

A NEW PARALLEL ALGORITHM FOR THE OPTIMUM EMBEDDED REBAR MESH GENERATION FOR LARGE-SCALE REINFORCED CONCRETE STRUCTURES

Dewald Z. Gravett¹ and George Markou¹

¹ Department of Civil Engineering, University of Pretoria, South Africa
e-mail: u16004664@tuks.co.za; George.markou@up.ac.za

Abstract

The modeling of large-scale reinforced concrete (RC) structures through 3D detailed modeling is not feasible due to the excessive computational demand that is required during different phases of the analysis. Especially when the overall objective is to perform nonlinear dynamic analysis of RC structures that also take into account the soil-structure interaction effect by discretizing the soil domain with hexahedral elements. These large-scale models have a significant computational demand when analysed, where the use of high-performance computing and parallel solution algorithms is currently the only solution in achieving reasonable computational times. One of the computationally demanding processes during this type of analysis, is the embedded rebar mesh generation, where millions of elements have to be created. This research work proposes a new load distribution algorithm that optimizes the numerical response of a previous parallel embedded rebar mesh generation algorithm incorporated in Reconan FEA. After the implementation of the proposed parallel algorithm, it was found that the computational time was decreased by an average of 57.48% compared to the performance of the old parallel algorithm.

Keywords: Parallel Processing, Embedded Rebars, Reinforced Concrete Structures, Large-Scale Models, Finite Element Method

1 INTRODUCTION

The finite element (FE) method is widely used to solve and optimize any type of engineering problem. Computationally demanding and complex engineering problems composed of several thousand degrees of freedom are not uncommon anymore, especially when performing assessments with nonlinear dynamic analysis of reinforced concrete (RC) structures such as bridges, nuclear reactor and high-rise buildings that foresee the discretization of the concrete domain through isoparametric hexahedral elements and the reinforcement mesh with embedded rod or beam FEs. This modeling approach is known as 3D detailed approach [1-2], where embedded rebar macro-elements are found within the hexahedral mesh.

A search procedure is required to be performed to find the intersection of the macro-elements [3] with the hexahedral mesh (Fig. 1), to allocate the embedded rebar elements found within each hexahedral element. This procedure can be time consuming for large-scale models that consist of millions of embedded rebar elements. An efficient method for the generation of embedded reinforcement mesh was proposed by Markou and Papadrakakis [4] and parallelized in [5]. This work extends the algorithm proposed in [5], where the computational distribution is optimized.

2 EMBEDDED REBAR MESH GENERATION

Barzegar and Maddipudi [6] proposed the numerical method for the allocation and generation of embedded rebars inside hexahedral elements, which was an extension of the work performed by Elwi and Hruday [7]. According to the hexahedral mesh, the standard Barzegar and Maddipudi [6] method is performed for cases of non-prismatic hexahedral elements, to allocate the natural coordinates of the corresponding virtual node. A point P1 with global coordinates $(x, y, z)_{P1}$ on the initial rebar mesh (Fig. 1), is contained in a given concrete element if its natural coordinates $\xi_{P1}, \eta_{P1}, \zeta_{P1}$ satisfy the following constraint:

$$|\xi_{P1}, \eta_{P1}, \zeta_{P1}| \leq 1 \quad (1)$$

In this formulation the global coordinates (x, y, z) of a common point within a solid element can be expressed as:

$$\begin{Bmatrix} x \\ y \\ z \end{Bmatrix} = \begin{bmatrix} N & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & N & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & N \end{bmatrix} \begin{Bmatrix} x_i \\ y_i \\ z_i \end{Bmatrix} \quad (2)$$

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.

$$\begin{Bmatrix} x \\ y \\ z \end{Bmatrix}_{P1} - \begin{bmatrix} N & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & N & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & N \end{bmatrix} \begin{Bmatrix} x \\ y \\ z \end{Bmatrix} = \mathbf{0} \quad (3)$$

Given that the natural coordinates $(\xi, \eta, \zeta)_{P1}$ are the roots of Eq. 3, a Newton-Raphson iterative procedure is required in order to compute the solution of the above equation as follows:

$$\begin{Bmatrix} \xi \\ \eta \\ \zeta \end{Bmatrix}_{P_1}^{n+1} = \begin{Bmatrix} \xi \\ \eta \\ \zeta \end{Bmatrix}_{P_1}^n + \begin{Bmatrix} \Delta\xi \\ \Delta\eta \\ \Delta\zeta \end{Bmatrix}_{P_1}^{n+1} \quad (4)$$

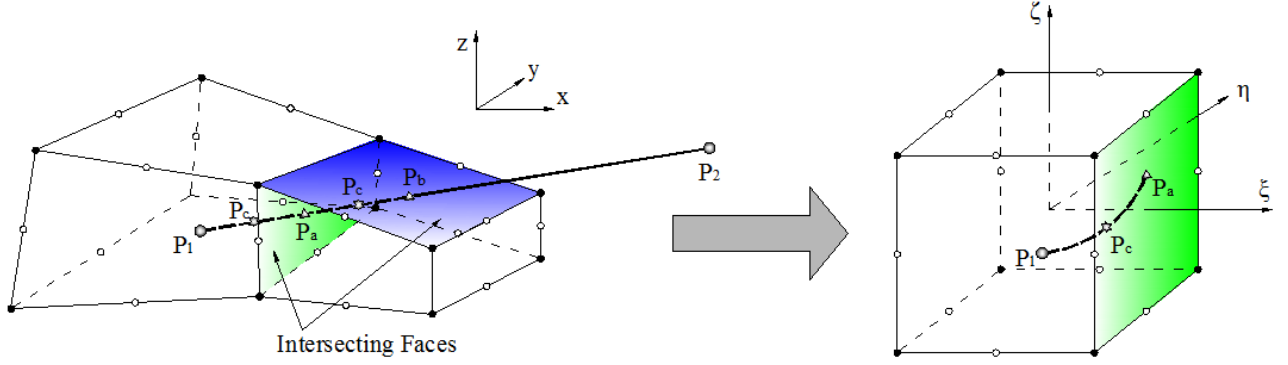


Figure 1. Embedded reinforcement in hexahedral concrete element [3].

Since,

$$\begin{Bmatrix} d\xi \\ d\eta \\ d\zeta \end{Bmatrix} = (\mathbf{J}^T)^{-1} \begin{Bmatrix} dx \\ dy \\ dz \end{Bmatrix} \quad (5)$$

where \mathbf{J} is the Jacobian matrix, the incremental natural coordinates can then be computed from Eq. 6.

$$\begin{Bmatrix} \Delta\xi \\ \Delta\eta \\ \Delta\zeta \end{Bmatrix}_{P_1}^{n+1} = (\mathbf{J}^{nT})^{-1} \begin{Bmatrix} x \\ y \\ z \end{Bmatrix}_{P_1} - \begin{bmatrix} \mathbf{N}^n & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{N}^n & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{N}^n \end{bmatrix} \begin{Bmatrix} x \\ y \\ z \end{Bmatrix} \quad (6)$$

with

$$\mathbf{J}^n = \mathbf{J}(\xi^n, \eta^n, \zeta^n); \quad \mathbf{N}^n = \mathbf{N}(\xi^n, \eta^n, \zeta^n)$$

The mesh generation method proposed by Barzegar and Maddipudi [6] was found to be hindered with additional computational demand, however, this computational demand was reduced and optimized within [3] by implementing a geometric constraint that decreases unnecessary searches of intersection points. Furthermore, the algorithm was further integrated by determining if a hexahedral element is a parallelepiped or not. The proposed mesh generation procedure [3] was then parallelized in [5] as shown in Fig. 2.

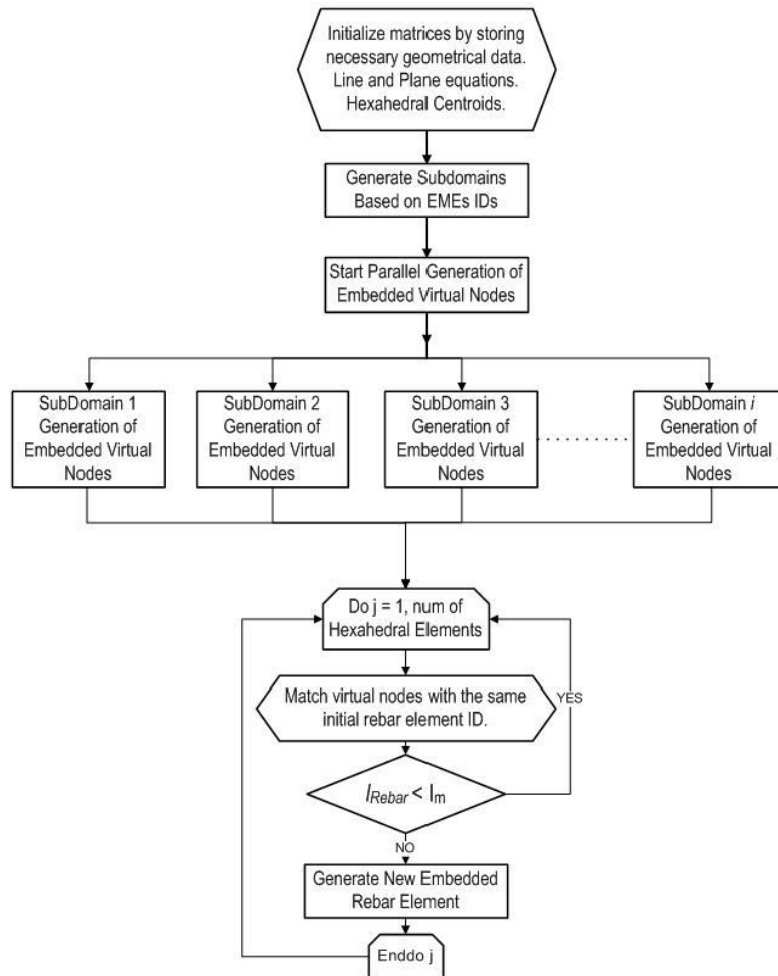


Figure 2. Flow chart of the embedded rebar mesh generation procedure [5].

This research work foresees the development of a parallel algorithm to improve the computational performance and efficiency of the algorithm shown in Fig. 2 for the large-scale embedded rebar mesh generation process.

3 PARALLEL PROCESSING METHODS AND TECHNIQUES

Engineering and scientific computing methods have significantly evolved over the past two decades, with the use of graphic processing units (GPU) in cooperation with a central processing unit (CPU). GPU-accelerated computing has become more popular due to its advanced features related to its parallel architecture, as well as, its capabilities on handling multiple problems simultaneously. Nevertheless, the use of GPUs is constrained from the need of customizing the build according to the hardware at hand. Furthermore, the appearance of the multicore processors brings about an important turning point in programming practices, especially with FE applications.

Fernández et al. [8] presented a new alternate way to formulate the finite element method (FEM) for parallel processing based on the solution of single mesh elements called FEM-SES. This method decouples the solution of a single element from the whole mesh, therefore, exposing parallelism at the element level. Each element solution is then superimposed node-wise using a weighted sum over the concurrent nodes. The research study performed by Fernández et al. [8] showed results up to 111 times faster than the results obtained on classic FEM problems.

Stavroulakis et al. [9] investigated a GPU domain decomposition solution for the spectral stochastic FEM. Within this research work they demonstrate the benefits achieved with the exploitation of the GPU capabilities, where they addressed problems where the solution of the FE algebraic equations was performed with the dual domain decomposition method by implementing specifically tailored preconditioners. This research concluded a significant enhancement of the spectral stochastic FEM, as well as the consumed energy efficiency achieved by this methodology.

Several other researchers [10-14] used parallel processing methods, introducing the basics of implementing parallel processing algorithms enhancing the efficiency and decrease the computational demand of FE software. The only research work found on parallel embedded rebar mesh generation was presented by [5] which is the work that will be used herein in proposing an improved parallel embedded rebar mesh generation framework. The main objective is to significantly improve the scalability and overall computational efficiency of the parallel embedded rebar mesh generation procedure.

4 PROPOSED PARALLEL ALGORITHM

The proposed parallel algorithm is implemented within the research software, Reconan FEA [15]. The initial parallel framework was discussed and tested within [5], by analysing a NuScale reactor building that was investigated in Markou and Filippo [16]. This RC building was discretized with 8-noded hexahedral elements as it is going to be discussed in the next section.

The proposed parallel algorithm suggests a new computational load distribution process that calculates the computational load that will be assigned to each core prior to the mesh generation. The initial parallel algorithm foresaw the division of the number of embedded macro-elements into equal numbers based on their ID. This caused the unequal computational load distribution not allowing the embedded rebar mesh generation procedure to be performed in a computationally optimum manner, especially in cases where the macro-element lengths had significant differences within the model.

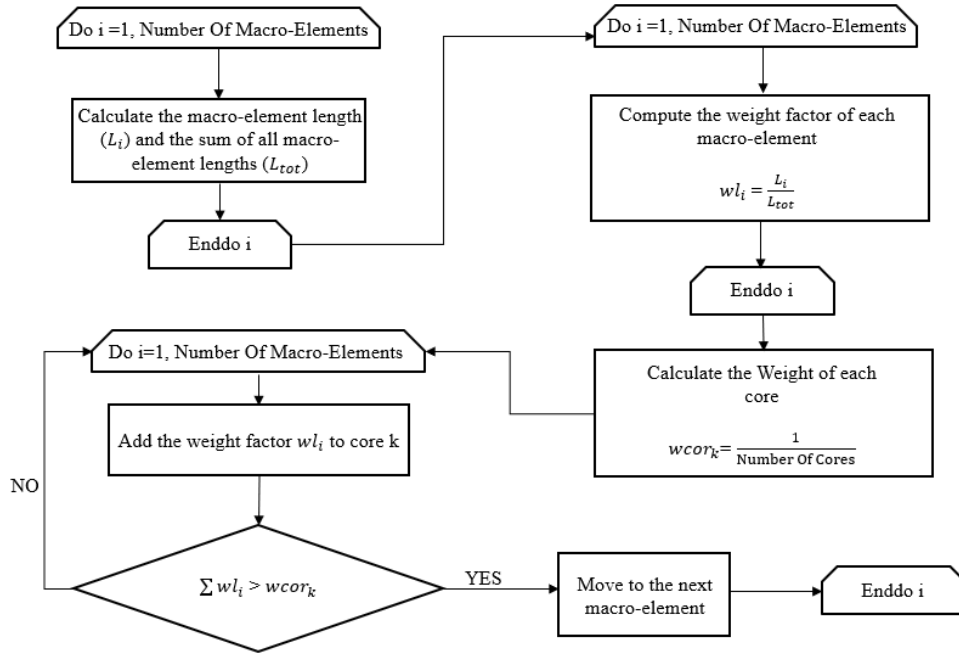


Figure 3. Flowchart for the proposed parallel algorithm.

An improved computational load distribution algorithm was developed for the needs of this research study that is performed based on the geometrical features of the embedded macro-elements found within the mesh. As it can be seen in Fig. 3, the proposed algorithm foresees the computation of the length of each macro-element L_i , while computing the total length of all macro-elements L_{tot} found within the mesh. A weight factor wl_i is computed for each macro-element, which equals to the length of the macro-element L_i divided by the total macro-elements' length:

$$L_{tot} = \sum_{j=1}^{num.elem.} L_j \quad (7)$$

$$wl_i = L_j / L_{tot} \quad (8)$$

Following the calculation of the weight factor wl_i , the computation of the weight factor $wcor_k$ is computed for each core as shown in Fig. 3. This weight factor determines the total length of macro-elements that will be assigned to core k . Therefore, the macro-elements are divided and assigned to each core according to the overall length of the macro-elements that is directly connected to the performed calculations during the search of embedded rebar elements. The longer the macro-element is, the larger the number of hexahedral intersections, thus the larger the calculations required for the allocation of the virtual embedded rebar nodes.

5 REINFORCED CONCRETE MODEL

The NuScale reactor building, developed and studied in [16] is analysed for the needs of investigating the numerical response of the proposed parallel algorithm. This RC model shown in Fig. 4 has a total length of 75.25 m, a width of 30 m and a maximum height of 39.55 m. The FE framing system of the NuScale model was discretized with 181,076 8-noded isoparametric hexahedral elements as it can be seen in Fig. 4. Additionally, to decrease the mesh construction procedure, very long embedded macro-elements were used (up to 75 m in length, where the shortest macro-elements had 0.5 m length). The embedded macro-element mesh can be seen in Fig. 5 as it was developed by Markou and Filippo [16]. It should be noted that this structure has a total of 177,504 embedded macro-elements, where after the execution of the embedded rebar mesh generation procedure, a total of 2,703,251 embedded rebar finite elements are created within the final FE numerical model.

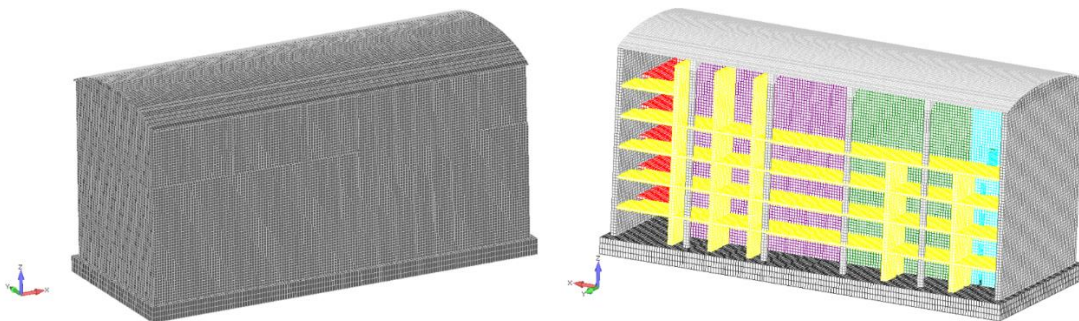


Figure 4: NuScale reactor building discretized with 8-noded isoperimetric hexahedral FEs.

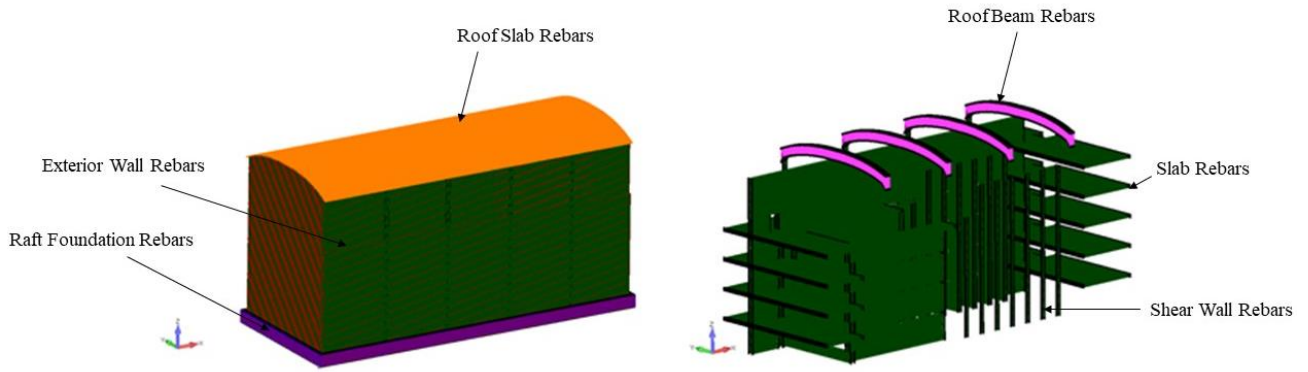


Figure 5. Macro-elements of the NuScale reactor building.

6 NUMERICAL INVESTIGATION OF THE PROPOSED ALGORITHM

The numerical performance of the proposed algorithm was investigated through the use of a 12-core computer, with 4.2 GHz per core computational processing unit. This investigation foresaw the execution of 66 analyses of the NuScale reactor building presented above, in generating the embedded rebar elements. It should be noted at this point that 3 analyses were performed for each solution case using the proposed and the old parallel algorithms by assuming different number of cores, ranging from 2 to 12. When performing the analysis in a serial manner (1 core), the rebar generation computational time equals to 176.93 min. The results obtained from the parallel embedded rebar mesh generations can be seen in Fig. 6.

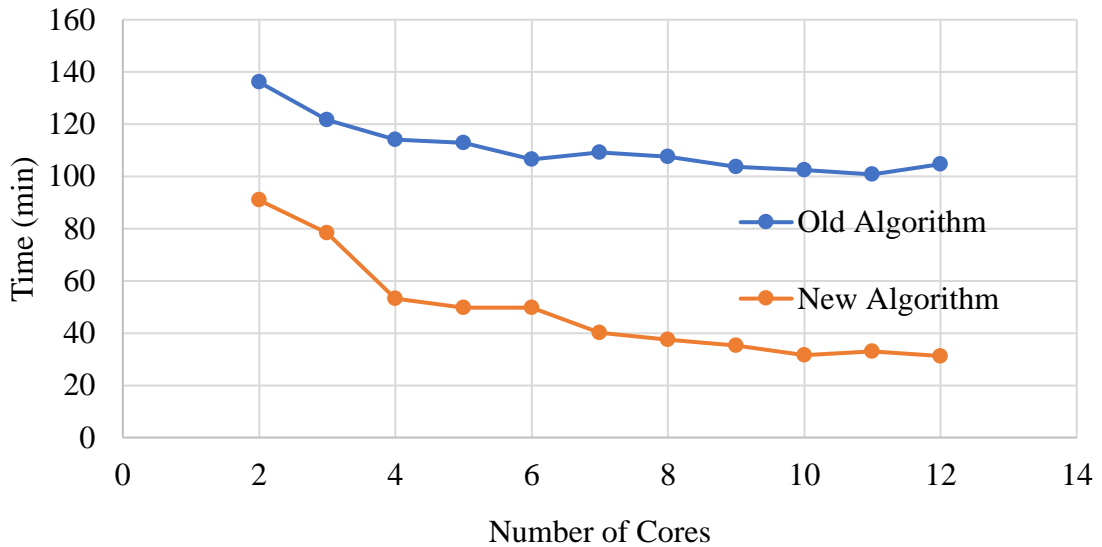


Figure 6: Algorithmic performance results for different number of cores when using the old and new algorithm.

As it can be depicted from Fig. 6, the computational time decreases as the number of cores increases, where the proposed distribution algorithm significantly improved the efficiency of the embedded rebar mesh generation procedure. When the number of cores used to solve the problem is set to 2, the proposed algorithm manages to achieve a 33.2% computational time decrease compared to the old algorithm. Furthermore, it is easy to observe that the new proposed algorithm decreased the computational demand from 104.69 minutes to 31.22 minutes when using 12 cores (see Table 1), which makes the proposed algorithm 3.4 times faster than the old one. As shown in Table 1, the scalability also improves when the number of cores used to solve this problem increases, when the proposed and the old algorithms are compared. In

addition to that, the proposed algorithm manages to decrease the total embedded rebar mesh generation time by 5.7 times when using 12 cores compared to the solution with 1 core.

Table 1: Computational time for each parallel analysis with the old and new algorithm.

	Number of cores										
	2	3	4	5	6	7	8	9	10	11	12
Old Algorithm parallel time (min)	136.10	121.63	114.14	112.81	106.44	109.15	107.59	103.72	102.39	100.76	104.69
New Algorithm parallel time (min)	90.91	78.37	53.26	49.82	49.80	40.11	37.56	35.25	31.52	32.91	31.22
Computational Time Decrease (%)	33.20	35.57	53.34	55.84	53.22	63.26	65.09	66.02	69.21	67.33	70.18

7 CONCLUSIONS

A newly developed and improved distribution load algorithm for parallel processing was presented and numerically investigated through the use of a NuScale reactor building model, consisting a total of 2,703,251 embedded rebar elements. The proposed algorithm took into account the length of each embedded macro-element utilizing a macro-element weight factor to distribute the computational load evenly to each core. Therefore, the distribution of the computational demand was performed based on the total macro-element length sent to each core and not based on the number of embedded macro-elements assigned to each core.

According to the numerical findings obtained from 66 analyses that foresaw the use of different number of cores, it was concluded that the proposed algorithm yielded a significant computational efficiency, allowing the embedded rebar mesh generation procedure to become up to 3.4 times faster compared to the performance of the older parallel algorithm [5].

Currently, the proposed algorithm is parametrically investigated through the use of more large-scale models, such as wind turbine structures, high-rise buildings and bridges. Furthermore, Reconan FEA has been integrated and is now being tested on a high-performance computer which has 40 cores per node. Future research work will investigate more computational load distribution approaches that will further optimize the numerical performance and scalability of the developed embedded rebar mesh generation procedure.

REFERENCES

- [1] Mourlas, C., Markou, G. and Papadrakakis, M., Accurate and Computationally Efficient Nonlinear Static and Dynamic Analysis of Reinforced Concrete Structures Considering Damage Factors, *Engineering Structures*, 178 (2019), pp. 258–285, 2019.
- [2] Mourlas, Ch., Khabele, N., Bark, H.A., Karamitros, D., Taddei, F., Markou, G. and Papadrakakis, M., The Effect of Soil-Structure Interaction on the Nonlinear Dynamic Response of Reinforced Concrete Structures, *International Journal of Structural Stability and Dynamics*, 2020, <https://doi.org/10.1142/S0219455420410138>.
- [3] Markou, G., Computational Performance of an Embedded Reinforcement Mesh Generation Method for Large-Scale RC Simulations, *International Journal of Computational Methods*, 12(3): 1550019-1:48, 2015.
- [4] Markou, G. and Papadrakakis, M., An efficient generation method of embedded reinforcement in hexahedral elements for reinforced concrete simulations, *Advances in Engineering Software ADES*, Vol. 45(1), Pp 175-187, 2012.

- [5] Markou, G., A Parallel Algorithm for the Embedded Reinforcement Mesh Generation of Large-Scale Reinforced Concrete Models, *9th GRACM International Congress on Computational Mechanics*, Chania. Markou G. (2015). Computational Performance of an Embedded Reinforcement Mesh Generation Method for Large-Scale RC Simulations, *International Journal of Computational Methods*, 12(3): 1550019-1:48, 2018.
- [6] Barzegar, F. and Maddipudi, S., Generating reinforcement in FE modeling of concrete structures, *Journal of Structural Engineering*, 120, pp.1656 –1662, 1994
- [7] Elwi, A.E. and Hrudey, T.M., Finite element model for curved embedded reinforcement, *Journal of Engineering Mechanics*, 115:740 –754, 1989.
- [8] Fernández, D.M., Mehri Dehnavi, M., Gross, W.J. and Giannacopoulos, D., Alternate Parallel Processing Approach for FEM, *IEEE Transactions on Magnetics*, VOL. 48, NO. 2, pp 399-402, 2012.
- [9] Stravroulakis, G., Giovanis, D.G., Papadopoulos, V. and Papadrakakis, M., A GPU domain decomposition solution for spectral stochastic finite element method, *Comput. Methods Appl. Mech. Engrg.* 327, pp 392-410, 2017.
- [10] Carey, G. F., Barragy, E., McLay, R. and Sharma, M., Element-by-element vector and parallel computations, *Commun. Appl. Numer. Methods*, VOL. 4, no. 3, pp. 299-307, 1988.
- [11] Okimura, T., Sasayama, T., Takahashi, N. and Ikuno, S., Parallelization of Finite Element Analysis of Nonlinear Magnetic Fields Using GPU, *IEEE Transactions on Magnetics*, vol. 49, no. 5, pp. 1557-1560, 2013.
- [12] Jimack, P.K. and Touheed, N., Developing Parallel Finite Element Software Using MPI, School of Computer studies, University of Leeds, UK, 2014.
- [13] Mpong, S.M., de Montleau, P., Godinas, A. and Habraken, A.M., A Parallel Computing Model for the Acceleration of a Finite Element Software, *PDPTA '02 International Conference*, 2002.
- [14] Cave, W.C., Wassmer, R.E., Ledgard, H.F., Salisbury, A.B., Irvine, K.T. and Mulshine, M.A., A New Approach to Parallel Processing, *IEEE Access*, Vol 8, 2020.
- [15] Reconan FEA v2.0, User's Manual, 2020.
https://www.researchgate.net/publication/342361609_ReConAn_v200_Finite_Element_Analysis_Software_User%27s_Manual
- [16] Markou, G. and Genco, F., Seismic Assessment of Small Modular Reactors: NuScale Case Study for the 8.8 Mw Earthquake in Chile, *Nuclear Engineering and Design*, 342(2019), pp. 176-204, 2019.