XLVI CILAMCE
VITÓRIA - ES - BRAZIL - 2025
XLVI IBERO-LATIN AMERICAN CONGRESS ON COMPUTIONAL METHODS IN ENGENEERING

Software development for adaptive processes: AMVIEW and FESTA experiences

Joaquim B. Cavalcante-Neto[1], Luiz F. Martha[2]

[1]*Dept. Computing, Federal University of Ceara*
Campus do Pici – Bloco 910, 60455-760, Fortaleza/Ceará, Brazil
joaquimb@dc.ufc.br
[2]*Dept. of Civil and Environmental Engineering, Pontifical Catholic University of Rio de Janeiro*
Rua Marques de São Vicente, 255, 22451-900, Rio de Janeiro/Rio de Janeiro, Brazil
lfm@tecgraf.puc-rio.br

Abstract. Adaptive processes are very important in computational mechanics. Basically, the idea is the system be able to adapt itself, without the help from the user, converging to a better solution. We developed two software programs to address this problem: initially the Adaptive Mesh View (AMVIEW), which was able to design and implement an adaptive solution and then this software evolved to the Finite Element System Technology in Adaptivity (FESTA), which is a more complete environment for adaptive processes. These works introduce a methodology for self-adaptive numerical procedures, which relies on the various components of an integrated, object-oriented, computational environment involving pre-, analysis, and post-processing modules. A basic platform for numerical experiments and further development is provided, which allows implementation of new elements/error estimators and sensitivity analysis. A general implementation of the Superconvergent Patch Recovery (SPR) and the Recovery by Equilibrium in Patches (REP) is presented. Both SPR and REP are compared and used for error estimation and for guiding the adaptive remeshing process. Moreover, the SPR is extended for calculating sensitivity quantities of first and higher orders. The mesh (re-)generation process is accomplished by means of methods combining quadtree and Delaunay triangulation techniques. Surface mesh generation in arbitrary domains is performed automatically (i.e. with no user intervention) during the self-adaptive analysis using either quadrilateral or triangular elements. The effectiveness and versatility of both software are demonstrated by representative numerical examples illustrating the interconnections among finite element analysis, recovery procedures, error estimation/adaptivity and automatic mesh generation.

Keywords: Adaptive Software, Error Estimation, Mesh Refinement, Finite Elements.

1 Introduction

This work presents an integrated (object-oriented) computational environment for self-adaptive analyses of generic two-dimensional (2D) problems. This environment includes analysis procedures to ensure a given level of accuracy according to certain criteria and, also, the procedures to generate and modify the finite element discretization. This system is called FESTA (Finite Element System Technology in Adaptivity) and it is an evolution of the AMVIEW (Adaptive Mesh View), which was initially designed to implement adaptive solutions and evolved to a more complex environment for these simulations. It involves five main components (Fig. 1):

1) A graphical preprocessor, for defining the geometry of the problem, the initial finite element mesh (together with boundary conditions), and the main parameters used in a self-adaptive analysis. Here the geometrical model is dissociated from the finite element model;

2) A finite element module for solving the current boundary value problem. The code has been developed so that it is highly modular, expandable, and user-friendly. Thus, it can be properly maintained and continued. Moreover, other users/developers should be able to modify the programming system to their specific applications;

3) An error estimation and sensitivity module. Discretization errors are estimated according to available recovery procedures, e.g. Zienkiewicz and Zhu (ZZ), superconvergent patch recovery (SPR) and recovery by equilibrium in patches (REP). Sensitivities of various orders (1st., 2nd. or higher) are calculated by means of a procedure analogous to the SPR. The user chooses the desired error estimator and sensitivity order;

4) A mesh (re-)generation (rather than mesh enrichment) procedure, based on the combination of quadtree and Delaunay triangulation techniques. According to the magnitude of the error, calculated in the previous module, a new finite element mesh is automatically generated (i.e. with no user intervention);

5) Finally, a postprocessor module, where all the analysis results can be visualized.

The remainder of this paper is organized as follows. A motivation to the work and a "brief" literature review are provided in Section 2. Afterwards, Section 3 presents some theoretical background on self-adaptive simulations and an overview of the computational aspects of FESTA, an evolution from AMVIEW with improvements. A discussion about the automatic mesh generation techniques used in this work is given in Section 4. In order to assess the effectiveness of the proposed computational system, representative numerical examples are given in Section 5. Finally, in Section 6, conclusions are inferred and directions for future research are discussed.

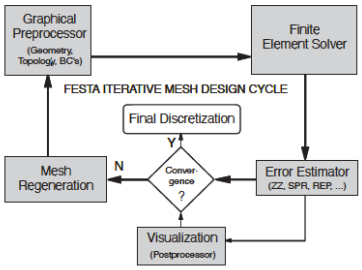


Figure 1. Simplified diagram of the interactive meshing

2 Motivation and related work

Normal practice to solve engineering problems by the Finite Element Method (FEM) or the Boundary Element Method (BEM) involves increasing the number of discretization points in the computational domain and resolving the resulting system of equations to examine the relative change in the numerical solution. In general, this procedure is time consuming, it depends on the experience of the analyst, and it can be misleading if the solution has not entered an asymptotic range. Ideally, with a robust and reliable self-adaptive scheme, it is possible to specify an initial discrete model which is sufficient to describe the geometry/topology of the domain and the boundary conditions (BCs), and to define a desired error tolerance, according to an appropriate criterion. Then, the system would, automatically and without any user intervention, refine the model until the error measure falls below the prescribed tolerance. This is the main goal which motivated the development of AMVIEW and FESTA. It increases the reliability of the analysis procedure since it does not depend on the experience of the analyst.

The need for developing better pre-processing techniques for the FEM, for performing automated analysis, and obtaining self-adaptive solutions have driven the development of automatic mesh generation algorithms which can discretize an arbitrary geometry into a consistent finite element mesh without any user intervention. Several algorithms for 2D geometries were developed (e.g. Baehmann et al. [1; Zhu et al. [2]; Potyondy et al. [3]; Borouchaki and Frey [4]), and approaches for 3D geometries have also appeared (e.g. Cass et al. [5]; Beall et al. [6]; Lo [7]). This work focus on automatic 2D mesh generation in connection with adaptive solutions. Also, the present work was the basis for an adaptive system for 3D problems, further developed by the same authors.

The first papers on adaptive finite elements appeared in the early seventies. Since then, an explosive number of papers on the subject have been published. Babuska and Rheinboldt [8] presented a pioneering paper about error estimates by evaluating the residuals of the approximate solution and using them to obtain local, more accurate answers. They developed the mathematical basis of self-adaptive techniques. With the concept of a posteriori error estimates, a self-adaptive strategy for the FEM can be developed such that only certain elements should be refined. Zienkiewicz et al. [9] presented a hierarchical approach for self-adaptive methods. Computer graphics techniques started to be used as standard tools by mesh generation programs. Shephard [10] published a paper where geometric modeling and automatic mesh generation techniques were used in conjunction with self-adaptive methods. Zienkiewicz and Zhu [11] introduced an error estimator based on obtaining improved values of gradients

(stresses) using some available recovery processes. Easy to be implemented in any finite element code, this technique, based on averaging and on the called L_2 projection, has been used to recover the gradients, and reasonable estimators were achieved. This technique was corrected/improved by the same authors, leading to the called Superconvergent Patch Recovery: SPR (Zienkiewicz and Zhu [12] [13] [14]). This method is a stress-smoothing procedure over local patches of elements and is based on a discrete least-squares fit of a higher-order local polynomial stress field to the stresses at the superconvergent sampling points obtained from the finite element calculation. Essentially these improved techniques incorporate equilibrium and boundary conditions on the recovery process. An exhaustive study by Babuskka et al. [15] showed, through numerous examples, the excellent performance and superiority of the SPR over residual-type approaches. Boroomand and Zienkiewicz [16] presented a new super-convergent method satisfying the equilibrium condition in a weak form, which does not require any knowledge of superconvergent points. The new recovery technique was called Recovery by Equilibrium in Patches: REP. As indicated above, the field of adaptivity is broad and advanced significantly over the years. For instance, Paulino et al. [17] have proposed a new class of error estimates based on the concept of nodal sensitivities, which can be used in conjunction with general purpose computational methods such as FEM, BEM or FDM. Mahomed and Kekana [18] have presented an adaptive procedure based on strain energy equalization. Moreover, a summary of recent advances in adaptive computational mechanics can be found in the book edited by Ladeveze and Oden [19]. Integration of error estimation and adaptivity is the focus of this work.

3 Theoretical and computational aspects

When a numerical method is used, error is introduced in the process. These errors are defined as the difference between the solution and its numerical approximation. Self-adaptive methods are schemes which automatically adjust themselves to improve the solution to a specified accuracy. The two components are error estimation and adaptive strategy. There are two types of error estimates: a priori and a posteriori. Although a priori estimates are accurate for particular problems, they usually do not provide information about the actual error. A posteriori estimates use information obtained during the solution, in addition to some a priori assumptions. It can provide accurate measures of the error and is adopted here. It includes h, p and r versions, which deals with changes in the size of the elements, the polynomial order or the position of the nodes, respectively.

Regarding error estimation and adaptive refinement, the local error is generally not convenient and can be misleading. Thus, norms are introduced to measure the discretization error which is often quantified based on the energy norm for the displacement error, $\|e\|$, expressed in terms of stresses where σ_{ex} and σ_{fi} are the exact and the finite element stress fields. The idea of error estimators is to substitute σ_{ex}, which is unknown, by σ_{ap}, obtained by means of recovery procedures (e.g. ZZ, SPR, REP). Therefore, the relative error distribution is expressed more easily. For a finite element i, $\|e\|$ can be recalculated based on recovered stress field σ_{ap}, where is assumed isoparametric elements and can be evaluated in the whole domain or part of it. The contribution of all elements can be calculated and the relative percentage error in the energy norm η_{ex} for the whole domain or part of it can be obtained. The h-version strategy can then be performed, based on the error which defines how the discretization model will be refined or coarsened. A simple criterion to achieve a solution error with an acceptable level can be stated as $\eta_{es} < \eta_{max}$ where η_{max} is the maximum permissible error and η_{es} is energy norm obtained based on the finite element solution. A sound criterion for an "optimal mesh" consists of requiring that the energy norm error be equidistributed among elements because it leads to meshes with high convergence rates. Thus, for each element i, by defining the ratio ξ_i which is a relation between the estimated energy norm for the element and average energy norm for the mesh, it is obvious that refinement is needed if $\xi_i > 1$. A more efficient procedure, which is adopted here, consists of designing a completely new mesh which satisfies the requirement $\xi_i <= 1$ in the limit of refinement. By assuming a certain rate of convergence, the value of the error ratio ξ_i can be used to decide the new size of the element h, based on the initial size of the element h_i and polynomial order of the approximation.

In relation to computation aspects, to obtain an efficient and robust computational system for self-adaptive simulations, a generic treatment for dealing with different finite elements must be considered. Thus, AMVIEW and FESTA were developed in C++ language, using object-oriented programming (OOP). The most important task in OOP is the definition of the class structure. In our case, several classes have been defined, such as FEM, Node, Element, Material, AnModel, Gauss, Shape, LoadElem, Error, and Sen (Fig. 2 shows this class organization). The FEM class represents the numerical discretization of the model. It contains references to objects

of several classes: to a list of objects of Node class (the nodes of the mesh), to a list of objects of Element class (the elements of the mesh), to a list of objects of Material class (the distinct materials used), and a reference to an object of the AnModel class, which is responsible for the specific features of the type of analysis being performed (e.g. plane-stress, plane-strain). The Gauss class holds information about the numerical integration process. The Shape class holds the geometric and field interpolation aspects of an element and is also responsible for the recovery techniques once the terms of the polynomial expansion are specified. The LoadElem class consists of fictitious elements that transfer natural boundary conditions from the elements to the nodes. The Error class is responsible for computing the error distribution and the Sens class computes the sensitivities.

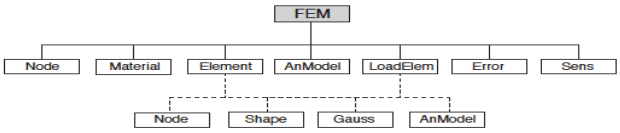
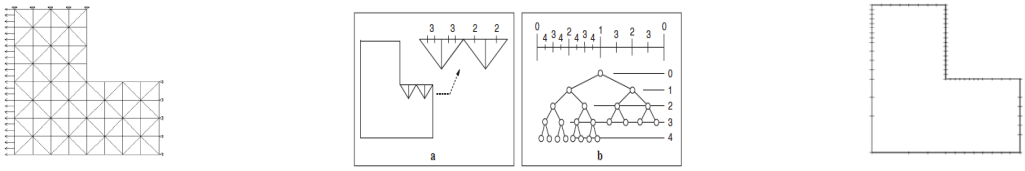


Figure 2. OOP class organization.

4 Mesh generation

The required characteristics of an integrated system for interactive, self-adaptive finite element analysis are robustness, versatility, and computational efficiency. This section summarizes the developed methodology for 2D self-adaptive process aimed on these aspects. The strategy is: 1) Geometry-based mesh generation - The simulation model is defined by its geometry and topology. The simulation attributes are linked to topological entities. A simple topological description is adopted: a list of regions, defined by a list of boundaries, which in turn are defined by a list of curves. The geometrical properties are attached to the curves; 2) Independent boundary and domain recursive decomposition - Boundary curves are discretized "a priori" using a recursive binary decomposition algorithm, which is based on the error estimation analysis. Each region domain is recursively treated using a quadtree decomposition, which is based on the boundary discretization and on the error estimation; 3) Mesh generation combining quadtree and Delaunay techniques - The mesh is generated in two stages. Initially, the interior of the region is meshed using templates based on quadtree decomposition. In a second stage, the areas between the interior mesh and the boundary of the region are meshed using a "boundary contraction technique" based on a Delaunay triangulation property; 4) Automatic analysis attribute redistribution - Because attributes are attached to topological and geometrical entities, they are automatically reapplied to the generated elements, element boundaries, and nodes generated in the self-adaptive process. Similar concepts can be applied in 3D.

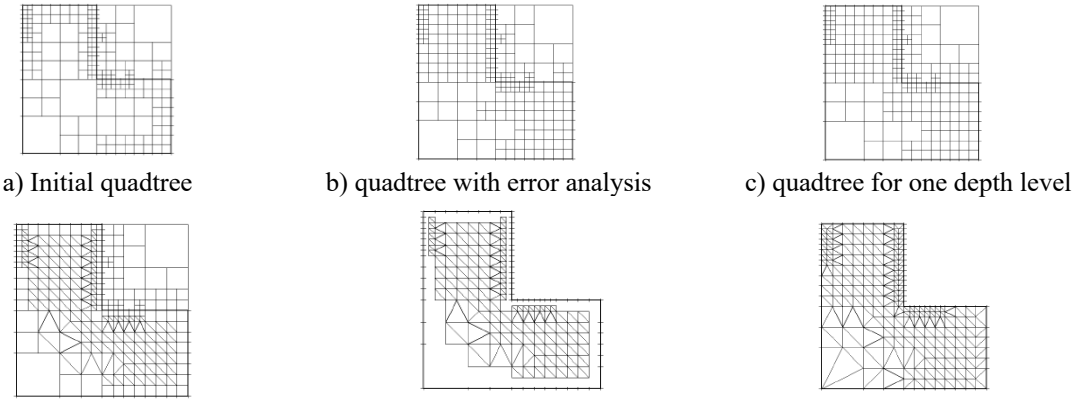
Regarding recursive boundary decomposition, one important aspect is that the boundary refinement is enforced independently of the domain refinement. Each boundary curve is discretized in its own parametric space. Therefore, the strategy is general for all classes of geometric curves. The algorithm used to refine each boundary curve is a one-dimensional version of the one used to refine the domain, which is based on a quadtree technique. Each curve is decomposed using a binary tree technique. The idea consists of recursively subdividing the curve into segments whose sizes are computed based on the characteristic sizes of finite elements adjacent to the curve. These sizes are dictated by the error estimation analysis of the previous step. The recursive binary refinement algorithm is straightforward: each test point is located with respect to the current tree; the cell where the point is located is recursively divided by two until the size of the resulting cell is less than the characteristic size of the element associated with the test point; this process is repeated for each element edge segment along the curve; each segment mid-point is located in the tree resulting from the refinement due to the previous point. In Fig. 3(b.a) it is shown an example of the binary refinement of a boundary curve of the model of Fig. 3(a). The resulting curve discretization and binary tree are shown in Fig. 3(a.b) and Fig. 3(c) shows the refinement of all boundary curves.



a) Initial mesh b) curve discretization and corresponding binary tree c) boundary refinement

Figure 3. Initial mesh and new boundary refinement.

In relation to recursive domain decomposition, there are several different strategies for automatic meshing of 2D regions. Among them, the ones based on recursive spatial enumeration using quadtree are the most robust. The quadtree technique has been successfully used for 2D finite element meshes for several years. Due to properties of the quaternary tree, it is fast, efficient, and produces a good transition between regions of different degrees of refinement. One of the problems with this technique is that the boundary refinement is dictated by the domain cell decomposition. This produces an irregular generation of boundary nodes with little positioning control. The algorithm, in its original version, cannot conform with a given boundary refinement, which is a very useful property for the combination of different meshing algorithms and is essential for local mesh modifications. The present adaptive meshing strategy combines the advantages of the quadtree, and boundary contraction techniques and it can be considered as a hybrid algorithm between the traditional quadtree approach and the boundary contraction technique. The domain discretization conforms with the "a priori" boundary refinement. No additional nodes are created on the boundary curves (Fig. 4 illustrates the strategy to the L-shape example for all stages).

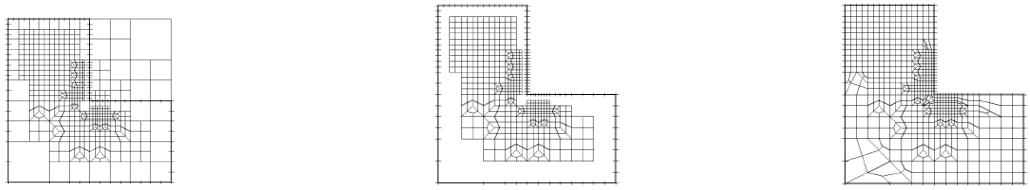


a) Initial quadtree b) quadtree with error analysis c) quadtree for one depth level

d) quadtree and interior mesh e) interior mesh f) resulting mesh

Figure 4. All stages for triangular mesh generation

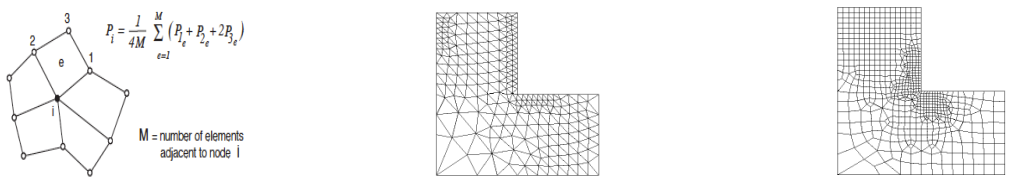
For the generation of quadrilateral elements, a combination of the procedure devised by Potyondy et al. [3] and the templates of the quadtree technique is adopted here. An example, parallel to triangular meshes, is given in Fig. 5, showing only the fases from the quadtree and interior mesh to the resulting mesh, since the previous fases are similar. The boundary contraction algorithm generates quadrilaterals (preferentially) and triangles using pairs of boundary segments. To this effect, pairs of boundary segments and double interior cells (as mentioned previously) in the quadtree are considered. After the generation of quadrilaterals and triangles by the boundary contraction procedure, templates are used for the creation of quadrilateral finite elements. This algorithm requires an even number of boundary segments. This is achieved in the boundary refinement by subdividing the smallest boundary segment of curve is odd. The subdivision of the smallest segment guarantees the boundary gradation.



a) quadtree and interior mesh b) interior mesh c) resulting mesh

Figure 5. Final stages for quadrilateral mesh generation

Finally, a smoothing procedure is applied based on na algorithm proposed by Zhu et al. [20]. This procedure tends to generate elements of optimum shape and is illustrated in Fig. 6 showing the final mesh after the smoothing.



a) smoothing procedure b) final smoothed triangular mesh c) final smoothed quadrilateral mesh

Figure 6. Smoothing procedure and final smoothed meshes

5 Computational experiments

The effectiveness of the adaptive process presented is assessed by means of representative numerical examples. The examples illustrate properties associated with the interconnections among finite element analysis, error estimators/adaptive techniques and automatic mesh generation schemes using quadtree and Delaunay triangulation. The following examples are presented: 1) L-Shaped Plate; 2) Plate with Holes. The first set of examples involves an L-shaped plate. The computational experiments are performed with Q8 elements and using an error tolerance η_{max} = 5%. Three adaptive steps are carried out using both the SPR and REP. Due to lack of space, only results for SPR are shown (Fig. 7) and the process converged in only three steps as it can be seen in Tab. 1. The second example consists of a three-hole plate under plane-strain, for which a complete self-adaptive analysis is performed. This problem has been studied before. However, it has been used only triangular elements, while in the present work, both linear and quadratic, triangular (T3, and T6) and quadrilateral (Q4, and Q8) elements are used. The error tolerance is η_{max} = 10%. Again, due to lack of space, only T3 meshes are shown (Fig. 8) and the process converged in even fewer steps (only two). For quadrilateral meshes, it also took only two steps, as can be seen in Tab. 2 as well as for the other type of elements.

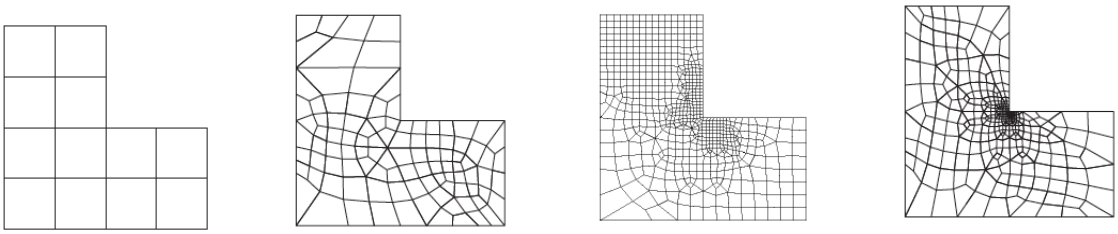


Figure 7. Meshes for the L-Shaped Plate: initial mesh and 1st, 2nd and 3rd refinement steps

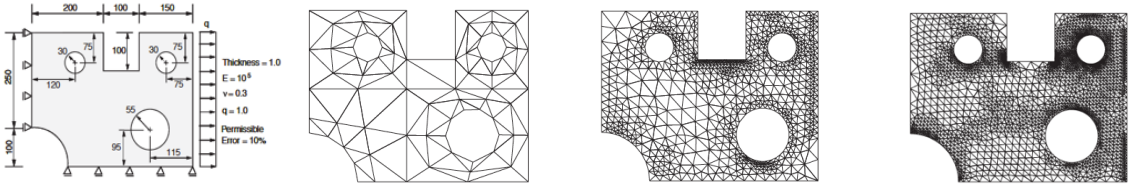


Figure 8. Meshes for the Plate with holes: Model, initial mesh and 1st and 2nd refinement steps

Table 1. Results obtained with the adaptive meshes (using SPR technique)

Step (#)	#Node	#Elem	#DOF	Time (sec)*		Error (%)
				FEA	Meshing	
Initial	53	12	96	0.62	0.58	12.60
1	344	103	670	8.60	1.62	5.28
2	767	242	1516	35.63	3.48	2.83
3	1194	381	2366	92.26	–	1.37

Table 2. Self-adaptive results for plate under plane-strain

Element	Adaptive Step #	#Node	#Elem	#DOF	Time (sec)*		Error (%)
					FEA	Meshing	
T3	(Initial Mesh)	81	118	152	2.78	41.52	28.71
	1	978	1740	1925	88.51	427.05	13.56
	2	2910	5415	5782	737.49	–	7.45
T6	(Initial Mesh)	282	118	546	5.15	5.85	15.91
	1	935	421	1842	19.55	9.26	7.07
	2	1394	642	2758	34.32	–	4.85
Q4	(Initial Mesh)	81	59	152	1.98	43.10	27.79
	1	1438	1335	2846	105.11	422.47	9.72
	2	5204	4964	10346	1310.53	–	4.66
Q8	(Initial Mesh)	223	59	428	3.60	5.88	16.56
	1	1421	439	2808	38.50	19.68	6.10
	2	2766	872	5494	96.93	–	3.83

6 Conclusions

This work addressed a methodology for self-adaptive numerical procedures, more specifically the AMVIEW and FESTA experiences, which relies on the various components of an integrated, object-oriented, computational environment involving pre-, analysis, and post-processing modules. A basic platform for numerical experiments and further development is provided, which allows implementation of new elements/error estimators and sensitivity analysis. A general implementation of the Superconvergent Patch Recovery (SPR) and the Recovery by Equilibrium in Patches (REP) is presented. Both SPR and REP are compared and used for error estimation and for guiding the adaptive remeshing process. Moreover, the SPR is extended for calculating sensitivity quantities of first and higher orders. The mesh (re-)generation process is accomplished by means of methods combining quadtree and Delaunay triangulation techniques. Surface mesh generation in arbitrary domains is performed automatically (i.e. with no user intervention) during the self-adaptive analysis using either quadrilateral or triangular elements. The effectiveness and versatility of both software are demonstrated by representative numerical examples illustrating the interconnections among finite element analysis, recovery procedures, error estimation/adaptivity and automatic mesh generation.

Authorship statement. The authors hereby confirm that they are the sole liable persons responsible for the authorship of this work, and that all material that has been herein included as part of the present paper is either the property (and authorship) of the authors, or has the permission of the owners to be included here.

References

[1] Baehmann PL, Wittchen SL, Shephard MS, Grice KR, Yerry MA. “Robust geometrically based, automatic two-dimensional mesh generation”. Int. J. Num. Meth. Eng. 24(6):1043-1078, 1987.
[2] Zhu JZ, Zienkiewicz OC, Hinton E, Wu J, “A new approach to the development of automatic quadrilateral mesh generation”. Int. J. Num. Meth. Eng. 32(4):849-866, 1991.
[3] Potyondy DO, Wawrzynek PA, Ingraffea AR, “An algorithm to generate quadrilateral or triangular element surface meshes in arbitrary domains with applications to crack propagation”. Int. J. Num. Meth. Eng. 38(16):2677-2701, 1995.
[4] Borouchaki H, Frey PJ, “Adaptive triangular-quadrilateral mesh generation”. Int. J. Num. Meth. Eng. 41(5):915-934, 1998.
[5] Cass RJ, Benzley SE, Meyers RJ, Blacker TD, “Generalized 3-D paving: An automated quadrilateral surface mesh generation algorithm”. Int. J. Num. Meth. Eng. 39(9):1475-1489, 1996.
[6] Beall MW, Dey S, Klaas O, Shephard MS, “Adaptive analysis within a geometry-based analysis framework”. In: Fourth U.S. National Congress on Computational Mechanics, p. 16, San Francisco, CA, 1997.
[7] Lo SH, “3D mesh refinement in compliance with a specified node spacing function”. Comput. Mech. 21(1):11-19, 1998.
[8] Babuska I, Rheinboldt WC. “A posteriori error estimates for the finite element method”. Int. J. Num. Meth. Eng. 12(10):1597-1615, 1978.
[9] Zienkiewicz OC, Kelly DW, Gago J, Babuska I, “Hierarquical finite element approaches, error estimator and adaptive refinement”. In: Whiteman, J. (Ed.), Mathematics of Finite Elements and Applications (IV). Academic Press, New York, 1982.
[10] Shephard MS, “Adaptive finite element analysis and CAD”. In: Babuska, I. et al. (Eds.), Accuracy Estimates and Adaptive Refinements in Finite Element Computations, pp. 205-225. John Wiley & Sons, 1986.
[11] Zienkiewicz OC, Zhu JZ, “A simple error estimator and adaptive procedure for practical engineering analysis”. Int. J. Num. Meth. Eng. 24(2):337-357, 1987.
[12] Zienkiewicz OC, Zhu JZ, “The superconvergent patch recovery and a posteriori error estimates. Part 1: The recovery technique”. Int. J. Num. Meth. Eng. 33(7):1331-1364, 1992.
[13] Zienkiewicz OC, Zhu JZ. “The superconvergent patch recovery and a posteriori error estimates. Part 2: Error estimates and adaptivity”. Int. J. Num. Meth. Eng. 33(7):1365-1382, 1992.
[14] Zienkiewicz OC, Zhu JZ. “The SPR recovery and boundaries”. Int. J. Num. Meth. Eng. 37(18):3195±3196, 1994.
[15] Babuska I, Strouboulis T, Upadhyay CS, Gangaraj SK, Copps K, “Validation of a posteriori error estimators by a numerical approach”. Int. J. Num. Meth. Eng. 37(7):1073-1123, 1994.
[16] Boroomand B, Zienkiewicz OC, “Recovery by equilibrium in patches (REP)”. Int. J. N. Meth. Eng. 40(1):137-164, 1997.
[17] Paulino GH, Shi F, Mukherjee S, Ramesh P, “Nodal sensitivities as error estimates in computational mechanics”. Acta Mechanica 121:191-213, 1997.
[18] Mahomed N, Kekana M, “An error estimator for adaptive mesh refinement analysis based on strain energy equalization”. Comput. Mech. 22(4):355-366, 1998.
[19] Ladeveze P, Oden JT, “Advances in Adaptive Computational Methods. Studies in Applied Mechanics”, Elsevier, 1998.
[20] Zhu JZ, Zienkiewicz OC, Hinton E, Wu J, “A new approach to the development of automatic quadrilateral mesh generation”. Int. J. Num. Meth. Eng. 32(4):849-866, 1991.