Received: 31 May 2019 | Accepted: 3 June 2019

DOI: 10.1002/pamm.201900419

A posteriori error control for the finite cell method

Paolo Di Stolfo[1], **Alexander Düster**[2], **Stefan Kollmannsberger**[3], **Ernst Rank**[3], and **Andreas Schröder**[1,*]

[1] Department of Mathematics, University of Salzburg, Austria
[2] Numerical Structural Analysis with Application in Ship Technology (M-10), Hamburg University of Technology, Germany
[3] Chair for Computation in Engineering, Technische Universität München, Germany

The paper presents some concepts of the finite cell method and discusses a posteriori error control for this approach. The focus is on the application of the dual weighted residual approach (DWR), which enables the control of the error with respect to a user-defined quantity of interest. Since both the discretization error and the quadrature error are estimated, the application of the DWR approach provides an adaptive strategy which equilibrates the error contributions resulting from discretization and quadrature. The strategy consists in refining either the finite cell mesh or its associated quadrature mesh. Numerical experiments confirm the performance of the error control and the adaptive scheme for a non-linear problem in 2D.

1 Introduction

The Finite Cell Method (FCM) is designed for the simulation of the mechanical behavior of a body by using an unfitted mesh and enables computations on extremely complex shapes. The FCM was originally introduced in [1] as an immersed boundary method and has been successfully applied to a variety of problems in solid mechanics, cf., e.g., [2, 3]. Its basic idea is to combine a fictitious domain approach with a finite element method, where the possibly complicated physical domain is replaced by an embedding domain of a geometrically simple shape, for instance, a (paraxial) quadrilateral in 2D or a (paraxial) hexahedron in 3D, which can easily be meshed. The variational formulation of the problem and its finite element discretization are defined on the embedding domain. The geometry of the physical domain is incorporated via an indicator function which necessitates the implementation of an appropriate quadrature scheme. This can be done by using an additional quadrature mesh, which approximates the physical domain and, thus, implies a certain quadrature error. Hence, mainly two computational error sources occur in the finite cell method: the discretization error and the quadrature error.

In this paper, we present some concepts and applications of the finite cell method and discuss a posteriori error control for this method. The focus is on the application of the dual weighted residual approach (DWR) as presented in [5]. It enables the control of the error with respect to a user-defined quantity of interest. The aim is to estimate both the discretization error and the quadrature error. The application of the DWR approach provides an adaptive strategy in which the error contributions resulting from discretization and quadrature are balanced. The strategy consists in refining either the finite cell mesh or its associated quadrature mesh. In several numerical experiments the performance of the error control and the adaptive scheme is demonstrated for a non-linear problem in 2D.

2 Finite cell method

In this section, we briefly introduce a general abstract setting of the finite cell method (FCM). Typically, one seeks a solution $u \in V$ of an operator equation $A(u) = 0$, where the operator $A : V \to V^*$ maps each element of a suitable function space V into its dual space V^*. A prototypical example for such an operator is $A(u)(v) := \int_\Omega \nabla u \nabla v \, dx - \int_\Omega f\, v \, dx$ encoding Poisson's problem in its weak formulation. Here, the operator is defined on the Sobolev space $V = H^1_{0,\Gamma_D}(\Omega)$ with Dirichlet zero data on the part Γ_D of the Lipschitz domain Ω, see Fig. 1a. The application of the FCM to such a problem consists of two steps. First, an enclosing domain $\widehat{\Omega} \supset \Omega$ is chosen which allows for a simple discretization, e.g., a rectangle in 2D or a cuboid in 3D, see Fig. 1b. Then, a function space $\widehat{V}$ on $\widehat{\Omega}$ is constructed, where an operator $\widehat{A} : \widehat{V} \to \widehat{V}^*$ replacing

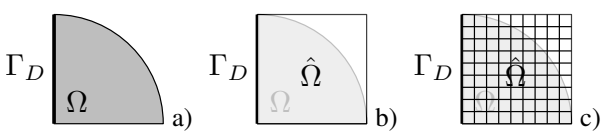

Fig. 1: a) Domain Ω, **b)** embedding of Ω into $\widehat{\Omega}$, **c)** paraxial mesh for $\widehat{\Omega}$

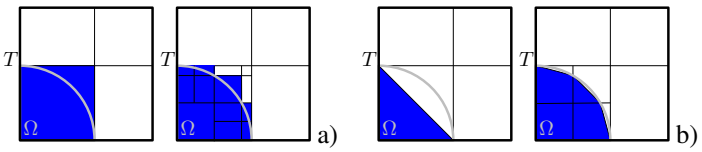

Fig. 2: a) Piecewise constant approximation, **b)** piecewise linear approximation of the boundary

the original operator A is defined. It is clear that this introduces a model error and, in particular, $\widehat{A}$ needs to be reasonably close to A. In the case of the Poisson problem, $\widehat{A}$ may be chosen as $\widehat{A}(\widehat{u})(\widehat{v}) := A(\widehat{u}|_\Omega)(\widehat{v}|_\Omega) + \epsilon \int_{\widehat{\Omega}\setminus\Omega} \nabla\widehat{u} \cdot \nabla\widehat{v}\, dx$. The

* Corresponding author: e-mail andreas.schroeder@sbg.ac.at

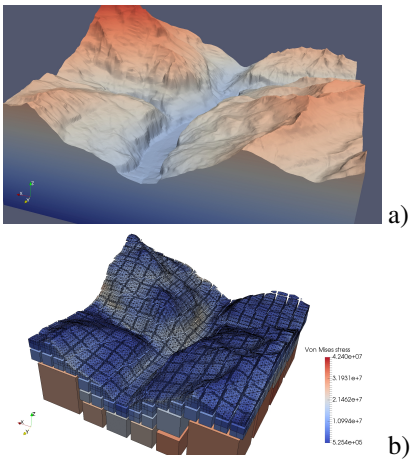

Fig. 3: Computation of stress distributions in mountain regions: **a)** Lauterbrunnen Valley, **b)** von Mises stresses

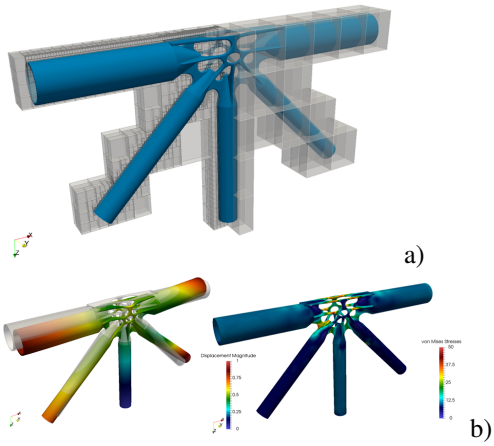

Fig. 4: Computation of discplacements/stresses in a steel joint: **a)** Octree, **b)** displacements and von Mises stresses

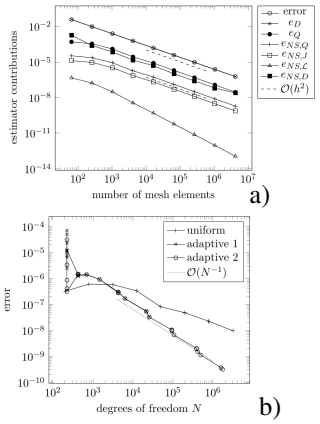

Fig. 5: **a)** Reduction of the error contribution with uniform refinements (mesh size h), **b)** comparison of uniform and adaptive refinements

second step consists in defining a finite-element space $V_h \subset \widehat{V}$ on a mesh $\mathcal{T}$ of $\widehat{\Omega}$. Due to the choice of $\widehat{\Omega}$, the meshing process is simplified substantially as, e.g., a paraxial mesh may be chosen, see Fig. 1c. The integrals of functions defined on Ω arising in the definition of the operator $\widehat{A}$ cannot be computed as usual since no triangulation of Ω is available. Instead, the possibly complicated geometry of elements cut by the boundary needs to be captured by the numerical integration. For this, several techniques have been developed. A first class of techniques aims at exact integration, i.e., the error attains the level of machine precision. One drawback of these techniques is the possibly high computational effort. Moreover, it is not necessary to perform exact integration if the overall error is dominated by the approximation error. Hence, it is often more convenient to adapt the quadrature such that the quadrature error is at a similar level as the approximation error. Such an adaptation of the quadrature necessitates the use of refinable quadrature schemes which comprise the second class of widely-used FCM quadrature techniques. Here, a quadrature level $\ell : \mathcal{T} \to \mathbb{N}_0$ is defined on the mesh which translates into individual approximations of each $T \cap \Omega$. Their union yields an approximate domain $\Omega^{(\ell)} \approx \Omega$. These schemes are usually constructed such that an increase in $\ell(T)$ delivers an improved approximation of the geometry of $T \cap \Omega$. Prototypical examples of such a refinable quadrature scheme are the quadtree in 2D and the octree in 3D. Here, the complicated geometry of $T \cap \Omega$ is captured by $\ell(T)$ recursive refinements of T towards the boundary of Ω passing through T, see Fig. 2a. Note that the quadtree delivers only a piecewise constant approximation of $\partial\Omega$, which implies that a large level ℓ is required to sufficiently resolve the boundary. It is therefore practical to use a piecewise linear approximation to $\partial\Omega$ as shown in Fig. 2b. By using inexact integration, the original operator $\widehat{A}$ is replaced by its perturbation $\widehat{A}^{(\ell)}$ where each integral in the definition of $\widehat{A}$ is replaced by the quadrature rule based on $\Omega^{(\ell)}$. Thus, in the application of the FCM, a perturbed discrete solution $u_h^{(\ell)} \in V_h$ with $\widehat{A}^{(\ell)}(u_h^{(\ell)})(v_h) = 0$ for all $v_h \in V_h$ is computed. The most important feature of the FCM is to resolve geometries with very complicated or fine structures at the boundary. Fig. 3a shows an example of such a geometry: a large mountain region with a complex and fine-structured surface (the Lauterbrunnen Valley in Switzerland). Such a terrain's surface is typically represented by a digital elevation model (DEM) as a heightmap. The application of the marching volume polytopes algorithm as introduced in [7] gives a piecewise linear approximation of the surface, see Fig. 3b. The resulting volume mesh can directly be used in the FCM, for instance, to compute stress distributions in order to detect potential sites of rock failure and to deduce erosive processes in mountain peaks, rock slopes and bedrocks. The FCM approach can also be applied to geometries with varying boundaries (in time). For instance, in [6] the varying boundary in an NC-milling process is resolved by using the marching volume polytopes algorithm in order to simulate thermoelastic deformations. We emphasize that using FCM drastically reduce the engineering effort for pre-processing (almost no cost for meshing) and one benefits from the high robustness of the method with respect to imprecise or even flawed CAD models [8]. Modeling flaws occur frequently in B-rep models used in everyday engineering practice. An example of a B-Rep model is given in Fig. 4a. It represents a steel joint and was supplied by ITE@Braunschweig. The B-Rep model is submerged into a non-boundary conforming computational grid and the FCM is applied to compute the spatial distribution of the von Mises stresses depicted in Fig. 4b. A hierarchical mesh refinement of two levels is applied and an octree integration is used as a refinable quadrature scheme with a fixed depth of $\ell(T) = 3$. The model contains only 3 360 degrees of freedom and the corresponding system can be solved using a direct solver in only 129 ms on a standard personal computer. However, the octree quadrature of the system matrices takes approx. two minutes. That is, one thousand times more time is spent on integration than on solving. This indicates the potential gain by including an estimation of the integration error (as proposed in the next section) when solving problems of industrial relevance.

3 A posteriori error control

While classical error control often aims for estimates in the energy or the L_2 error, we focus on goal-oriented error control. Here, estimates for the error $|J(u) - J(u_h)|$ measured with respect to a goal functional $J : V \to \mathbb{R}$ are sought. The goal functional encodes a user-defined quantity of interest such as point evaluations, averages over subsets, line integrals, or nonlinear quantities such as $|\nabla u(p)|^2$ for $p \in \Omega$. In the setting of the FCM, error contributions resulting not only from the discretization, but also from the quadrature need to be considered. In this section, an a posteriori error estimate allowing for a separation of these two error sources is presented. To this end, the dual weighted residual method (DWR) as introduced in [4] is extended to the FCM. The main idea is to represent the goal functional J and its approximation $J^{(\ell)}$ by means of solutions $z \in V$ and $z_h^{(\ell)} \in V_h$ of the dual problems $A'(u)(v, z) = J'(u)(v)$ for all $v \in V$ and $\widehat{A}'^{(\ell)}(u_h^{(\ell)})(v_h, z_h^{(\ell)}) = J'^{(\ell)}(u_h^{(\ell)})(v_h)$ for all $v_h \in V_h$. The following error representation is derived in [5] and contains the evaluation of the approximated operators as well as the exact operators. The evaluation of the latter involves integrals on Ω which cannot be calculated exactly. To overcome this problem, a refineable quadrature scheme $\ell : \mathcal{T} \to \mathbb{R}$ is used. By performing a sufficiently large number k of additional refinements on each element, the error incurred by the quadrature is negligibly small. The operators whose integrals are based on the refinable quadrature schemes are indexed by ℓ and $\ell + k$.

Theorem 3.1 *Let $u^+, z^+ \in V^+$ be approximations of u and z in a higher-order approximation space V^+ with $\widehat{V} \supseteq V^+ \supseteq V_h$. Then,*

$$J(u) - J^{(\ell+k)}(u_h^{(\ell)}|_\Omega) = e_D^{(\ell+k)} + e_Q^{(\ell+k)} + e_{NS,Q} + e_{NS,J} + e_{NS,\mathcal{L}} + e_{NS,D} + e_\epsilon + \mathcal{R}_h^{(3)},$$

where $\mathcal{R}_h^{(3)}$ is a higher-order remainder.

The terms $e_D^{(\ell+k)}$ and $e_Q^{(\ell+k)}$ are the dominating discretization and quadrature error contributions. They are defined as

$$e_D^{(\ell+k)} := \frac{1}{2}(\rho(z^+|_0 - (z_h^{(\ell)})_0) + \rho^*(u_0^+ - (u_h^{(\ell)})_0)), \quad e_Q^{(\ell+k)} := -\widehat{A}^{(\ell+k)}(u_h^{(\ell)})(z_h^{(\ell)})$$

in terms of the primal and dual residuals ρ and ρ^* given by

$$\rho(\cdot) := -\widehat{A}^{(\ell+k)}((u_h^{(\ell)})_0)(\cdot), \quad \rho^*(\cdot) := J'^{(\ell+k)}(u_h^{(\ell)}|_\Omega)(\cdot|_\Omega) - \widehat{A}'^{(\ell+k)}((u_h^{(\ell)})_0)(\cdot, (z_h^{(\ell)})_0).$$

The notation $(y)_0$ for a function y on $\hat{\Omega}$ denotes the extension of the restriction $y|_\Omega$ by zero onto $\hat{\Omega}$. The error contributions $e_{NS,\cdot}$ mainly consist of differences of evaluations between the exact operators and their approximations based on improved quadrature rules and can, thus, be assumed to be negligibly small. Abbreviating the primal error $e := u - u_h^{(\ell)}|_\Omega$ and the dual error $e^* := z - z_h^{(\ell)}|_\Omega$, these contributions are

$$\begin{aligned}
e_{NS,D} &:= \frac{1}{2}(\rho(z_0 - z^+|_0) + \rho^*(u_0 - u^+|_0)), \quad e_{NS,Q} := \widehat{A}^{(\ell+k)}((u_h^{(\ell)})_0)((z_h^{(\ell)})_0) - A(u_h^{(\ell)}|_\Omega)(z_h^{(\ell)}|_\Omega),\\
e_{NS,J} &:= (J - J^{(\ell+k)})(u_h^{(\ell)}|_\Omega), \quad e_\epsilon := \widehat{A}^{(\ell+k)}(u_h^{(\ell)})(z_h^{(\ell)}) - \widehat{A}^{(\ell+k)}((u_h^{(\ell)})_0)((z_h^{(\ell)})_0) = \mathcal{O}(\epsilon),\\
e_{NS,\mathcal{L}} &:= \frac{1}{2}\Big((J' - J'^{(\ell+k)})(u_h^{(\ell)}|_\Omega)(e)) - (A'(u_h^{(\ell)}|_\Omega)(e, z_h^{(\ell)}|_\Omega) - \widehat{A}'^{(\ell+k)}((u_h^{(\ell)})_0)(e_0, (z_h^{(\ell)})_0)\\
&\qquad -(A(u_h^{(\ell)}|_\Omega)(e^*) - \widehat{A}^{(\ell+k)}((u_h^{(\ell)})_0)((e^*)_0))\Big).
\end{aligned}$$

4 Numerical results

In this section, we present the results of two numerical examples from [5]. In the first one, the claim that the terms $e_{NS,\cdot}$ in the error representation are negligibly small compared to the dominating error terms e_D and e_Q is validated. We consider the semilinear problem $-\Delta u + u^3 = f$ on a quarter disk $\Omega := B_1(0) \cap (0,1)^2$ with $\Gamma_D := ([0,1] \times \{0\}) \cup (\{0\} \times [0,1])$ and $\Gamma_N := \partial\Omega \setminus \Gamma_D$. The domain is embedded into the unit square $\widehat{\Omega} := (0,1)^2$. The operator associated with the problem reads $A(u,v) := \int_\Omega \nabla u \cdot \nabla v + u^3 v \, dx - \int_\Omega f v \, dx - \int_{\Gamma_N} g v \, ds$. The data f, g and J are chosen according to $u(x,y) := \sin(\pi x)\sin(\pi y)$ and $z(r,\varphi) := (-r^2 + 2r)(-\frac{16}{\pi^2}\varphi^2 + \frac{8}{\pi}\varphi)$. We use a piecewise linear quadrature scheme and set the quadrature refinement level $\ell = 2$ as well as the additional level $k = 2$ and perform uniform mesh refinement of lowest-order finite elements. In Fig. 5a, the decrease of error terms is $O(h^2)$ as expected. Moreover, the terms $e_{NS,\cdot}$ are indeed at least one order of magnitude smaller than e_D, e_Q.

In the second example, automatic mesh adaptation is performed. The usual Solve–Estimate–Mark–Refine loop needs to be modified to account for the adaptation of the quadrature mesh, see Alg. 1. As a numerical experiment, we consider the semilinear problem from the first example, but on a domain $\Omega := B_1(0) \setminus ([0,1] \times [-1,0])$ with $\Gamma_D := ([0,1] \times \{0\}) \cup (\{0\} \times [-1,0])$ and $\Gamma_N := \partial\Omega \setminus \Gamma_D$ which is embedded into the L-shaped domain $\widehat{\Omega} := (-1,1)^2 \setminus ([0,1] \times [-1,0])$.

Algorithm 1 Solve–Estimate–Mark–Refine loop with quadrature mesh adaptation

Require: Coarse level $\ell(T) = r \in \mathbb{N}_0$, $T \in \mathcal{T}$, additional quadrature levels $k \in \mathbb{N}$, error ratio $\sigma \in (0, 1)$
repeat
Setup the quadrature meshes according to $\ell, \ell + k$
Solve the primal and dual problem with the quadrature mesh based on ℓ and compute e_D, e_Q with level $\ell + k$
if $e_Q > \sigma \cdot e_D$ **then**
Localize e_Q and increase ℓ for mesh elements with highest contributions
else
Localize e_D (e.g. by filtering, see [5] and references therein)
Mark mesh elements with highest contributions, refine $\mathcal{T}$, update ℓ and reduce ℓ by 1 on new mesh elements
end if
until termination criterion is met

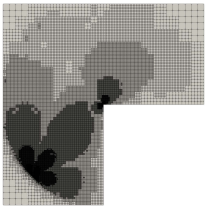

Fig. 6: Refined mesh from [5]

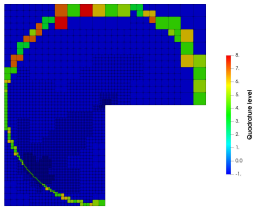

Fig. 7: Quadrature levels from [5]

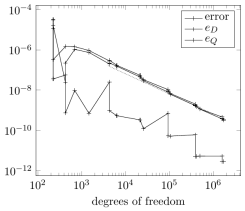

Fig. 8: Error and dominating error terms

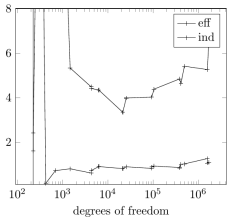

Fig. 9: Estimation efficiency

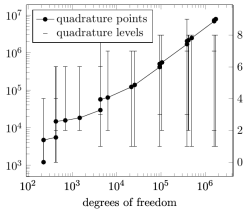

Fig. 10: Quadrature points, range of levels

The data f, g are chosen according to $u(x, y) := u(r, \phi) = r^{2/3} \sin(\frac{2}{3}\phi)$. To perform error control for the goal functional $J(u) := |\nabla u|^2(p)$ at the point $p := (\cos(\frac{5}{4}\pi), \sin(\frac{5}{4}\pi)) \in \partial\Omega$, the regularization $J(v) := \int_{\widehat{\Omega}} \tilde{I}_S |\nabla v|^2$ with a smoothed indicator function $\tilde{I}_S$ with $S := B_{0.01}(p) \cap \Omega$ is considered. The results comparing uniform refinement with two variants of adaptive refinement are displayed in Fig. 5b, where the parameter σ in Alg. 1 is chosen as $\sigma := 0.01$. The variant "adaptive 1" creates a quadrature rule based on the quadtree, while "adaptive 2" performs a piecewise linear approximation of $\partial\Omega$. The quadtree computation stops at approx. $N = 10\,000$ due to the large number of quadrature points, which demonstrates the superiority of the linear approximation. While the adaptive refinement yields an optimal algebraic convergence of $O(N^{-1})$, uniform refinement does not attain the optimal rate. In Fig. 6, adaptive mesh refinements towards the reentrant corner and the point p can be seen. The quadrature levels after 12 steps of the refinement loop are displayed in Fig. 7. In Fig. 8, the effect of the adaptive strategy is visualized. Whenever e_Q is too large, only the quadrature mesh is refined while the finite element mesh remains unchanged. This leads to a reduction of the quadrature error at the same N. The overestimation of the error by the estimator, eff, and the overestimation of the error by the sum of the absolute values of the error indicators, ind, is displayed in Fig. 9: These numbers are bounded between 1.5 and 8 for meaningful problem sizes and, therefore, indicate a good representation of the error by the estimator. Finally, Fig. 10 displays the linear increase in the total number of quadrature points for the piecewise linear quadrature and the range of the quadrature level ℓ per computation step, which shows a local adaptation of the quadrature. We refer to [5] where further numerical experiments are presented, in particular, in the context of model plasticity with linear isotropic hardining. Moreover, implementation aspects as, for instance, the computation of u^+ and z^+ and localization techniques are discussed in more detail.

References

[1] Parvizian, J. and Düster, A. and Rank, E. Finite cell method – h- and p-extension for embedded domain problems in solid mechanics, Computational Mechanics, **41** (1), 121–133 (2007)
[2] Schillinger, D. and Ruess, M. and Zander, N. and Bazilevs, Y. and Düster, A. and Rank, E.: Small and large deformation analysis with the p- and B-spline versions of the Finite Cell Method. Computational Mechanics **50** (4) 445–478 (2012)
[3] Abedian, A. and Parvizian, J. and Düster, A. and Rank, E.: The finite cell method for the J2 flow theory of plasticity. Finite Elements in Analysis and Design **69** 37–47 (2013)
[4] Becker, R., and Rannacher, R.: An optimal control approach to a posteriori error estimation in finite element methods. Acta numerica **10** 1–102 (2001).
[5] P. Di Stolfo, A. Rademacher, and A. Schröder: Dual weighted residual error estimation for the finite cell method, Journal of Numerical Mathematics, published online (2019).
[6] A. Byfut, A. Schröder: A fictitious domain method for the simulation of thermoelastic deformations in NC-milling processes. Int. J. Numer. Meth. Engng. **113** (2), 208–229 (2017).
[7] A. Byfut, F. Hellwig, A. Schröder: Marching volume polytopes algorithm, Int. J. Numer. Meth. Engng. **117** (12), 1171–1204 (2019).
[8] B. Wassermann, S. Kollmannsberger, S. Yin, L. Kudela, and E. Rank: Integrating CAD and Numerical Analysis – 'Dirty Geometry' handling using the Finite Cell Method, Comp. Meth. in Appl. Mech. and Engng. **351** (13) 808–835 (2019)

