Proceedings of the Fifteenth International Conference on
Computational Structures Technology
Edited by: P. Iványi, J. Kruis and B.H.V. Topping
Civil-Comp Conferences, Volume 9, Paper 2.1
Civil-Comp Press, Edinburgh, United Kingdom, 2024
ISSN: 2753-3239, doi: 10.4203/ccc.9.2.1

Computational Procedure for Finite Element Analysis of Functionally Graded Metamaterials

V. H. Yanes Francisco[1] and F. J. Montans Leal[1,2]

[1] Escuela Técnica Superior de Ingeniería Aeronáutica y del Espacio, Universidad Polotécnica de Madrid, Spain
[2] Department of Mechanical and Aerospace Engineering, University of Florida, Florida, USA

Abstract

The extremely fine meshes needed to run problems with details on two scales have promoted several computational homogenization techniques. However, these techniques typically require periodicity of the deformations and the separation of scales. In functionally graded metamaterials, two scales become relevant, but since every metamaterial cell may differ slightly from the neighboring ones, neither the separation of scales nor periodicity conditions may be assumed to hold. Nevertheless, the fact that every metamaterial cell differs only slightly and that we are typically concerned with the global behavior may be considered to speed up the computational procedure. This paper introduces a novel method specifically designed for metamaterials to address these challenges by structuring the metamaterial in macroelements, the stiffness matrix of elements being reduced prior to assembling in the global matrix. The advantage is taken from the fact that macroelement matrices differ only in a few terms. This strategy significantly improves computational efficiency, enabling the handling of very large meshed structures in metamaterials with improved performance compared to conventional FEA techniques.

Keywords: finite element analysis, metamaterials, hierarchical functions, numerical optimization, computational mechanics, macroelement techniques, structural analysis

1 Introduction

Metamaterials, engineered composites with properties not typically found in nature, represent a frontier in material science and engineering. These materials are designed with microstructures that grant them unique mechanical, optical, or electromagnetic properties, such as negative refractive indices or exceptional strength-to-weight ratios [1-6]. The utility of metamaterials spans various industries, including telecommunications, aerospace, and medical devices, offering transformative potential in each.

The advent of advanced 3D printing techniques has further amplified the importance of metamaterials within the industry. These manufacturing technologies facilitate the creation of complex geometries with precision that traditional methods cannot achieve, enabling the practical realization of metamaterial designs that were once purely theoretical. This convergence of design capability and manufacturing flexibility has led to a surge in the exploration and application of metamaterials that push the boundaries of what is possible in material science.

However, the intricate nature of metamaterials poses substantial challenges in their modeling and simulation, primarily due to the computational demands of handling detailed microstructural designs within large-scale simulations. Finite element analysis (FEA), while a robust tool for assessing material behavior, is often hindered by high computational costs when applied to metamaterials, due to the need to manage large mesh sizes and invert massive global stiffness matrices. Our research addresses these challenges by introducing a novel hierarchical function approach tailored for functionally graded metamaterials. This innovative method improves the efficiency and scalability of FEA by significantly reducing the computational load. Crucially, it maintains an accurate representation of the microscale features that are essential for the unique properties of functionally graded metamaterials. In doing so, our approach ensures that simulation efforts keep pace with the rapid advancements in 3D printing and material design, particularly in the context of these specialized materials.

In the following sections, we provide a comprehensive overview of the formulation of our proposed method. In addition, we detail the application of this method to a two-dimensional example, which serves as a reference to demonstrate the efficacy and robustness of our approach. This example will highlight how our method facilitates more efficient and scalable computations, offering a clearer insight into the practical advantages and applications of our technique in the analysis of functionally graded metamaterials. Furthermore, we introduce the concept of enriching the macroelement with higher-order approximation functions, akin to including the gradient of the deformation gradient. This promising idea will be further explored and exploited in our

future works to enhance the fidelity and predictive capabilities of our simulations.

2 New finite element formulation for enhanced multi-scale computations

This section introduces our novel finite element formulation, designed to optimize multiscale computations without necessitating a significant disparity in scale between the micro and macro levels. The formulation capitalizes on the concept of scale separation but does not rely on the traditional assumption that the microscale must be substantially smaller than the macroscale. Instead, it provides a flexible framework that adapts to the varying scales inherent in functionally graded metamaterials, utilizing hierarchical functions to effectively manage internal degrees of freedom within each macroelement.

2.1 Multi-Scale Condensation via Hierarchical Functions (MSC-HF)

To visualize the proposed method, refer to Figure 1, which illustrates a meshed domain. In this representation, the blue nodes represent the internal nodes of the domain, while the red nodes indicate the external nodes. Figure 2 provides a detailed view of the macroelement (pseudo-RVE), adhering to the same color scheme–blue for internal nodes and red for external nodes.

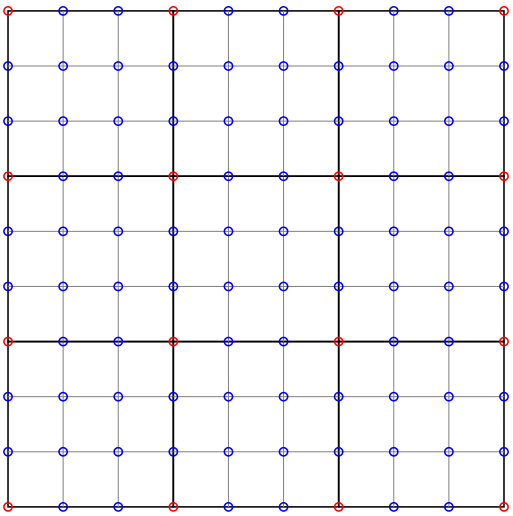

Figure 1: Example of a meshed domain.

A macro element within our model is constructed analogously to a classical RVE but comprises numerous microelements. The stiffness matrix of the macroelement,

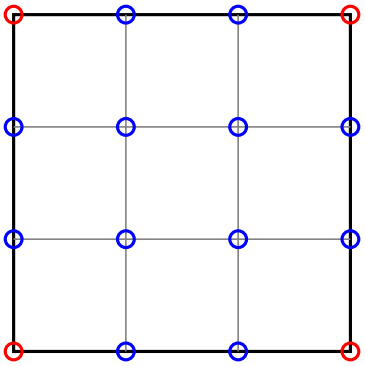

Figure 2: Blow up of a meshed macroelement.

denoted as $\boldsymbol{K}^e$, is a square matrix with dimensions $N_e \times N_e$, where N_e is the number of degrees of freedom of the macroelement, typically large. The respective displacement field is $\boldsymbol{u}^e$ and the forces applied to the macroelement are $\boldsymbol{f}^e$. The relationship between the forces and the displacement field is expressed by the equation:

$$\boldsymbol{K}^e \boldsymbol{u}^e = \boldsymbol{f}^e. \tag{1}$$

The complete system is assembled as follows:

$$\left(\bigwedge_{e=1}^{N_e} \boldsymbol{K}^e \right) \boldsymbol{U} = \boldsymbol{F} \quad \text{with} \quad \bigwedge_{e=1}^{N_e} \boldsymbol{u}^e = \boldsymbol{U} \quad \text{and} \quad \bigwedge_{e=1}^{N_e} \boldsymbol{f}^e = \boldsymbol{F}, \tag{2}$$

Due to the large number of elements N_e, this results in a significantly large system.

To further elaborate on the internal function of the macroelement, consider a displacement function $u(x, y)$, which may represent u_x or u_y. Within the macroelement, this function is approximated using shape functions of the microelement, resulting in a piece-wise bilinear function $u^e(\xi, \eta)$, where u^e encapsulates the nodal values and (ξ, η) are the macroelement coordinates. This function can be further approximated by a different set of functions as shown:

$$u(\xi, \eta) \simeq \sum_{I=1}^{N_\psi} \Psi_I(\xi, \eta) v_I \approx \boldsymbol{\psi}^{\mathrm{T}} \boldsymbol{v}, \tag{3}$$

where $\Psi_I(\xi, \eta)$ are the approximation functions and v_I are the function multipliers.

Consider the simplest linear case in one dimension, where the displacement $u_e(\xi)$ within the macroelement can be expressed as a linear combination of the nodal displacements, u_1 and u_2, and a series of higher-order terms that enhance the approximation:

$$u^e(\xi) = h_1(\xi) u_1 + h_2(\xi) u_2 + \sum_{i=3}^{n_\psi} \psi_i(\xi) v_i^e, \tag{4}$$

where $h_1(\xi)$ and $h_2(\xi)$ are the linear shape functions:

$$h_1(\xi) = \frac{1}{2}(1-\xi), \tag{5}$$

$$h_2(\xi) = \frac{1}{2}(1+\xi), \tag{6}$$

The functions $\psi_i(\xi)$ are chosen such that they vanish at the element boundaries, ensuring u_1 and u_2 remain the primary nodal values:

$$\psi_i(-1) = \psi_i(1) = 0 \tag{7}$$

This choice is particularly advantageous because it maintains the nodal values of the element while allowing for an enriched representation within the element through the addition of hierarchical shape functions. These functions, often derived from Legendre polynomials, are modified to be zero at the nodes, thus satisfying the boundary conditions and contributing to the element's internal representation [7-9].

In the context of this work, we opted for harmonic functions to construct the series of higher-order terms in equation (4). Harmonic functions are particularly suited for this purpose due to their orthogonality and smoothness, which can effectively capture the displacement field within the element while satisfying boundary conditions:

$$\psi_i(\xi) = \begin{cases} \cos\left(\frac{(i-2)\pi\xi}{2}\right) & \text{for odd } i > 2, \\ \sin\left(\frac{(i-2)\pi\xi}{2}\right) & \text{for even } i > 2. \end{cases} \tag{8}$$

Harmonic functions, as described in equation (8), offer the advantage of providing a complete and orthonormal basis in the function space of the element. A graphical representation of the standard shape functions augmented by the four first harmonic functions can be found in Figure 3.

For the 2D case under consideration, the approximation functions are defined as follows:

$$\psi_{ij}(\xi,\eta) = \begin{cases} \text{linear 2D shape functions} & \text{for } i,j \in [1,2] \\ g(\xi,i) * g(\eta,j) & \text{for } i,j > 2, \end{cases} \tag{9}$$

where $g(x, idx)$ are the harmonic 1D approximation functions described in equation (8). A graphical representation of the harmonic shape functions $(i,j) = (3,3)$, $(i,j) = (3,4)$ and $(i,j) = (4,3)$ can be found in Figure 4.

Let us now return to the interpolation of displacements within the macroelement, utilizing the defined 2D approximation functions

$$\boldsymbol{u}^e(\xi,\eta) \simeq \boldsymbol{\psi}(\xi,\eta)\boldsymbol{v}^e. \tag{10}$$

For a macroelement with N nodes and N_ψ approximation functions, the displacement

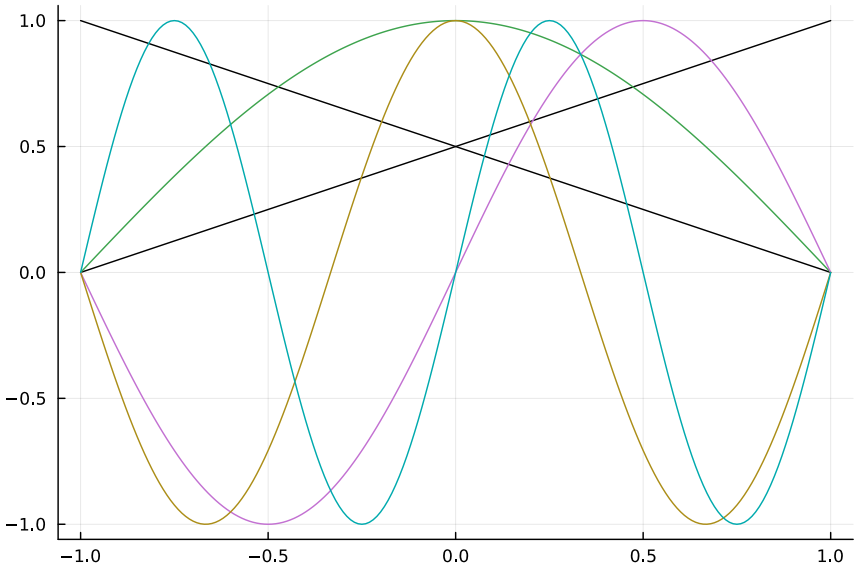

Figure 3: 1D standard functions augmented by the four first harmonic functions.

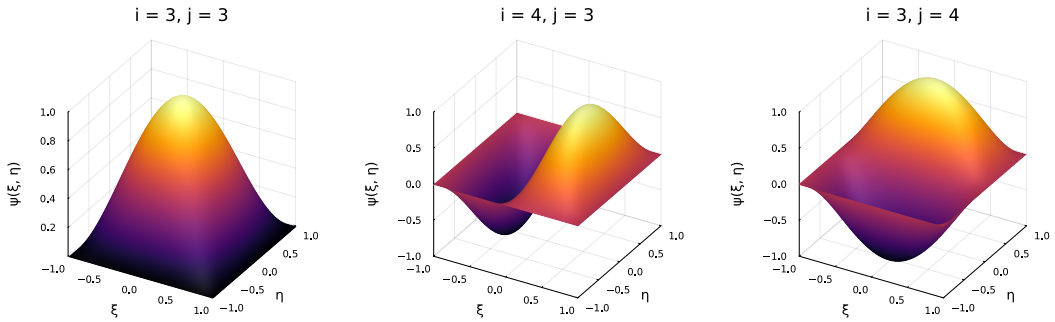

Figure 4: 2D harmonic functions (3, 3), (4, 3) and (3, 4).

vector has the form:

$$\begin{bmatrix} u_{1x} \\ u_{1x} \\ u_{2x} \\ u_{2y} \\ \vdots \\ u_{Nx} \\ u_{Ny} \end{bmatrix} = \begin{bmatrix} \psi_{11} & 0 & \psi_{12} & 0 & \dots & \psi_{1N_\psi} & 0 \\ 0 & \psi_{11} & 0 & \psi_{12} & \dots & 0 & \psi_{1N_\psi} \\ \psi_{21} & 0 & \psi_{22} & 0 & \dots & \psi_{2N_\psi} & 0 \\ 0 & \psi_{21} & 0 & \psi_{22} & \dots & 0 & \psi_{2N_\psi} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \psi_{N1} & 0 & \psi_{N2} & 0 & \dots & \psi_{NN_\psi} & 0 \\ 0 & \psi_{N1} & 0 & \psi_{N2} & \dots & 0 & \psi_{NN_\psi} \end{bmatrix} \begin{bmatrix} v_{1x} \\ v_{1x} \\ v_{2x} \\ v_{2y} \\ \vdots \\ v_{N_\psi x} \\ v_{N_\psi y} \end{bmatrix}.$$

The displacement field of the macroelement can be partitioned in boxes:

$$\boldsymbol{u}^e = \begin{bmatrix} \boldsymbol{\psi}_S & \boldsymbol{\psi}_H \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_S^e \\ \boldsymbol{v}_H^e \end{bmatrix}. \tag{11}$$

Using Equation (10) in conjunction with Equation (1), and considering Equation (11):

$$\begin{aligned} \boldsymbol{K}^e \boldsymbol{u}^e &= \boldsymbol{f}^e \\ \boldsymbol{K}^e \boldsymbol{\psi} \boldsymbol{v}^e &= \boldsymbol{f}^e \\ \boldsymbol{K}^e \boldsymbol{\psi} \boldsymbol{v}^e &= \boldsymbol{f}^e \\ \boldsymbol{\psi}^{\mathrm{T}} \boldsymbol{K}^e \boldsymbol{\psi} \boldsymbol{v}^e &= \boldsymbol{\psi}^{\mathrm{T}} \boldsymbol{f}^e. \end{aligned} \tag{12}$$

Being

$$\boldsymbol{K}_{**}^e = \boldsymbol{\psi}^{\mathrm{T}} \boldsymbol{K}^e \boldsymbol{\psi} \tag{13}$$

the projected stiffness matrix in the space of the orthonormalized approximation functions.

Following from Equation (12)

$$\begin{bmatrix} \boldsymbol{\psi}_S^{\mathrm{T}} \\ \boldsymbol{\psi}_H^{\mathrm{T}} \end{bmatrix} \boldsymbol{K}^e \begin{bmatrix} \boldsymbol{\psi}_S & \boldsymbol{\psi}_H \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_S^e \\ \boldsymbol{v}_H^e \end{bmatrix} = \begin{bmatrix} \boldsymbol{\psi}_S^{\mathrm{T}} \\ \boldsymbol{\psi}_H^{\mathrm{T}} \end{bmatrix} \boldsymbol{f}^e. \tag{14}$$

Note that the forces inside the macroelement are typically null; therefore, Equation (14) can be written as:

$$\begin{bmatrix} \boldsymbol{\psi}_S^{\mathrm{T}} \boldsymbol{K}^e \boldsymbol{\psi}_S & \boldsymbol{\psi}_S^{\mathrm{T}} \boldsymbol{K}^e \boldsymbol{\psi}_H \\ \boldsymbol{\psi}_H^{\mathrm{T}} \boldsymbol{K}^e \boldsymbol{\psi}_S & \boldsymbol{\psi}_H^{\mathrm{T}} \boldsymbol{K}^e \boldsymbol{\psi}_H \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_S^e \\ \boldsymbol{v}_H^e \end{bmatrix} = \begin{bmatrix} \boldsymbol{\psi}_S^{\mathrm{T}} \boldsymbol{f}^e \\ \boldsymbol{0} \end{bmatrix}. \tag{15}$$

In compact form

$$\begin{bmatrix} \boldsymbol{K}_{SS}^e & \boldsymbol{K}_{SH}^e \\ \boldsymbol{K}_{HS}^e & \boldsymbol{K}_{HH}^e \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_S \\ \boldsymbol{v}_H \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_S^e \\ \boldsymbol{0} \end{bmatrix}. \tag{16}$$

Note that $\boldsymbol{K}_{SS}^e$, $\boldsymbol{K}_{SH}^e$, $\boldsymbol{K}_{HS}^e$, $\boldsymbol{K}_{HH}^e$ and $\boldsymbol{f}_S^e$ are the projections of the stiffness matrix and the forces over the space of the approximation functions $\boldsymbol{\psi}$ defined in Equation (9).

From the system of Equations (14), the multipliers of the hierarchical functions $(\boldsymbol{v}_H)$ can be condensed, and system is reduced to

$$\left(\boldsymbol{K}^e_{SS} - \boldsymbol{K}^e_{SH}\left(\boldsymbol{K}^e_{HH}\right)^{-1}\boldsymbol{K}^e_{HS}\right)\boldsymbol{v}^e_S = \boldsymbol{f}^e_S. \tag{17}$$

The term between parentheses is the condensed macroelement stiffness and is the matrix to be assembled into the global system of equations.

$$\bar{\boldsymbol{K}}^e_{SS} = \boldsymbol{K}^e_{SS} - \boldsymbol{K}^e_{SH}\left(\boldsymbol{K}^e_{HH}\right)^{-1}\boldsymbol{K}^e_{HS}. \tag{18}$$

The dimensionality of the problem is reduced from $2N \times 2N$ to $2N_{\text{macro}} \times 2N_{\text{macro}}$, where N, the number of nodes, is much larger than N_{macro}, the number of macronodes. The computationally expensive operation in this reduced system is given by:

$$\bar{\boldsymbol{K}}^e_{HS} = \left(\boldsymbol{K}^e_{HH}\right)^{-1}\boldsymbol{K}^e_{HS}. \tag{19}$$

This reduction highlights the need for efficient computational strategies to manage the complexities involved. The subsequent subsection will be devoted to explaining effective methodologies for performing this condensation, aiming to optimize computational resources and ensure accurate results.

2.2 Strategies for efficient matrix condensation in macroelement analysis

Note that the matrix $\boldsymbol{\psi}$ is predefined for a given RVE and remains immutable. Furthermore, the stiffness matrix $\boldsymbol{K}^e$ of the macroelement can be constructed using various finite elements, such as triangles or squares, treating it as if it were an independent structure. Each macroelement must be indexed if microelements are included within its structure. The projected matrix K^e_{**}, given by Equation (13), is carried forward from one macroelement to the next and is modified only when a microelement is added. This modification is a key advantage of the method, as it involves operations only with the newly added components.

$$\boldsymbol{K}^e_{HH} = \boldsymbol{K}^{e-1}_{HH} + \psi^{\mathrm{T}}_H\boldsymbol{\delta}\boldsymbol{\psi}_H, \tag{20}$$

$$\boldsymbol{K}^e_{HS} = \boldsymbol{K}^{e-1}_{HS} + \psi^{\mathrm{T}}_H\boldsymbol{\delta}\boldsymbol{\psi}_S, \tag{21}$$

$$\boldsymbol{K}^e_{SS} = \boldsymbol{K}^{e-1}_{SS} + \psi^{\mathrm{T}}_S\boldsymbol{\delta}\boldsymbol{\psi}_S, \tag{22}$$

where

$$\boldsymbol{\delta} = \boldsymbol{K}^e - \boldsymbol{K}^{e-1}. \tag{23}$$

Notice that if the changes from macroelement to macroelement are small then $\boldsymbol{\delta}$ is a sparse matrix, with most entries being zeros. Additionally, the multiplication of a sparse matrix by a dense matrix is highly optimized in most programming languages, making this a very fast computation.

The number of hierarchical approximation functions required typically does not need to be extensive, as these primarily account for the internal degrees of freedom. For most engineering applications, approximately 10 harmonic functions are sufficient to adequately capture the internal structure of the macroelement. The computation of $\bar{\boldsymbol{K}}^e_{HS}$ as outlined in Equation (19) can efficiently be performed using direct LU decomposition.

It is important to remark that the interpolation of the external nodes of the macroelement is conducted using linear functions. However, in cases where increased precision is necessary to more accurately represent the contribution of the internal structure of the macroelements, a larger number of approximation functions may be required. Under such circumstances, the following iterative process is proposed to enhance the model's computational efficiency.

If the changes from macroelement to macroelement are small, then we can propose

$$\bar{\boldsymbol{K}}^{e-1}_{HS} - \bar{\boldsymbol{K}}^{e}_{HS} \to 0, \tag{24}$$

$$(\boldsymbol{K}^{e-1}_{HH})^{-1}\boldsymbol{K}^{e-1}_{HS} - (\boldsymbol{K}^{e}_{HH})^{-1}\boldsymbol{K}^{e}_{HS} \to 0, \tag{25}$$

$$\boldsymbol{K}^{e}_{HH}(\boldsymbol{K}^{e-1}_{HH})^{-1}\boldsymbol{K}^{e-1}_{HS} - \boldsymbol{K}^{e}_{HS} \to 0, \tag{26}$$

denoting

$$X^e \equiv (K^e_{HH})K^e_{HS} \tag{27}$$

and replacing into Equation (26)

$$\boldsymbol{K}^{e}_{HH}X^{e-1} - \boldsymbol{K}^{e}_{HS} \to 0, \tag{28}$$

$$(\boldsymbol{K}^{e-1}_{HH} + \Delta\boldsymbol{K}^{e}_{HH})X^{e-1} - \boldsymbol{K}^{e}_{HS} \to 0, \tag{29}$$

$$\boldsymbol{K}^{e}_{HS} = (\boldsymbol{K}^{e-1}_{HH} + \Delta\boldsymbol{K}^{e}_{HH})X^{e-1}, \tag{30}$$

$$\boldsymbol{K}^{e}_{HS} = \boldsymbol{K}^{e-1}_{HS} + \Delta\boldsymbol{K}^{e}_{HH}X^{e-1}, \tag{31}$$

$$(\boldsymbol{K}^{e}_{HS})^{-1}\boldsymbol{K}^{e}_{HS} = (\boldsymbol{K}^{e}_{HS})^{-1}(\boldsymbol{K}^{e-1}_{HS} + \Delta\boldsymbol{K}^{e}_{HH}X^{e-1}), \tag{32}$$

$$X^e = (\boldsymbol{K}^{e}_{HS})^{-1}(\boldsymbol{K}^{e-1}_{HS} + \Delta\boldsymbol{K}^{e}_{HH}X^{e-1}). \tag{33}$$

2.3 Enriching the macroelement

A bilinear macroelement is analogous to a first-order FE^2 approach, where a continuum deformation gradient is transferred to the microscale to derive the resultant stresses.

However, the current procedure can accommodate higher-order approaches, similar to incorporating the gradient of the deformation gradient. The concept involves utilizing quadratic macroelements. For these elements, standard 1D shape functions are

designed with a third node positioned centrally.

$$h_1(\xi) = \frac{1}{2}\xi(\xi - 1), \tag{34}$$

$$h_2(\xi) = \frac{1}{2}\xi(\xi + 1), \tag{35}$$

$$h_3(\xi) = (1 + \xi)(1 - \xi). \tag{36}$$

The standard functions for this case include the terms constant, linear, and quadratic. Therefore, the hierarchical functions will add from cubic terms, but need to banish at the nodes (see Figure 5). This requirement ensures continuity and compatibility at the interfaces between elements within the finite element mesh.

$$\psi_i = \sin{(i - 3)\pi\xi}. \tag{37}$$

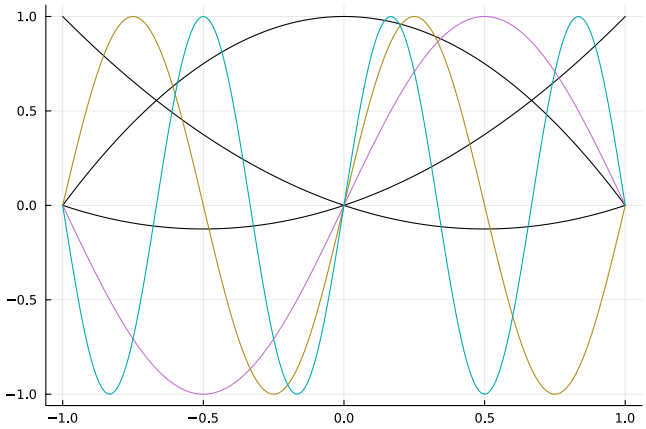

Figure 5: Second order approximation functions in 1D.

For the 2D case, a combination of sine and cosine harmonic approximation functions can be used, provided that the combination includes at least one sine function. This inclusion is essential to meet the compatibility criteria, ensuring that the functions satisfy the necessary boundary conditions at the nodes and interfaces of the elements.

3 Benchmarking

To validate the effectiveness and applicability of our theoretical models, we have conducted a series of benchmark tests using various configurations of metamaterials. The creation of each metamaterial involves specifying the layout and mechanical characteristics that define the material's structure and response properties. This process includes setting the overall dimensions of the metamaterial and preparing the detailed arrangement of its constituent microstructures. These microstructures are arranged in a pattern that alternates between solid and void elements.

The fill pattern used in our metamaterial testing is designed with an "'X" shaped macroelement that is progressively filled from the outer edges toward the center. This pattern simulates a scenario where the exterior of the material is exposed to environmental influences or structural loads before the interior, mimicking the conditions often encountered in real-world applications such as protective coatings or architectural elements.

Figure 6 illustrates a detailed representation of one of the metamaterial configurations tested during our benchmarking study.

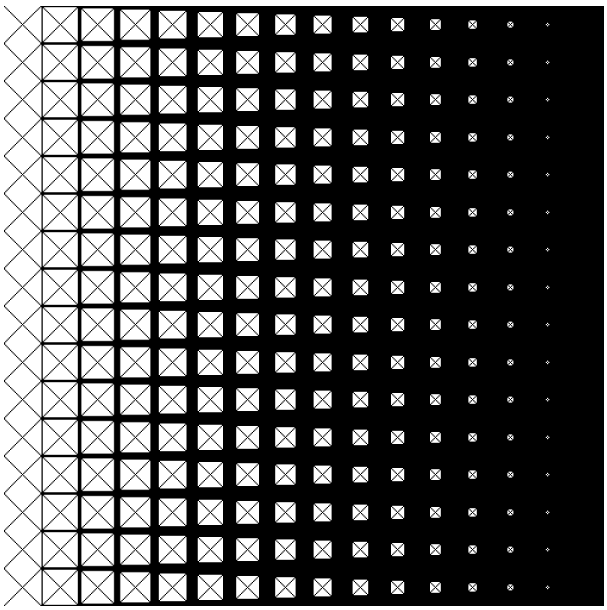

Figure 6: Example of a tested metamaterial with 246016 microelements.

To demonstrate the differences in performance between the proposed method and the traditional FEM, we have conducted a series of benchmark tests. These tests compare memory usage and computational times for each method across various configurations of element numbers. The resulting data, as presented in Table 1, offer valuable insights into the efficiency and scalability of each method. This comparison is essential for identifying potential advantages and determining the suitability of the condensed method for various engineering applications.

Number of Elements	Condensed Method		Classic FEM	
	Memory	Time	Memory	Time
36,100	304.78 MiB	787.700 ms	1.65 GiB	5.632 s
53,361	462.22 MiB	1.144 s	2.43 GiB	8.170 s
76,176	672.43 MiB	1.916 s	3.47 GiB	11.079 s
105,625	946.65 MiB	2.770 s	4.94 GiB	20.380 s
142,884	1.27 GiB	3.207 s	6.54 GiB	23.435 s
189,225	1.69 GiB	3.951 s	8.73 GiB	29.655 s

Table 1: Benchmark Comparisons: Condensed Method vs. Classic FEM.

4 Concluding remarks

The paper proposes a novel FEM that uses the characteristics of functionally graded metamaterials. By efficiently dual-scaling the problem into micro- and macro-elements, we manage to project and condense the stiffness matrix of each macroelement using a set of hierarchical functions, which are harmonic in this case. This technique significantly improves computational efficiency by reducing the complexity of the matrix operations involved.

The benchmarks demonstrate that this method outperforms traditional FEM approaches in terms of both computational time and memory usage. Especially in large-scale computations, our method shows a marked advantage, making it a valuable tool for simulations that require high efficiency. This method not only facilitates a more detailed understanding of the material behaviors at different scales but also optimizes resource utilization, which is critical in complex engineering tasks involving functionally graded materials.

The advancements introduced in this study provide a robust framework for further development and application in various engineering fields, particularly where the properties of materials vary spatially. Future research will continue to refine this approach and expand its application to more complex geometries and loading conditions.

Acknowledgements

VY and FJM would like to acknowledge the financial and research support provided by the XS-Meta project of the Marie Słodowska-Curie research fellowship program.

Funding

VY and FJM acknowledge the funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 956401.

References

[1] M. Kadic, G.W. Milton, M. van Hecke, et al., "3D metamaterials", Nat Rev Phys, 1, 198-210, 2019.

[2] A. Valipour, M.H. Kargozarfard, M. Rakhshi, A. Yaghootian, H.M. Sedighi, "Metamaterials and their applications: An overview", Proceedings of the Institution of Mechanical Engineers, Part L: Journal of Materials: Design and Applications, 236(11), 2171-2210, 2022.

[3] Z. Jia, F. Liu, X. Jiang, L. Wang, "Engineering lattice metamaterials for extreme property, programmability, and multifunctionality", J. Appl. Phys., 127 (15), 150901, 2020.

[4] Y. Liu, X. Zhang, "Metamaterials: a new frontier of science and technology", Chem. Soc. Rev., 40 (5), 2494-2507, 2011.

[5] A. Poddubny, I. Iorsh, P. Belov, et al., "Hyperbolic metamaterials". Nature Photon, 7, 948-957, 2013.

[6] J.U. Surjadi, L. Gao, H. Du, X. Li, X. Xiong, N.X. Fang, Y. Lu, "Mechanical Metamaterials and Their Engineering Applications", Adv. Eng. Mater., 21, 1800864, 2019.

[7] K.J. Bathe, Finite Element Procedures, Prentice Hall, USA, 1996, 2nd edition [K.J. Bathe], Watertown, MA, USA, 2014; also published by Higher Education Press, China, 2016.

[8] B. Szabó, I. Babuška, "Finite Element Analysis: Method, Verification and Validation", 2nd Edition, Willey, 2021.

[9] J.E. Akin, "Finite Elements for Analysis and Design: Computational Mathematics and Applications Series", Elsevier, 2014.