Graph grammar model for h-adaptation for meshes with quadrilateral, pentagon, and hexagon elements

Anna Paszyńska[1][0000−0002−0716−0619], Paweł Maczuga[2][0000−0002−5111−6981], Mateusz Dobija[1,3][0000−0003−4557−3534], Albert Oliver-Serra[4][0000−0002−3783−8670], Edyta Paruch[2], and Jakub Radek[2]

[1] Faculty of Physics, Astronomy and Applied Computer Science, Jagiellonian University, ul. prof. Stanisława Łojasiewicza 11, Kraków, Poland
anna.paszynska@uj.edu.pl
mateusz.dobija@doctoral.uj.edu.pl
[2] Faculty of Computer Science, AGH University of Science and Technology, Al. Mickiewicza 30, Kraków, Poland
pawel.maczuga@agh.edu.pl
[3] Doctoral School of Exact and Natural Sciences, Jagiellonian University
[4] University of Las Palmas de Gran Canaria (ULPGC), Gran Canaria, Spain
albert.oliver@ulpgc.es

Abstract. The paper introduces a hypergraph grammar for modeling mesh adaptation in 2D meshes with quadrilateral, pentagon, and hexagonal elements. In this approach, the finite element mesh is represented as a hypergraph, with all mesh transformations defined by hypergraph grammar rules. These rules enable the execution of the h-adaptation of elements, namely the mesh refinement.

Keywords: Finite element method · mesh adaptation · graph grammar.

1 Introduction

This paper explores the modeling of the h-adaptive Finite Element Method (h-FEM) with quadrilateral, pentagon, and hexagon elements using a hypergraph grammar. The h-FEM [1],[2] involves solving an engineering problem by constructing a series of approximation spaces that progressively increase the solution's accuracy. The finite element mesh consists of finite elements along with shape functions corresponding to the nodes, edges, and interiors of these elements. The approximation of the solution is a linear combination of these shape functions. To enhance solution accuracy, some finite elements may be subdivided into smaller elements. There were several attempts to use a grammar approach to model mesh generation and mesh adaptation. The initial study in this field was by Flasiński and Schaefer [3] in 1996, who used graph grammar rules to define transformations for two-dimensional triangular adaptive meshes. This graph grammar approach allows only for modeling uniform refinement of the mesh. To

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_16

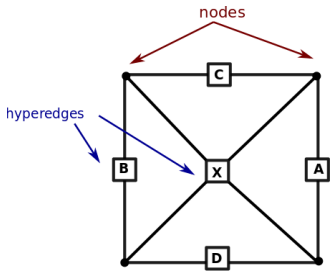

Fig. 1. An exemplary hypergraph with denoted nodes and hyperedges.

enforce used in non-uniform adaptation the "1-irregularity rule" - a finite element can only be subdivided once, without breaking its larger neighboring elements - the context graph grammar should be used. This rule prevents the unbroken edges of elements from being adjacent to more than two other finite elements. In 1993, Grabska ([4], [5]) introduced composition graph grammar (CP-graph grammar) as a new mathematical framework for modeling various design processes. CP-graph grammars describe transformations of the graph representation of domain objects and are effective for various problems due to their contextual nature. CP-graph grammars have been used to model the adaptive finite element method in two dimensions for grids with triangular, rectangular, and mixed triangular and rectangular ([8],[9]) elements. In 1987, Habel and Kreowski ([6], [7]) introduced hypergraph grammars as a tool for generating hypergraph languages. The first application of hypergraph grammars for mesh generation and adaptation was presented by Ślusarczyk and Paszyńska ([10]) for rectangular non-uniform meshes in 2013. Hypergraph grammars were also used to model non-uniform meshes with triangular elements (with the longest edge algorithm used for mesh adaptation) in [11]. This paper extends previous hypergraph grammar models to adapt two-dimensional non-uniform meshes with quadrilateral, pentagonal, and hexagonal elements.

2 Hypergraphs and hypergraphs grammar

In this paper, we will represent finite element mesh, consisting of quadrilateral, pentagon, and hexagonal elements by hypergraphs and the finite element refinement process by hypergraph grammar productions. A hypergraph is a graph consisting of nodes and the so-called hyperedges. A hyperedge can join two or more nodes. The nodes and hyperedges of a graph can represent modeled objects. Thus, additional information, such as labels representing the type of object and attributes representing the objects properties, is assigned to nodes and hyperedges. An exemplary graph representing the one-element quadrilateral mesh is presented in figure 1. Changes in the structure of a graph can be obtained by graph grammar productions. The production is defined by a so-called left-hand-side graph and the right-hand-side graph. The left-hand-side graph should

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_16

have external nodes specified. The right-hand-side graph should have the same number of external nodes specified. These nodes will correspond to target nodes of replaced hyperedges during the application of a production. The application of the production to a hypergraph H can be seen as a three-step process:

- Find a subgraph l of graph H isomorphic to the left-hand-side-graph of the production
- replace the subgraph l in graph H with the right-hand-side graph
- replacing the external nodes of the removed subgraph with the corresponding external nodes of the inserted graph. Figure 3 presents the graph of figure 1 after applying the production of figure 2. Additionally, the so-called applicability predicate of the production can be defined, describing the non-structural conditions, like, for example, constraints on attribute values, which should be satisfied in the graph to apply the production to that graph.

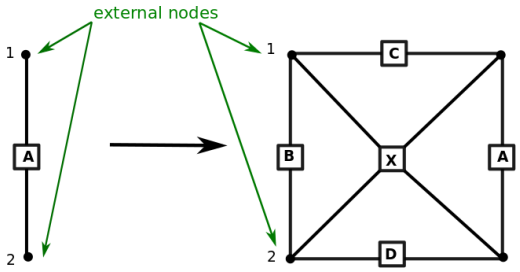

Fig. 2. Exemplary hypergraph production with denoted external nodes numbered by 1 and 2.

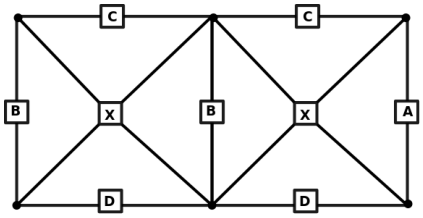

Fig. 3. Graph from figure 1 after applying the production from figure 2

3 Graph grammar model for h-adaptation for meshes with quadrilateral, pentagon and hexagon elements

The mesh generation process usually starts with a simple mesh that is iteratively adapted to a mesh on which a solution can be obtained with the desired

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_16

precision (breaking, for example, a quadrilateral element into two smaller quadrilateral elements). In the process of adaptation, the so-called "hanging nodes" can occur. The hanging nodes in two-dimensional space represent edges that have two smaller elements on one side and one larger one on the other side (see Fig. 4. In the finite element mesh we distinguish boundary edges (on the border of the domain) and shared edges. The hanging nodes can be created only on shared edges. In order to enforce during performing mesh refinements the "1-irregularity rule"- an element can be broken only once without breaking its neighbors - the process of refinement can be split into three steps. First step, the so-called virtual adaptation, which means marking an element for refinement (according to the refinement criterion, for example, big solution error on this element). The second step is propagation of virtual adaptation: if an element marking for breaking is neighboring to bigger, not refined element, the bigger element is also marked for breaking. Finally, in the last step, all the elements marked for breaking are divided into smaller elements.

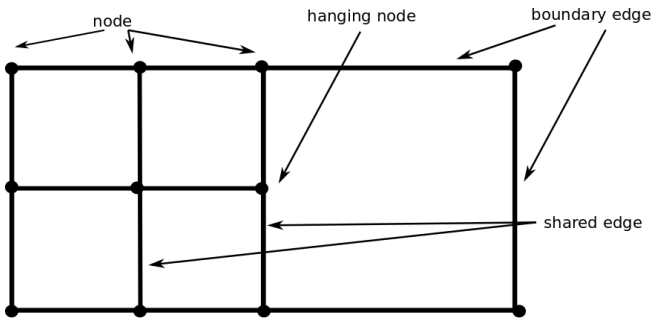

Fig. 4. An exemplary mesh with hanging node

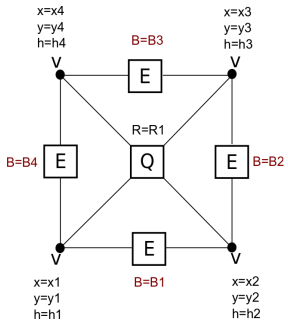

Fig. 5. Representation of quadrilateral element with the usage of hypergraph

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_16

In the paper, three types of elements are used: quadrilateral, pentagon, and hexagon. The elements are represented in the form of a hypergraph - see Figs. 5, 14.

Let us define the hypergraph nodes with the label **v** as the nodes that represent the polygon nodes. We define it's attributes as: $\mathbf{x}, \mathbf{y}$ - nodes coordinates, **h** - boolean value defining if node v is a hanging node (value 1 represents case of a hanging node, 0 represents case of a normal node). Let us define hyperedges of the hypergraph with label **E**, as representation of edges of the polygon. We define it's attribute as: **B** - boolean value defining if the edge is a boundary edge ($B = 1$ for a boundary edge, 0 for a shared one). Let us define the hyperedge with label **Q**, as a representation of the interiors of the quadrilateral element, the hyperedge with label **P**, as a representation of the interiors of the pentagon element, and the hyperedge with label **S**, as a representation of the interiors of the hexagonal element. We define the attribute of hyperedges with labels Q,P,S that represent the interiors of elements in the following way: **R** - boolean value defining if an element needs to be broken further ($B = 1$ for an element that needs further breaking, 0 for not needing this one). As the hypergraph representation of the element is defined, we introduce the productions describing process of breaking of element of the mesh into smaller ones. For each production, we define an applicability predicate, indicating when the production can be applied.

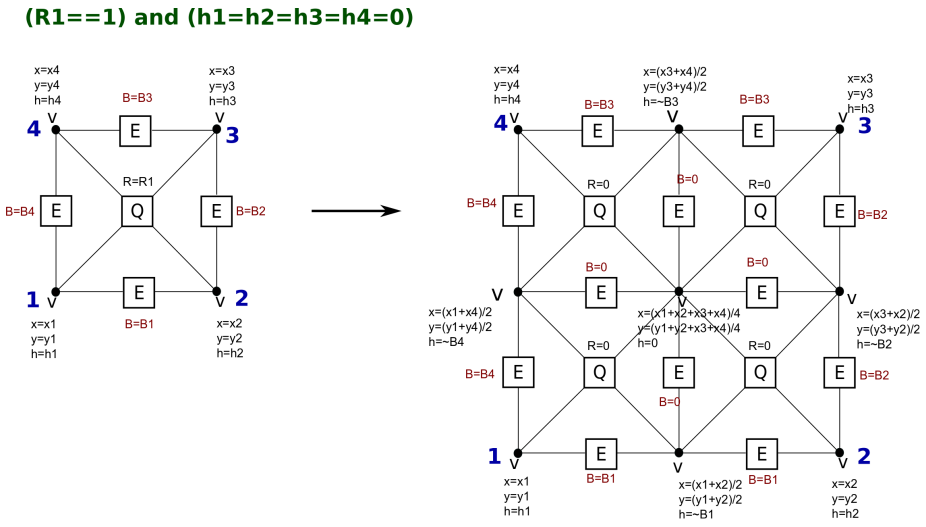

Fig. 6. Production **P1** - quadrilateral element breaking.

Production **P1** (see Fig. 6) considers the breaking of the quadrilateral element (marked for breaking, without hanging nodes) into four smaller ones. For production $P1$, the applicability predicate is defined as: ($R1 = 1$) and

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_16

$(h1 = h2 = h3 = h4 = 0)$ The predicate means, that we can apply production only if the quadrilateral element is marked to be broken $(R1 = 1)$, and all of the nodes are normal nodes (not hanging nodes, $h1 = h2 = h3 = h4 = 0$). The attributes x, y for newly created graph nodes with the label v are calculated as the arithmetic mean of the coordinates of the proper adjacent nodes with label v (see Fig. 6). If the broken edge was on the domain boundary, the attribute h assigned to the newly created node with the label v is set to 0. For the newly created node with the label v on the shared edge, the attribute h is set to 1, which indicates hanging node ($h =\sim B$). The newly created hyperedges with label E, corresponding to small edges of elements created by breaking edges of element, have the value of attribute B the same,as the corresponding big edge, before breaking of the element ($B = B$ of corresponding big edge). The newly created hyperedges with label E, corresponding to shared edges between newly created elements have attribute B set to 0 (shared edge).

Production **P2** (see Fig. 7) considers the breaking of the quadrilateral element (marked for breaking, while one of the edges is placed with the hanging node) into four smaller ones. For the production **P2**, the predicate is defined in the following form: $(R1 = 1)$ and $(h1 = h2 = h3 = h4 = 0)$ and $(h5 = 1)$. In this case, we can apply production if the element is marked to be broken $(R1 = 1)$, and the nodes $1 - 4$ are normal nodes (not hanging nodes, $h1 = h2 = h3 = h4 = 0$), and node 5 is a hanging node $(h5 = 1)$. What is important is that node 5 after application of the production is no longer a hanging node $(h5 = 0)$.

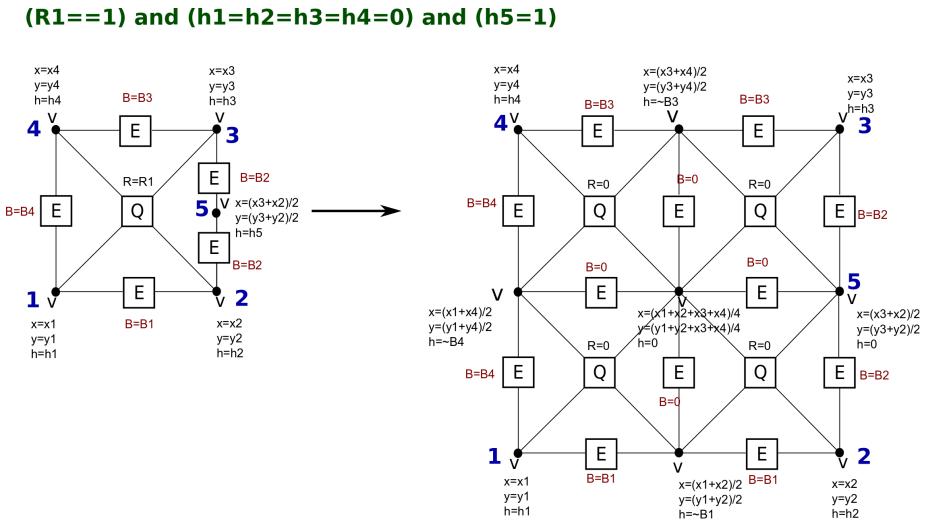

Fig. 7. Production **P2** - breaking of the quadrilateral element with one hanging node.

Production **P3** (see Fig. 8) considers the breaking of the quadrilateral element (marked for breaking, with hanging nodes placed on 2 adjacent edges) into 4

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_16

smaller ones. For production **P3**, the applicability predicate is defined in such a way that we can apply production only if the element is marked to be broken ($R1 = 1$), and nodes $1-4$ are normal nodes (not hanging nodes, $h1 = h2 = h3 = h4 = 0$), and nodes 5 and 6 are hanging nodes ($h5 = h6 = 1$). What is important nodes 5 and 6 after application of the production are no longer hanging nodes ($h5 = h6 = 0$).

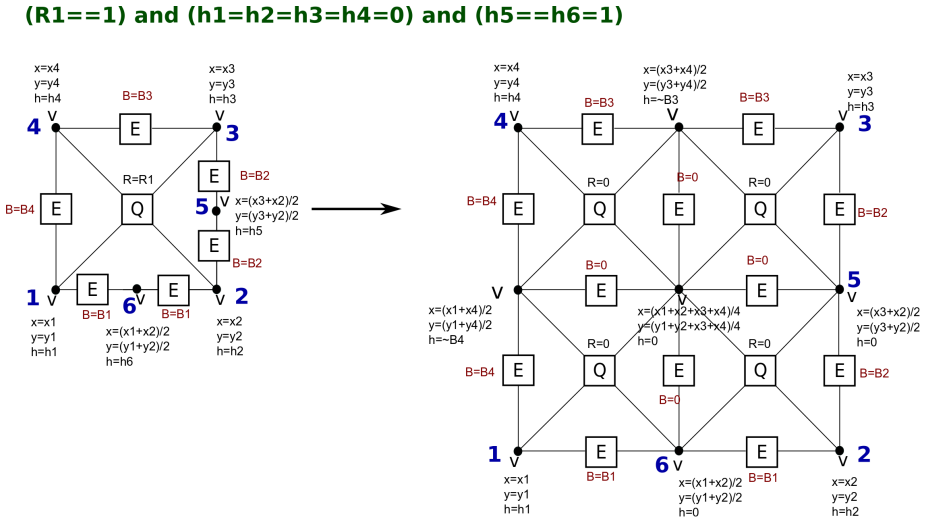

Fig. 8. Production **P3** - breaking of the quadrilateral element with hanging nodes on the neighboring edges.

Production **P4** (see Fig. 9) considers the breaking of the quadrilateral element (marked for breaking, with hanging nodes placed on two opposite edges) into four smaller ones. For the production **P3**, the applicability predicate is defined in such a way that we can apply the production only if the element is marked to be broken ($R1 = 1$), and nodes $1-4$ are normal nodes (not hanging nodes, $h1 = h2 = h3 = h4 = 0$), and nodes $5, 6$ are hanging nodes ($h5 = h6 = 1$). The value of attribute h for nodes 5,6 is set as in the production P3.

Production **P5** (see Fig. 10) considers the breaking of the quadrilateral element (marked for breaking, with a hanging node placed on three adjacent edges) into four smaller ones. For the production **P5**, the applicability predicate is defined in such a way that we can apply production only if the element is marked to be broken ($R1 = 1$), nodes $1-4$ are normal nodes (not hanging nodes, $h1 = h2 = h3 = h4 = 0$), and nodes $5, 6, 7$ are hanging nodes ($h5 = h6 = h7 = 1$). What is important, nodes $5, 6, 7$ after the application of the production are no longer hanging nodes ($h5 = h6 = h7 = 0$).

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_16

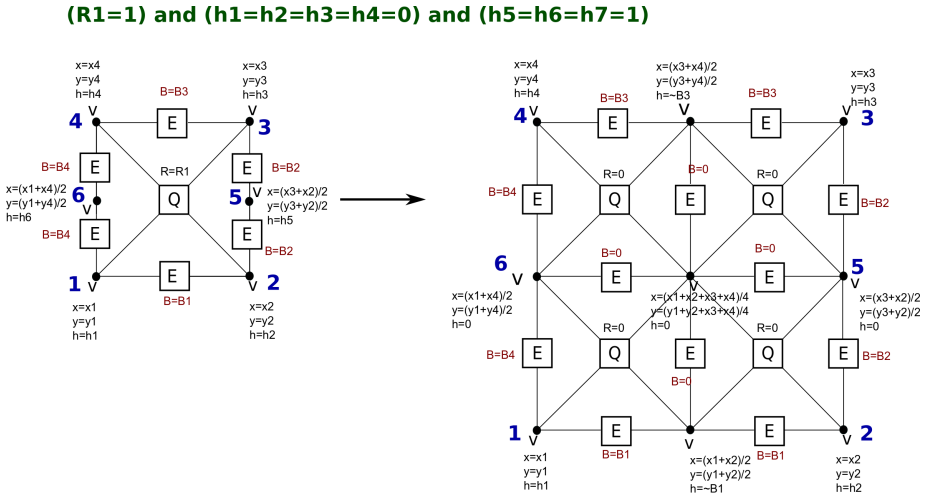

Fig. 9. Production **P4** - breaking of the quadrilateral element with hanging nodes on the opposite edges.

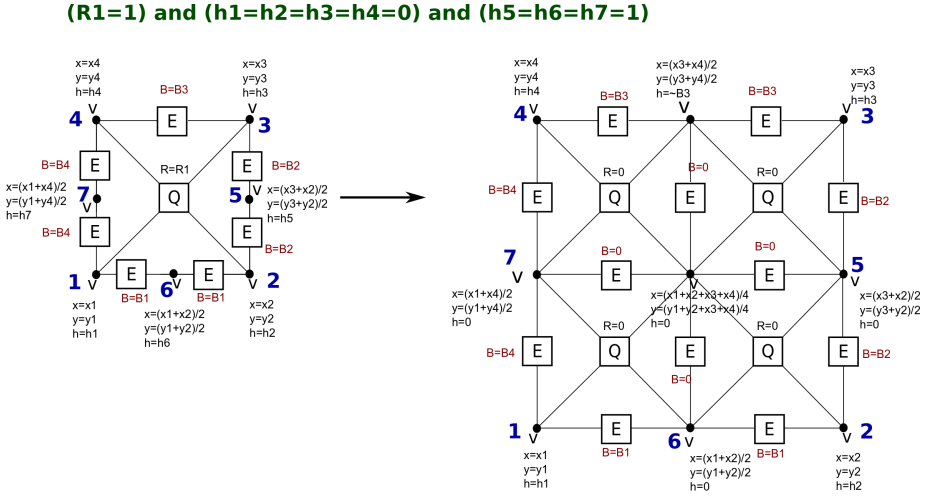

Fig. 10. Production **P5** - breaking of the quadrilateral element with three hanging nodes.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_16

Production **P6** (see Fig. 11) considers the breaking of the quadrilateral element (marked for breaking, with hanging nodes placed on all four edges) into four smaller ones. For the production $P6$, the predicate is defined in such a way that it can be applied only if the element is marked to be broken ($R1 = 1$), and nodes $1 - 4$ are normal nodes, and nodes $5, 6, 7, 8$ are hanging nodes. What is important, nodes $5, 6, 7, 8$ after the application of the production are no longer hanging nodes ($h = 0$).

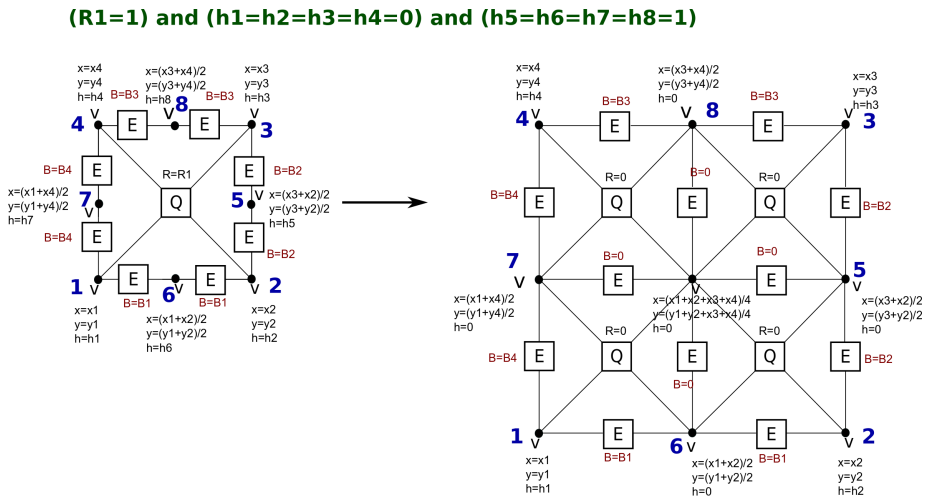

Fig. 11. Production **P6** - breaking of the quadrilateral element with four hanging nodes.

Production **P7** models marking the quadrilateral element for breaking (virtual adaptation). We can apply the production only if the applicability predicate is fulfilled - if the element is not yet marked to be broken ($R1 = 0$) and the refinement criterion is fulfilled (RFC). It enables to perform further breakages on the elements.

We will enforce the "1-irregularity rule": "The element can be broken only once without breaking adjacent big element". This means that, after the application of production $P7$, there can be additional virtual adaptations needed for particular elements. Production **P8** (see Fig. 13) is called "virtual adaptation propagation". If a small element is marked to be broken, it also marks the adjacent big element as needing a break.

The applicability predicate for production $P8$ is defined as ($R1 = 0$) and ($R2 = 1$) and ($h5 = 1$). In this case, we can apply production only if the big element is not marked for breaking ($R1 = 0$), small neighboring element is marked for breaking ($R2 = 1$), and there is also a hanging node between them ($h5 = 1$).

The generated mesh can also consist of pentagon and hexagonal elements (Figure 14). presents hypergraphs representing pentagons and hexagons. The

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_16

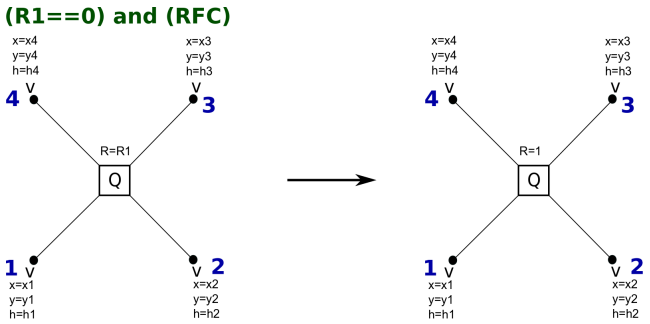

Fig. 12. Production $P7$ - marking quadrilateral element for breaking (virtual adaptation

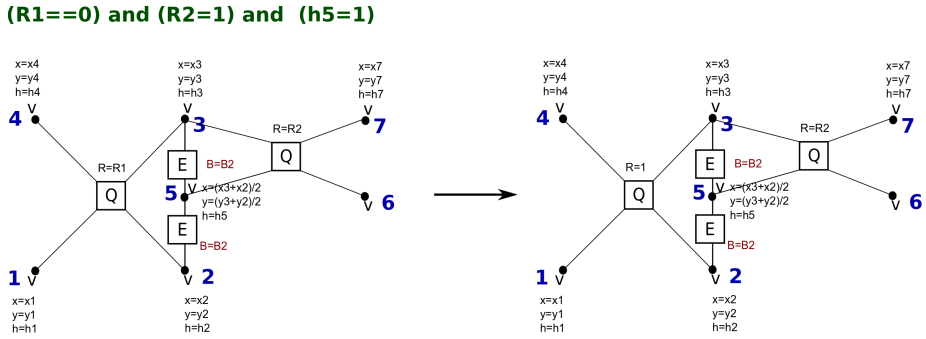

Fig. 13. Propagation of virtual adaptation for quadrilateral elements - production $P8$.

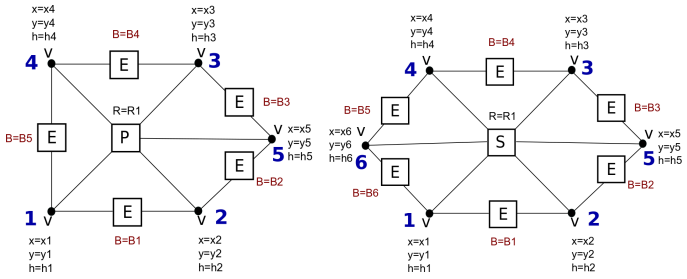

Fig. 14. Left panel: Hypergraph representation of pentagonal element. Right panel: Hypergraph representation of hexagonal element.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_16

hyperedges that represent the interiors of pentagon elements have label P, and the hyperedges that represent the interiors of hexagonal elements have label S. The other attributes of the hypergraph are defined as for the quadrilateral elements.

As for the quadrilateral case, we define also productions for pentagon and hexagonal elements that define possible refinements of the mesh. Each pentagon element marked to be broken can be broken into five quadrilateral elements by breaking each edge of the pentagon at the midpoint, adding new edges between these midpoints and the centroid of the element. Similarly, each hexagonal element marked to be broken can be broken into six quadrilateral elements. Production **P9** breaks the pentagon element (marked to be broken, without hanging nodes) into five small quadrilaterals. The values of attribute h for newly created nodes are defined similarly as in production $P1$. The attributes x, y for newly created graph nodes with the label v are calculated as the arithmetic mean of the coordinates of the proper adjacent nodes with the label v. If the broken edge was on the domain boundary, the attribute h assigned to the newly created node with the label v is set to 0. For the newly created node with the label v on the shared edge, the attribute h is set to 1, which indicates the hanging node ($h =\sim B$).

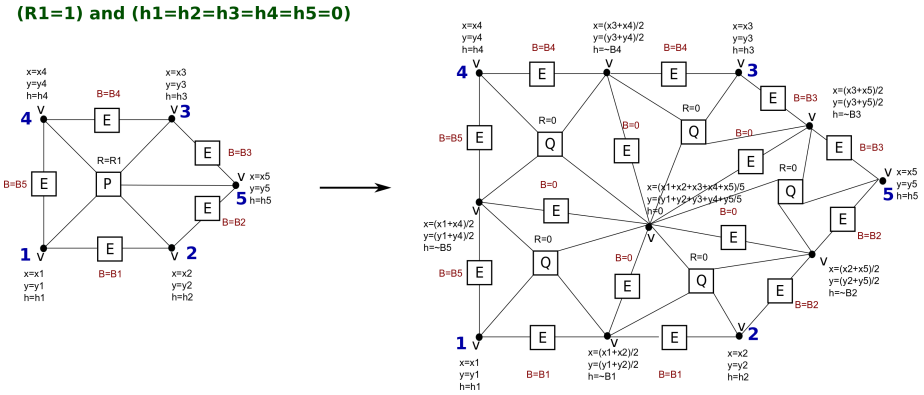

Fig. 15. Production $P9$ - Pentagon element breaking (without hanging nodes).

Similarly, the productions **P10** – **P15** for breaking the pentagon element marked to be broken with one or more hanging nodes are defined. Figure 16 presents the production **P11** for breaking the pentagon element marked to be broken with two hanging nodes on the neighboring edges.

Production **P16** allows for virtual adaptation of pentagon elements (similar to production $P7$ for quadrilateral elements).

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_16

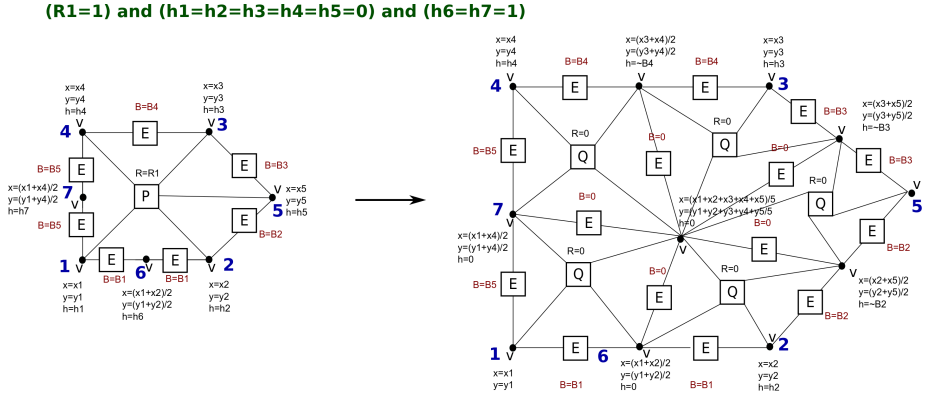

Fig. 16. Production $P11$ - Pentagon element breaking (with two broken edges, with two hanging nodes on neighboring edges).

Figure 17 presents the production **P17** for the propagation of the adaptation, when we have one big pentagon element, not marked for refinement, and a neighboring small quadrilateral element marked for refinement.

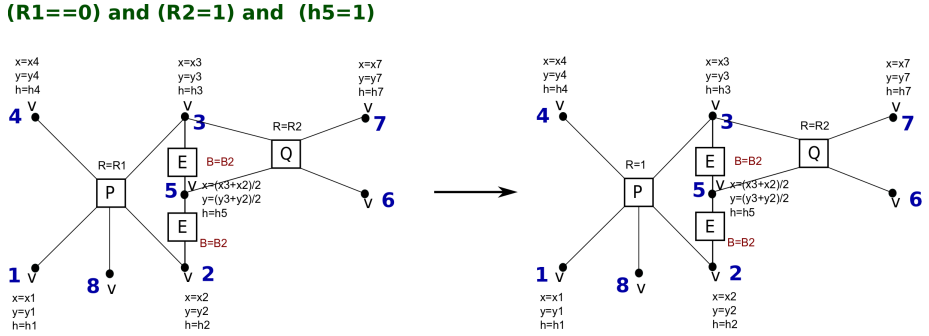

Fig. 17. Production $P17$ - virtual adaptation propagation.

There are also defined productions for breaking hexagonal elements. Fig. 18 presents the production **P18** for breaking the hexagonal element, marked to be refined without hanging nodes. In a similar way as for the case of quadrilateral elements, the production for breaking hexagonal elements marked to be broken, with one, two, etc. hanging nodes, as well as production for virtual adaptation and virtual propagation of adaptation are defined.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_16

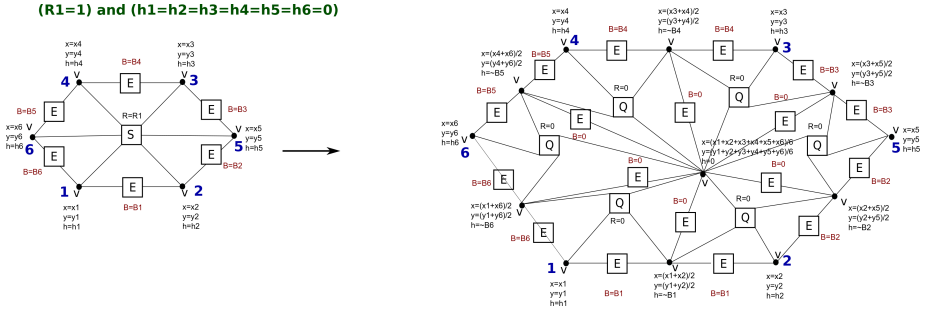

Fig. 18. Production $P18$ for breaking hexagonal element, marked to be refined with no hanging nodes.

4 An exemplary hypergraph grammar derivation

In Figure 19 an exemplary mesh adaptation is presented. The red dot denotes the element marked for adaptation, according to the refinement criterion. Figure 20 presents the derivation process, step by step, for the mesh adaptation from Figure 19. For clarity of the figure, the labels of hyperedges representing edges of element are omitted.

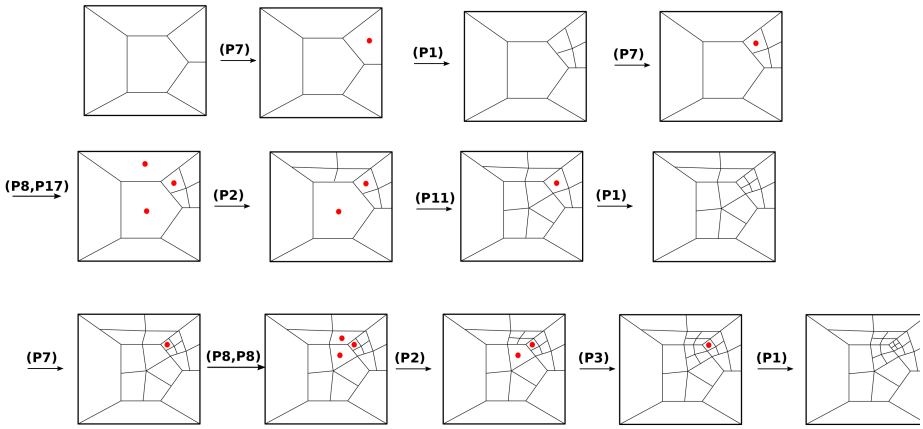

Fig. 19. h-adaptation of the mesh, step by step. The red dot denotes an element marked for adaptation.

Hyperedges drawn with a continuous line represent boundary edges and hyperedges drawn with a dashed line represent shared edges. Nodes of the hypergraph denoted by blue circles represent hanging nodes and nodes denoted by blue circles with border represent non-hanging nodes. Hyperedges representing

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_16

an interior of an element are denoted by pink squares. The pink square with border denotes interior of element marked to be broken. The pink square without border denotes the interior of the element that is not marked to be broken.

5 Conclusions

This paper introduced a hypergraph grammar designed to model the h-adaptation of 2D meshes with quadrilateral, pentagon, and hexagonal elements. The finite element mesh is represented through hypergraphs, and mesh transformations are captured via hypergraph grammar productions. This grammar enables marking of the element for the adaptation, virtual adaptation, and propagation of virtual adaptation. Future work will involve integrating the graph grammar with the solver algorithm, extending the presented approach to other element types and 3D meshes, and solving numerical problems with error estimates using the proposed adaptation technique.

Acknowledgements

References

1. Hughes, TJR.: Linear static and dynamic finite element analysis. Dover Publications, (2000)
2. Demkowicz, L.: Computing with Hp-Adaptive Finite Elements, Vol. 1: One and Two Dimensional Elliptic and Maxwell Problems. Chapman & Hall / CRC Press (2006)
3. Flasinski, M., Schaefer, R.: Quasi context sensitive graph grammars as a formal model of FE mesh generation. Computer-Assisted Mechanics and Engineering Science **3**, 191–203 (1996)
4. Grabska, E.: Theoretical Concepts of Graphical Modeling. Part One: Realization of CP-Graphs, Machine Graphics and Vision **2(1)**, 3–38 (1993)
5. Grabska, E.: Theoretical Concepts of Graphical Modeling. Part Two: CP-Graph. Grammars and Languages, Machine Graphics and Vision **2(2)**, 149–178 (1993)
6. Habel, A, Kreowski, HJ.: May We Introduce to You: Hyperedge Replacement. Lecture Notes in Computer Science **291** 5–26 (1987)
7. Habel, A., Kreowski, HJ.: Some Structural Aspects of Hypergraph Languages Generated by Hyperedge Replacement. Lecture Notes in Computer Science **247** 207-–219 (1987)
8. Paszynska, A., Paszynski, M.: Graph transformations for modeling parallel hp-adaptive finite element method. Lecture Notes in Computer Science **4967** Berlin/Heidelberg 1313—1322 (2008)
9. Paszynska, A., Paszynski, M., Grabska, E.: Graph transformations for modeling hp-adaptive finite element method with mixed triangular and rectangular elements. Lecture Notes in Computer Science **5545** Berlin/Heidelberg 875—884 (2009)
10. Slusarczyk, G., Paszynska, A.: Hypergraph grammars in hp-adaptive finite element method. Procedia Computer Science **18** 1545—1554 (2013)
11. Podsiadło, K., Oliver Serra, A, Paszynska, A., Montenegro, R., Henriksen, I., Paszyński, M., Pingali, K.: Parallel graph-grammar-based algorithm for the longest-edge refinement of triangular meshes and the pollution simulations in Lesser Poland area. Engineering with Computers 1–24 (2021)

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_16

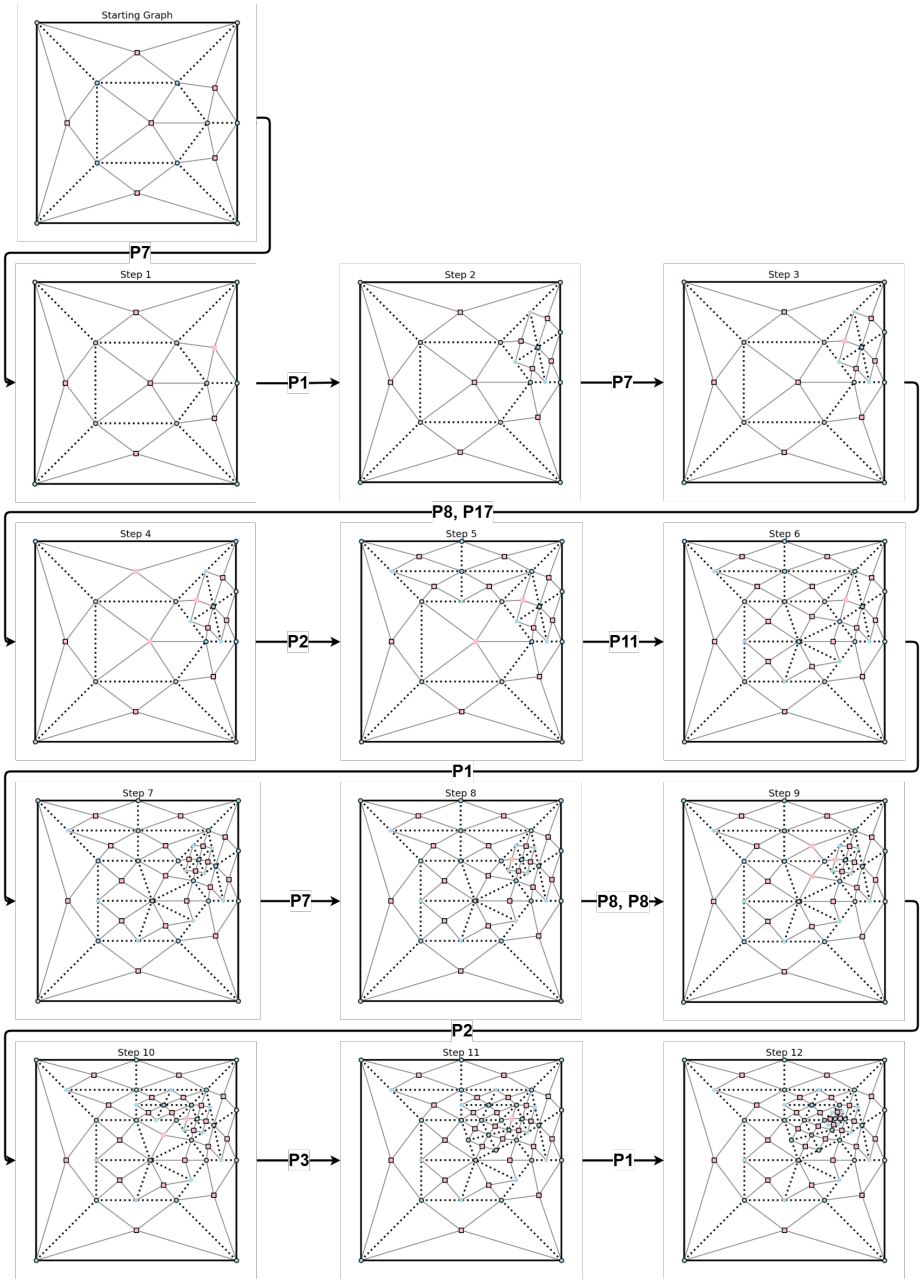

Fig. 20. Hypergraph grammar derivation for the mesh from the figure 19 of the mesh, step by step.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_16