	336,908
3	Number of Macro-Elements	95,082
4	Total Number of Embedded Rebar FEs Generated	1,052,892
5	Total Number of Short Embedded Rebar FEs that were Discarded by the Filter Algorithm	2,878
6	Required Embedded Mesh Generation Time	304 m 5 s
7	Required RAM for the Stiffness Matrix	19.52 Gb
8	Max Required RAM Allocated by the Software	30.0 Gb
9	Computational Time for Solving 1 Load Increment	27 hr
10	Computational Time for Writing the Output Data	10 hr 32 min
11	Total Computational Time: i. Read/Initialize Problem ii. Generate Embedded Mesh iii. Solve the System of Equations for 1 Load Increment / 1 Internal Iteration iv. Write Output Data (out.txt file size: 970 Mb)	48 hr 29 min

Table 8. Double Deck RC Bridge. General numerical details that derived after the solution of the complete FE model.

Table 8 shows the numerical details that derived after the solution of the increased mesh, where it can be seen that the total number of embedded rebar elements that were generated was 1,052,892. If the number of generated embedded rebar elements is compared with the one that resulted from the previous section, then it will result that an additional 11,644 embedded rebar elements were generated. This is attributed to the connection beam that connects the two pile caps as shown in Fig. 55. The embedded mesh generation procedure managed to complete the embedded rebar allocation/generation procedure in 304 minuites and 5 seconds.

The required computational time for the embedded rebar mesh generation procedure represents the10.48% of the total computational time. An important observation that derives by comparing the new ratio with that resulted in the previous section (36%), is that the computational performance of the under study method maintained its efficiency in relation to the solution algorithm that required an excessive time so as to solve a single load increment. This also underlines the need of using a parallel solver that will significantly reduce the computational time for solving the nonlinear numerical problem.

Fig. 56 shows the deformed shape of the model as it resulted from the numerical analysis.

As it can be seen, the embedded rebar elements have the same deformed shape thus follow the deformation of the concrete domain.

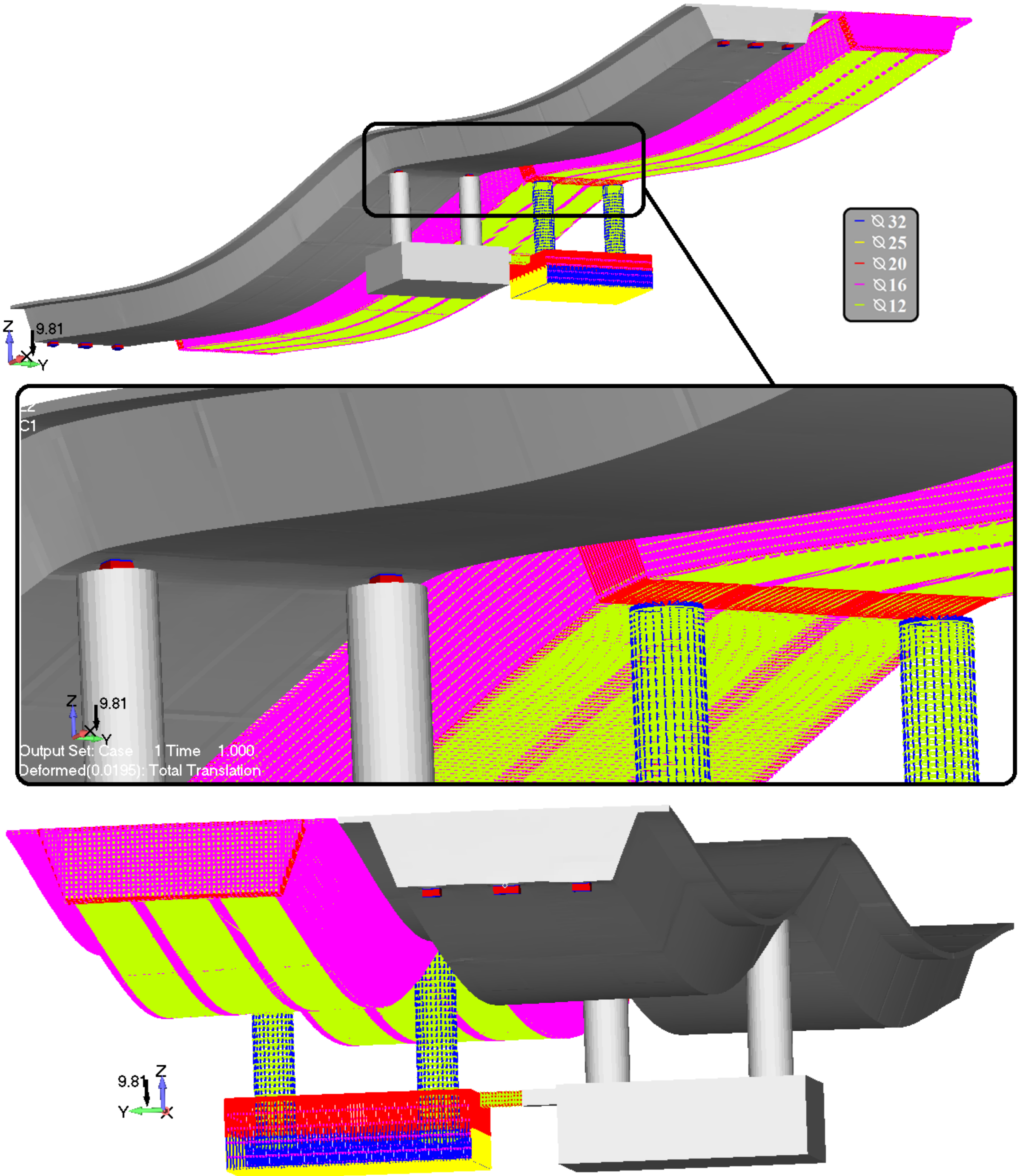

Ø32
Ø25
Ø20
Ø16
Ø12
Z
9.81
X
Y
C1
Output Set: Case 1 Time 1.000
Deformed(0.0195): Total Translation

Figure 56. Double Deck RC Bridge. Deformed shape of the hexahedral and embedded rebar FE meshes.

5 Conclusions

The embedded mesh generation method proposed in [8] was integrated in a 64-bit operating system and was used to generate the embedded rebar meshes of three different finite element models. The parametric investigation performed for the required computational time in generating the embedded rebar elements for the case of the first numerical model (2-storey RC building) revealed that the 64-bit operating system outperforms the corresponding computational capabilities of the 32-bit operating system, while the integrated method managed to generate the embedded mesh in a minimal time (51,064 embedded rebar elements in 12.8 seconds).

Through the analysis of the second numerical model (7-storey RC building) the efficiency and robustness of the integrated embedded rebar mesh generation method was illustrated where 167,824 embedded rebar elements were generated in 183 seconds. This shows the computational efficiency of the method that overcomes previous limitations related to handling relatively large-scale numerical models that use 3D detailed modeling simulations for assessing the mechanical behavior of RC structures.

In order to investigate the limitations of the integrated embedded rebar mesh generation method, the complete model of a RC bridge was constructed and analyzed. The construction of the model eventually derived a total number of 102,934 hexahedral elements and 47,839 embedded rebar macro-elements. Prior to the initiation of the mesh construction, for both hexahedral and embedded rebar macro-elements, a set of 56 and 48 Layers were defined, respectively, in order to manage and control the mesh during the construction phase. This topological sorting of the mesh provided the ability of controlling the bridge's mesh while it made it possible to visualize each part of the bridge separately during the construction of the mesh.

The division of the mesh into 10 main parts (O1-O8 deck openings, middle diaphragm, pylons/pile cap) utilized the mesh construction procedure with the ability of performing the proposed Convergence Analysis by Part procedure for each part of the RC bridge mesh in order to ensure that the final numerical model will be ready to processed thus numerical instabilities will be avoided when the full model was used to generate the final FE numerical model. The at hand numerical model incorporates 623,576 finite elements from which 102,622 are hexahedral concrete FEs that treat the cracking phenomenon with the smeared crack approach. Solving a FE model that incorporated a numerically unstable material formulation with numerical discontinuities is by default a cumbersome procedure even for benchmark problems that assume a limited number of hexahedral elements (100-1000 hexahedral elements). Dealing with a large-scale numerical simulation, controlling the resulted FE mesh from the embedded rebar mesh generation level is of significant importance given that it controls the numerical outcome of the embedded mesh generation and solution procedures.

When constructing the finite element mesh of the RC bridge, it was discovered that the geometrical irregularity of the hexahedral mesh, the dense reinforcement grid and the arbitrary positioning of the embedded rebar macro-elements, derived embedded rebar elements that had relatively short lengths (< 5 mm). This numerical phenomenon can result into the shear locking phenomenon when modeling the embedded rebars with the beam FE. To address this numerical phenomenon, the embedded rebar mesh generation method was upgraded with a filtering algorithm that controls this type of embedded rebar elements and provides the ability of discarding them during the allocation procedure. This filtering algorithm uses a geometrical constraint that does not allow an embedded rebar element generation when the at hand element has a length below the minimum allowable length (l_m). This algorithm can be used in cases where perfect bond slip is assumed and provides the required numerical tools to control the quality of the resulted mesh.

The embedded mesh generation method was used to generate the embedded rebar mesh of the full RC bridge FE model through the use of a 64-bit operating system. After performing the analysis of this numerical implementation it derived that the embedded mesh generation method managed to allocate 522,063 embedded rebar elements from which 1,439 had a length shorter than 5 mm and were excluded from the analysis procedure. The mesh generation procedure required 42 minutes 22 seconds, which illustrates the computational efficiency of the method, while the limitations of the embedded mesh generation algorithm were not practically reached. Given that the total required RAM for solving one load increment was

11.5 Gb and the available RAM of the at hand desktop was 12 Gb, the under-study numerical model managed to reach the hardware’s limitations but not those of the embedded mesh generation method that managed to generate the embedded rebar mesh in a relatively short period of time.

In order to further investigate the computational performance of the under study method, the mesh of the RC Bridge was increased by 2. After the numerical analysis of the increased FE model, which consisted of 205,928 hexahedral concrete FEs and 95,082 embedded rebar macro-elements, the algorithm managed to generate 1,055,770 embedded rebar elements from which 2,878 had a length shorter than 5 mm and were excluded from the analysis procedure. The required computational time for the generation of the embedded rebar elements was 304 m and 5 s. This illustrates the computational efficiency of the embedded rebar mesh generation method which managed to maintain its computational performance in comparison to that of the solver that required more than a day to compute the unknown displacements for a single load increment. Through this numerical implementation it was also concluded that the limitations of the proposed mesh generation methodology were not reached, given that the required computational time was within an acceptable limit, while the solution of the system of equations required significant computational effort.

Finally, it is important to state that modeling RC structures through the use of 3D detailed modeling that discretize the reinforcement grid with embedded rebar elements, has a drawback that relates to the construction of the embedded rebar macro-element mesh, especially in cases where the geometry of the structure has an irregular geometry. This procedure requires a significant effort which makes it prohibitive to be used for commercial purposes. So as to address this issue, an automatic mesh generation algorithm is required to be developed for generating the embedded rebar macro-element mesh inside the hexahedral elements in order to optimize the embedded rebar macro-element mesh construction procedure.

6 Acknowledgements

The author would like to acknowledge the financial support from the Alhosn University of Abu Dhabi and the Vice Chancellor Prof. Abdul Rahim Sabouni for his support throughout this research work.

7 References

[1] Jendele L. and Cervenka J. (2009), “On the solution of multi-point constraints – Application to FE analysis of reinforced concrete structures”, Computers and Structures 87, pp. 970–980.

[2] Kotsovos MD, Pavlovic MN. (1995), “Structural concrete. Finite Element Analysis for Limit State Design”, Thomas Telford: London.

[3] Spiliopoulos KV and Lykidis G. (2006), “An efficient three-dimensional solid finite element dynamic analysis of reinforced concrete structures”, Earthquake Engng Struct. Dyn.35, pp. 137–157.

[4] Hartl H. (2000), “Development of a continuum mechanics based tool for 3d FEA of RC Structures and application to problems of soil-structure interaction” Ph.D. thesis, Faculty of Civil Engineering, Graz Univ. of Technology.

[5] Markou G. (2011), “Detailed Three-Dimensional Nonlinear Hybrid Simulation for the Analysis of Full-Scale Reinforced Concrete Structures”, Ph.D. Thesis, Institute of Structural Analysis and Seismic Research, National Technical University of Athens.

[6] Vladimir Cervenka, (2010), “Large Deflections”, Cervenka Consulting.

[7] Hristovski V.T. and Noguchi H., (2002) “Finite Element Modeling of RC Members Subjected to Shear”, Third DIANA World Conference, Tokyo, Japan 9-11 October 2002.

[8] Markou G. and Papadrakakis M., (2012) “An efficient generation method of embedded reinforcement in hexahedral elements for reinforced concrete simulations”, Advances in Engineering Software ADES, Volume 45(1), Pages 175-187.

[9] Barzegar F. and Maddipudi S., (1994), “Generating reinforcement in FE modeling of concrete structures”, Journal of Structural Engineering 120, pp.1656 –1662.

[10] Elwi A.E. and Hrudey T.M., (1989), “Finite element model for curved embedded reinforcement”, Journal of Engineering Mechanics 115, pp.740 –754.

[11] ASCE Task Committee on Concrete and Masonry Structures. (1982). "Finite element analysis of reinforced concrete." ASCE.

[12] Abdel-Halim M. A. H. and Abu-Lebdeh T. M., (1989), “Analytical study of concrete confinement in tied columns”, J. Struct. Engrg., ASCE, 115(11), pp. 2810-2828.

[13] Gonzalez Vidosa F., Kotsovos M. D. and Pavlovic M. N., (1990). "Three-dimensional finite element analysis of structural concrete”, Proc., Second Int. Conf. on Computer Aided Anal. and Des. of Concrete Struct., N. Bicanic, and H. Mang, eds., Vol. II, Pineridge Press, Swansea, Wales, pp. 1029-1040.

[14] Siemens PLM Software, (2009), “World-class finite element analysis (FEA) solution for the Windows desktop”, Siemens Product Lifecycle Management Software Inc.

Appendix A: Generation of Embedded Rebar Elements

In order to allocate and generate the embedded rebar elements that are located inside each hexahedral element the allocation of the virtual nodes of each embedded rebar element need to be performed. The virtual nodes correspond to the intersections of the rebars with hexahedral faces or edges as shown in Fig. 2 (nodes i1, i2 of the rebar 1-2 and i3 of the rebar 3-4). This procedure becomes cumbersome when the FE model consists of a large number of hexahedral and rebars. It is obvious that if we attempt to compute these possible intersection points without implementing any constraint on the search space, the computational cost of the search algorithm will be significant. The problem arises from the fact that it is required to locate all possible intersections that may exist between hexahedral faces and rebars (Figs. A1 and A2).

To avoid unnecessary calculations, a geometric constraint was introduced in order to restrict the search in the vicinity of the corresponding steel reinforcement. The geometric constraint is implemented with the definition of an active sphere with radius R$_c$:

$$R_c = c \cdot L, \tag{A1}$$

where

$$L = \sqrt{s_x^2 + s_y^2 + s_z^2}$$

and

$$s_x = \frac{\left(\sum_{n=1}^{8}\left(s_x^{cen} - s_x^n\right)\right)}{8}, \quad s_y = \frac{\left(\sum_{n=1}^{8}\left(s_y^{cen} - s_y^n\right)\right)}{8}, \quad s_x = \frac{\left(\sum_{n=1}^{8}\left(s_z^{cen} - s_z^n\right)\right)}{8}$$

with $\mathbf{s}^{\mathrm{n}}$ and $\mathbf{s}^{\mathrm{cen}}$ being the coordinates of node n and the centroid of the hexahedral under consideration and parameter c defines the active volume around each hexahedral where the constraint is implemented. The search for intersection is performed when the below relation is satisfied

$$d_{n1}^{i} \text{ or } d_{n2}^{i} < R_c \tag{A2}$$

where $d_{n1}^{i} = \left| s^{cen} - s_{rebar}^{n1} \right|$, $d_{n2}^{i} = \left| s^{cen} - s_{rebar}^{n2} \right|$ are the distances of the rebar end-nodes 1 and 2 from the hexahedral centroid *i* under consideration, as it is illustrated in Fig. A2. The geometric constraint reduces the computational effort because it restricts the search space considerably during this allocation process especially when dealing with large-scale problems.

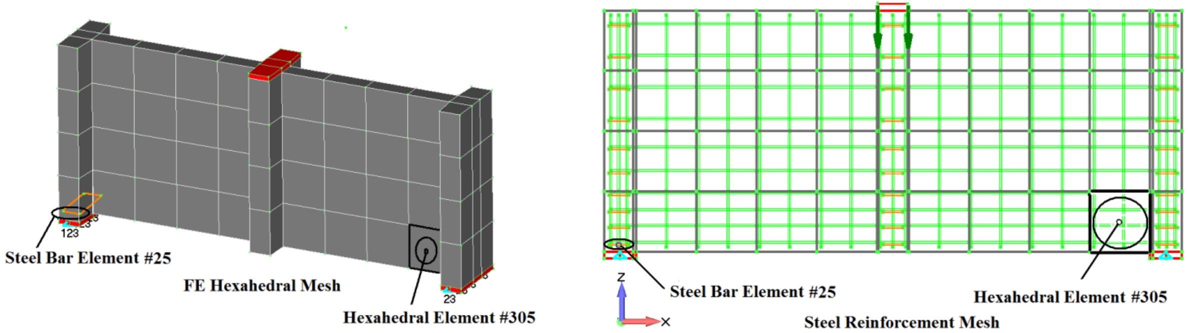
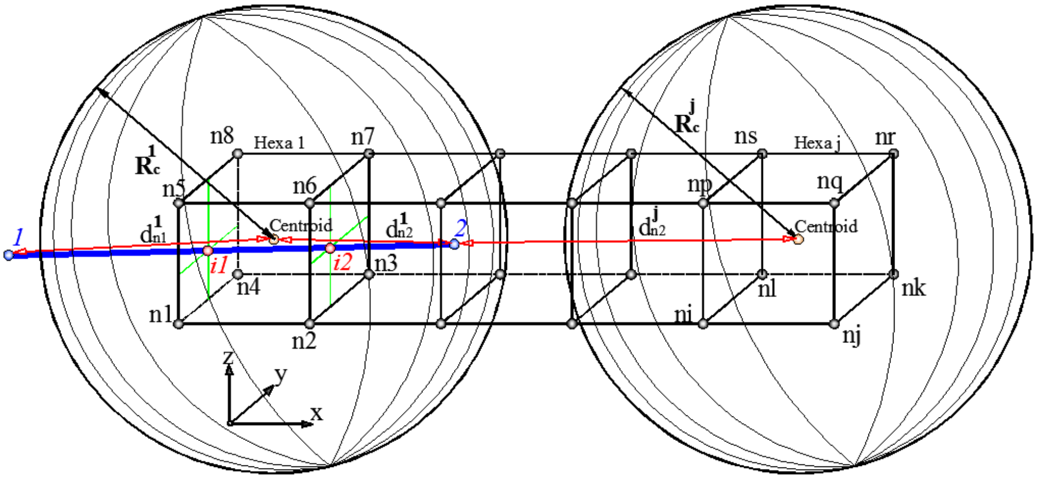


Figure A1. Concrete FE mesh and steel reinforcement rebars of a shear wall [8].



(a) (b)

Figure A2. Geometric constraint for the search of embedded rebar nodes. Geometric constraint is: (a) satisfied: $d_{n2}^1 < R_c^1$; (b) not satisfied: $R_c^j < d_{n2}^i < d_{n1}^j$ [8].

After the satisfaction of the constraint equation (Eq. A2), the generation of the reinforcement rebar elements proceeds according to the following three cases.

Case 1: Initial Rebar Nodes Located on Hexahedral Faces

In this case, a check is performed to detect whether one or both rebar macro-element nodes (1, 2) are located on the hexahedral face(s) (Fig. A3a). If this is the case, then the corresponding node(s) are being denoted as rebar mesh nodes. In order to compute the corresponding nodal local coordinates, the distances dX, dY and dZ between the hexahedral centroid and its first node are required:

$$d_X^{Hexa} = s_x^{Hn1} - s_x^{cen}, \quad d_Y^{Hexa} = s_y^{Hn1} - s_y^{cen}, \quad d_Z^{Hexa} = s_z^{Hn1} - s_z^{cen} \tag{A3}$$

Using Eqs. A3, the distances dX, dY and dZ can be calculated. Then, the natural coordinates of any given point P inside a hexahedral, are given from the following expressions:

$$\xi_P = \frac{s_x^P - s_x^{cen}}{d_X^{Hexa}}, \quad \eta_P = \frac{s_y^P - s_y^{cen}}{d_Y^{Hexa}}, \quad \zeta_P = \frac{s_z^P - s_z^{cen}}{d_Z^{Hexa}} \tag{A4}$$

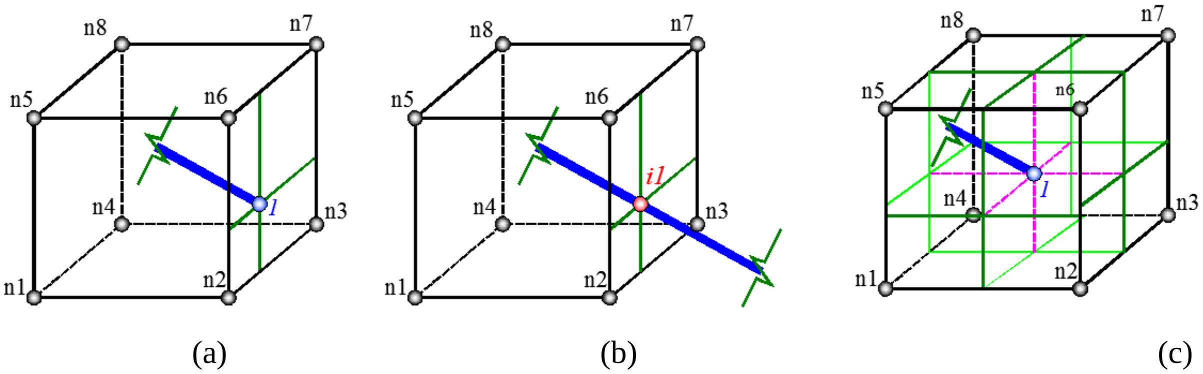


Figure A3. Geometric configuration of: (a) Case 1: Rebar node on hexahedral face, (b) Case 2: Rebar-hexahedral face intersection, (c) Rebar node inside hexahedral volume [8].

If the rebar is located on a hexahedral face, then the proposed algorithm searches for intersections with the hexahedral face edges and creates the corresponding mesh nodes of the rebar finite element (nodes 4, i3 in Fig. 2). For this subcase, the stiffness matrix of the embedded rebar finite element (element ER 3 in Fig. 2) is distributed between the two neighboring hexahedral elements (Hexahedral 2 and 3 in Fig. 2).

For the case of non-prismatic hexahedral elements, the standard Barzegar and Maddipudi [9] method is performed in order to allocate the natural coordinates of the corresponding virtual node as follows: A point P1 with global coordinates $(x, y, z)_{P1}$ on the initial rebar mesh (Fig. A4), is contained in a given concrete element if its natural coordinates $\xi_{P1}, \eta_{P1}, \zeta_{P1}$ satisfy the constraint

$$\left|\xi_{P_1}, \eta_{P_1}, \zeta_{P_1}\right| \leq 1 \tag{A5}$$

associated with this particular hexahedral element.

In the isoparametric formulation the global coordinates (x, y, z) of a generic point within a solid element are expressed as

$$\begin{Bmatrix} x \\ y \\ z \end{Bmatrix} = \begin{bmatrix} \boldsymbol{N} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{N} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{N} \end{bmatrix} \begin{Bmatrix} \boldsymbol{x}_i \\ \boldsymbol{y}_i \\ \boldsymbol{z}_i \end{Bmatrix} \tag{A6}$$

where x_i, y_i, z_i are the global coordinate vectors of the hexahedral nodes and ***N*** represents the row vector of the displacement-shape functions.

$$\begin{Bmatrix} x \\ y \\ z \end{Bmatrix}_{P_1} - \begin{bmatrix} \boldsymbol{N} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{N} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{N} \end{bmatrix} \begin{Bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \\ \boldsymbol{z} \end{Bmatrix} = \boldsymbol{0} \tag{A7}$$

Given that the natural coordinates $(\xi, \eta, \zeta)_{P1}$ are the roots of Eq. A7, a NR iterative procedure is required in order to compute the solution of the above equation as follows:

$$\begin{Bmatrix} \xi \\ \eta \\ \zeta \end{Bmatrix}_{P_1}^{n+1} = \begin{Bmatrix} \xi \\ \eta \\ \zeta \end{Bmatrix}_{P_1}^{n} + \begin{Bmatrix} \Delta\xi \\ \Delta\eta \\ \Delta\zeta \end{Bmatrix}_{P_1}^{n+1} \tag{A8}$$

Since

$$\begin{Bmatrix} d\xi \\ d\eta \\ d\zeta \end{Bmatrix} = \left(\boldsymbol{J}^T\right)^{-1} \begin{Bmatrix} dx \\ dy \\ dz \end{Bmatrix} \tag{A9}$$

where ***J*** is the Jacobian matrix, the incremental natural coordinates are computed from

$$\begin{Bmatrix}\Delta\xi\\ \Delta\eta\\ \Delta\zeta\end{Bmatrix}_{P_1}^{n+1} = \left(\boldsymbol{J}^{n^T}\right)^{-1}\left[\begin{Bmatrix}x\\ y\\ z\end{Bmatrix}_{P1} - \begin{bmatrix}\boldsymbol{N}^n & \boldsymbol{0} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{N}^n & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{N}^n\end{bmatrix}\begin{Bmatrix}\boldsymbol{x}\\ \boldsymbol{y}\\ \boldsymbol{z}\end{Bmatrix}\right] \tag{A10}$$

with

$$\boldsymbol{J}^n = \boldsymbol{J}\left(\xi^n,\eta^n,\zeta^n\right); \quad \boldsymbol{N}^n = \boldsymbol{N}\left(\xi^n,\eta^n,\zeta^n\right).$$

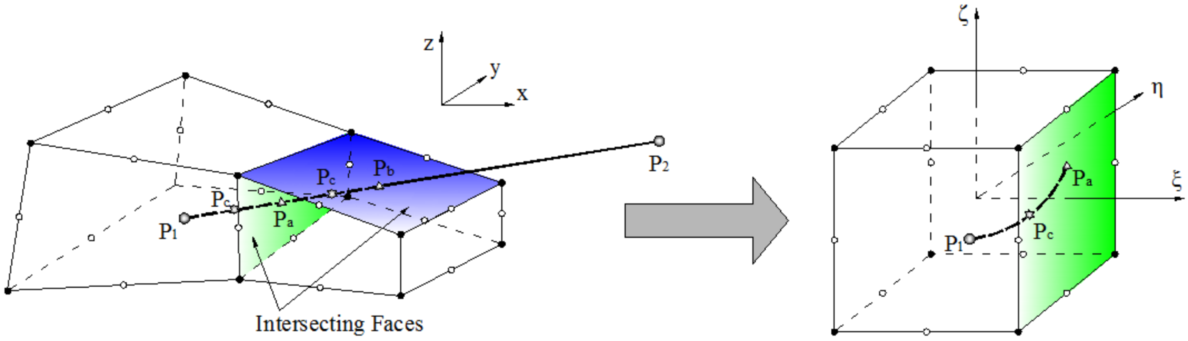


Figure A4. Embedded reinforcement in hexahedral concrete element [8].

Barzegar and Maddipudi [9] found that the preceding solution scheme has a high convergence ratio which was also confirmed in this study. If the converged values do not satisfy (Eq. A5), the procedure proceeds to the next hexahedral element until the geometric constraint is satisfied.

Case 2: Rebar-Hexahedral Face Intersection

In the second case a rebar macro-element intersects one or two hexahedral faces (Figs. A3b, A5). In order to find a potential line-plane intersection the corresponding algebraic equation has to be solved. Then, if an intersection exists, the following constraint is checked, which guaranties that the nodal intersection is located inside the face of the hexahedral under consideration:

$$\left|\xi_P,\ \eta_P,\ \zeta_P\right| \le 1 \tag{A11}$$

where ξ_P, η_P, ζ_P are the natural coordinates of the rebar node obtained either from Eq. A4 or from the Barzegar and Maddipudi [9] procedure, depending on the element's shape. If this constraint is not satisfied, the intersection point is not retained and the algorithm proceeds with the computation of the next intersection point.

Case 3: Rebar Node Inside the Hexahedral Element

When cases 1 and 2 are not applicable then a check is performed for the satisfaction of the following constraint:

$$\left|\xi_R,\ \eta_R,\ \zeta_R\right| < 1 \tag{A12}$$

where ξ_R, η_R, ζ_R are the natural coordinates of the rebar macro-element node. If the above inequality is satisfied, it means that the rebar macro-element node is located inside the volume of the hexahedral (Fig. A3c) and it is stored, otherwise the node is located outside the hexahedral volume and no action is taken. After the computation of the mesh nodes for each of the reinforcement macro-element rebars, the mesh initialization of the embedded rebar elements is performed.

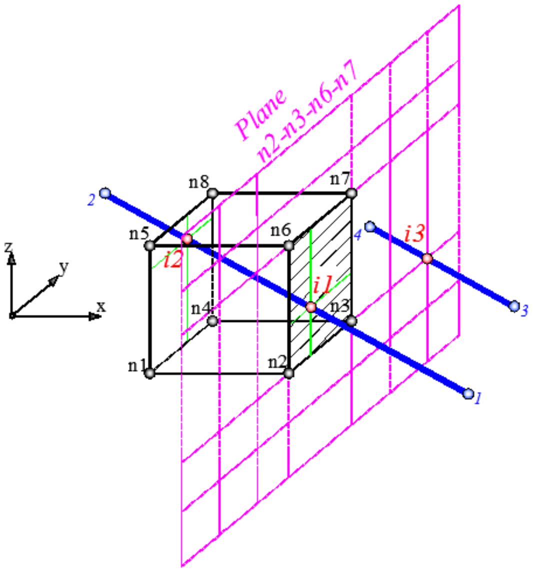


Figure A5. Rebar element intersections with hexahedron faces. Nodes i1 and i2 are retained, node i3 is not acceptable [8].

Following the described generation algorithm, all necessary data of each hexahedral element are determined regarding the corresponding embedded rebar nodes that were located inside the volume or on its faces. At this point, the main features of the embedded rebar elements are calculated and stored: the type of element (Beam or Rod), nodal coordinates and the material properties. Fig. 1 illustrates the complete flow chart of the embedded rebar element mesh generation algorithm.