Adaptive SQP Method for Shape Optimization

P. Morin, R. H. Nochetto, M. S. Pauletti and M. Verani

Abstract We examine shape optimization problems in the context of inexact sequential quadratic programming. Inexactness is a consequence of using adaptive finite element methods (AFEM) to approximate the state equation, update the boundary, and compute the geometric functional. We present a novel algorithm that uses a dynamic tolerance and equidistributes the errors due to shape optimization and discretization, thereby leading to coarse resolution in the early stages and fine resolution upon convergence. We discuss the ability of the algorithm to detect whether or not geometric singularities such as corners are genuine to the problem or simply due to lack of resolution — a new paradigm in adaptivity.

1 Shape Optimization as Adaptive Sequential Quadratic Programming

Shape optimization problems governed by partial differential equations (PDE) can be formulated as constrained minimization problems with respect to the shape of a domain Ω in $\mathbb{R}^d$. If $u = u(\Omega)$ is the solution of a PDE in Ω, the state equation is

$$\mathscr{A} u(\Omega) = f, \tag{1}$$

and $J(\Omega) = J(\Omega, u(\Omega))$ is a cost functional, then we consider the minimization problem

P. Morin
Instituto de Matemática Aplicada del Litoral, Universidad Nacional del Litoral, CONICET, Santa Fe, Argentina, e-mail: pmorin@santafe-conicet.gov.ar. Partially supported by Universidad Nacional del Litoral through Grant CAI+D PI-62-312, and CONICET through Grant PIP 112-200801-02182.

R. H. Nochetto
Department of Mathematics and Institute for Physical Science and Technology, University of Maryland, College Park, USA e-mail: rhn@math.umd.edu. Partially supported by NSF grants DMS-0505454 and DMS-0807811.

M. S. Pauletti
Department of Mathematics, University of Maryland, College Park, and Department of Mathematics, Texas A&M, USA e-mail: seba@math.umd.edu. Partially supported by NSF grants DMS-0505454 and DMS-0807811.

M. Verani
MOX - Dipartimento di Matematica “F. Brioschi”, Politecnico di Milano, Milano, Italy e-mail: marco.verani@polimi.it. Partially supported by Italian FIRB RBIP06HF8S.

$$\Omega^* \in \mathscr{U}_{ad}: \qquad J(\Omega^*) = \inf_{\Omega \in \mathscr{U}_{ad}} J(\Omega), \tag{2}$$

within the set $\mathscr{U}_{ad}$ of admissible domains in $\mathbb{R}^d$. This is a constrained minimization problem for J.

In this paper we formulate an Adaptive Sequential Quadratic Programming algorithm (or ASQP), that adaptively builds a sequence of domains $\{\Omega_k\}_{k\geq 0}$ converging to a local minimizer of the shape optimization problem (1)–(2). To motivate and briefly describe the ideas underlying ASQP, we need the concept of shape derivative $dJ(\Omega;\mathrm{w})$ of $J(\Omega)$ in the direction of a normal velocity w

$$dJ(\Omega;\mathrm{w}) = \int_\Gamma g(\Omega)\mathrm{w}, \tag{3}$$

see [13] for its precise definition. We observe that $g(\Omega)$, the *Riesz representation* of the shape derivative $dJ(\Omega)$, depends on $u(\Omega)$. We present ASQP in two steps: we first introduce an infinite dimensional Sequential Quadratic Programming (Exact SQP) algorithm, and next we introduce and motivate its adaptive finite dimensional version, responsible for the inexact nature of ASQP.

Exact SQP Algorithm. We let Ω_k be the current iterate and Ω_{k+1} be the new one. We let $\Gamma_k := \partial\Omega_k$ and let $\mathbb{V}(\Gamma_k)$ be a Hilbert space defined on Γ_k, with scalar product $b_{\Gamma_k}(\cdot,\cdot): \mathbb{V}(\Gamma_k)\times\mathbb{V}(\Gamma_k)\to\mathbb{R}$ and norm $\|\cdot\|_{\mathbb{V}(\Gamma_k)}$. This gives rise to the elliptic selfadjoint operator $\mathscr{B}_k : \mathbb{V}(\Gamma_k)\to\mathbb{V}(\Gamma_k)^*$ defined by $\langle\mathscr{B}_k\mathrm{v},\mathrm{w}\rangle_{\Gamma_k} = b_{\Gamma_k}(\mathrm{v},\mathrm{w})$. We then consider the following *quadratic model* $Q_k : \mathbb{V}(\Gamma_k)\to\mathbb{R}$ of J around Ω_k

$$Q_k(\mathrm{w}) := J(\Omega_k) + dJ(\Omega_k;\mathrm{w}) + \frac{1}{2}\langle\mathscr{B}_k\mathrm{w},\mathrm{w}\rangle. \tag{4}$$

It is easy to check that the unique minimizer v_k of $Q_k(\mathrm{w})$ satisfies

$$\mathrm{v}_k \in \mathbb{V}(\Gamma_k): \qquad b_{\Gamma_k}(\mathrm{v}_k,\mathrm{w}) = -\langle g_k,\mathrm{w}\rangle_{\Gamma_k} \qquad \forall \mathrm{w}\in\mathbb{V}(\Gamma_k), \tag{5}$$

with $g_k := g(\Omega_k)$; i.e. $\mathrm{v}_k = -\mathscr{B}_k^{-1}g_k$. Moreover, v_k is an admissible descent direction; i.e. $dJ(\Omega_k;\mathrm{v}_k)<0$ because $b_{\Gamma_k}(\cdot,\cdot)$ is a scalar product. Once v_k has been found, we need to determine a stepsize that is not too small and guarantees sufficient decrease of the functional J. To accomplish this goal we identify a range of admissible stepsizes by adapting the classical *Armijo-Wolfe conditions* in $\mathbb{R}^n$: given $0<\alpha<\beta<1$, we seek a stepsize $\mu\in\mathbb{R}^+$ satisfying

$$J(\Omega_k+\mu\mathbf{v}_k) \leq J(\Omega_k) + \alpha\mu\; dJ(\Omega_k;\mathrm{v}_k), \qquad dJ(\Omega_k+\mu\mathbf{v}_k;\mathrm{v}_k) \geq \beta\; dJ(\Omega_k;\mathrm{v}_k), \tag{6}$$

where $\partial(\Omega_k+\mu\mathbf{v}_k) := \{\mathbf{y}\in\mathbb{R}^d:\ \mathbf{y}=\mathbf{x}+\mu\mathbf{v}_k(\mathbf{x}),\ \mathbf{x}\in\partial\Omega_k\}$ is the updated domain boundary and $\mathbf{v}_k = \mathrm{v}_k\boldsymbol{\nu}_k$ is a normal vector field. We are now ready to introduce the Exact Sequential Quadratic Programming algorithm for solving the constrained optimization problem (1)–(2): given the initial domain Ω_0, set $k=0$ and iterate

Compute $u_k = u(\Omega_k)$ by solving (1)
Compute the Riesz representation $g_k = g(\Omega_k)$ of (3)
Compute the search direction v_k by solving (5)
Determine an admissible stepsize μ_k satisfying (6)
Update: $\Omega_{k+1} = \Omega_k + \mu_k\mathbf{v}_k;\ \ k\leftarrow k+1$

This algorithm is not feasible as it stands, because it requires the exact computation of the following quantities at each iteration: the solution u_k to the state equation (1); the solution v_k to the linear subproblem (5);

the values of the functional J and of its derivative dJ in the line search routine. Replacing all of the above non-computable operations by finite approximations yields a practical algorithm.

Adaptive SQP Algorithm (ASQP). This method adjusts the accuracies of the various approximations relative to the energy decrease for each iteration. It is worth noticing that the adaptive procedure driving our algorithm has to deal with two distinct sources of error:

- *PDE Error*: this hinges on the approximation of (1) and the values of the functional J and its derivative (3);
- *Geometric Error*: this relates to the approximation of (5) which yields the new domain.

Since it is wasteful to impose a PDE error finer than the expected geometric error, we have a natural mechanism to balance the computational effort. The ASQP algorithm is an iteration of the form:

$$\ldots \rightarrow \mathscr{E}_k \rightarrow \texttt{APPROXJ} \rightarrow \texttt{SOLVE} \rightarrow \texttt{RIESZ} \rightarrow \texttt{DIRECTION} \rightarrow \texttt{LINESEARCH} \rightarrow \texttt{UPDATE} \rightarrow \mathscr{E}_{k+1} \rightarrow \ldots$$

where $\mathscr{E}_k = \mathscr{E}_k(\Omega_k, \mathbb{S}_k, \mathbb{V}_k)$ is the total error incurred in at step k, $\mathbb{S}_k = \mathbb{S}_k(\Omega_k)$ is the finite element space defined on Ω_k and $\mathbb{V}_k = \mathbb{V}_k(\Gamma_k)$ is the finite element space defined on the boundary Γ_k. To describe briefly each module along with the philosophy behind ASQP, we let G_k be an approximation to the shape derivative $g_k = g(\Omega_k)$ given by RIESZ, and $V_k \in \mathbb{V}_k(\Gamma_k)$ be an approximation to the exact solution $\mathrm{v}_k \in \mathbb{V}(\Gamma_k)$ of (5) given by DIRECTION. The discrepancy between v_k and V_k leads to the geometric error. Upon using a first order Taylor expansion around Ω_k, together with (5) for the exact velocity v_k, we obtain

$$\big|J(\Omega_k + \mu_k \mathbf{V}_k) - J(\Omega_k + \mu_k \mathbf{v}_k)\big| \simeq \mu_k \big|dJ(\Omega_k; V_k - \mathrm{v}_k)| = \mu_k \big|b_{\Gamma_k}(\mathrm{v}_k, V_k - \mathrm{v}_k)\big| \leq \mu_k \|\mathrm{v}_k\|_{\Gamma_k} \|\mathrm{v}_k - V_k\|_{\Gamma_k}.$$

Motivated by this expression, we now describe the modules APPROXJ and DIRECTION , in which adaptivity is carried out. These modules are driven by different adaptive strategies and corresponding different tolerances, say a PDE tolerance γ and a geometric tolerance θ. Their relative values allow for different distributions of the computational effort in dealing with the PDE and the geometry. The routine DIRECTION enriches/coarsens the space $\mathbb{V}_k$ to control the quality of the descent direction:

$$\|V_k - \mathrm{v}_k\|_{\Gamma_k} \leq \theta \|V_k\|_{\Gamma_k}, \tag{7}$$

where $\theta \leq 1/2$ guarantees that the angle between V_k and v_k is $\leq \pi/6$; in particular $\|\mathrm{v}_k\|_{\Gamma_k} \leq (1+\theta)\|V_k\|_{\Gamma_k}$. This implies a geometric error proportional to $\mu_k \|V_k\|^2_{\Gamma_k}$, namely

$$\big|J(\Omega_k + \mu_k \mathbf{V}_k) - J(\Omega_k + \mu_k \mathbf{v}_k)\big| \leq \delta \mu_k \|V_k\|^2_{\Gamma_k}, \tag{8}$$

with $\delta := \theta(1+\theta) \leq \frac{3}{2}\theta$. On the other hand, the module APPROXJ enriches/coarsens the space $\mathbb{S}_k$ to control the error in the approximate functional value $J_k(\Omega_k + \mu_k \mathbf{V}_k)$ to the prescribed tolerance $\gamma \mu_k \|V_k\|^2_{\Gamma_k}$,

$$\big|J(\Omega_k + \mu_k \mathbf{V}_k) - J_k(\Omega_k + \mu_k \mathbf{V}_k))\big| \leq \gamma \mu_k \|V_k\|^2_{\Gamma_k}, \tag{9}$$

where $\gamma = \frac{1}{2} - \delta \geq \delta$ prevents excessive numerical resolution relative to the geometric one. This is achieved within the module APPROXJ via the *Dual Weighted Residual* method (DWR) [2], taylored to the approximation of the functional value J. The remaining modules perform the following tasks. The module SOLVE finds approximate solutions $U_k \in \mathbb{S}_k$ of (1) and $Z_k \in \mathbb{S}_k$ of an adjoint equation (necessary for the computation of $g(\Omega_k)$), while RIESZ builds on $\mathbb{S}_k$ an approximation G_k to the shape derivative g_k. Finally, the module LINESEARCH enforces an inexact version of (6).

Energy Decrease. The triangle inequality, in conjunction with conditions (8) and (9), yields

$$\left|J_k(\Omega_k+\mu_k\mathbf{V}_k)-J(\Omega_k+\mu_k\mathbf{v}_k)\right| \le \frac{1}{2}\mu_k\|V_k\|^2_{\Gamma_k}, \tag{10}$$

which is a bound on the local error incurred in at step k. On the other hand, the exact energy decrease reads

$$J(\Omega_k)-J(\Omega_k+\mu_k\mathrm{v}_k) \approx -\mu_k dJ(\Omega_k;\mathrm{v}_k)=\mu_k b_{\Gamma_k}(\mathrm{v}_k,\mathrm{v}_k)=\mu_k\|\mathrm{v}_k\|^2_{\Gamma_k} \ge (1-\theta)^2\mu_k\|\mathbf{V}_k\|^2_{\Gamma_k}, \tag{11}$$

and leads to the further constraint $(1-\theta)^2 > \frac{1}{2}$ to guarantee the energy decrease $J_k(\Omega_k+\mu_k\mathbf{V}_k) < J(\Omega_k)$.

If ASQP converges to a stationary point, i.e. $\mu_k\|V_k\|^2_{\Gamma_k} \to 0$ as $k\to\infty$, then the routines `DIRECTION` and `APPROXJ` approximate the descent direction V_k and functional $J(\Omega_k)$ increasingly better as $k\to\infty$, as dictated by (7) and (9). In other words, this imposes a dynamic error tolerance and progressive improvement in approximating U_k, Z_k and G_k as $k\to\infty$. This argument is a consistency check of ASQP.

We observe that the test (9) is not very demanding for DWR. So we expect coarse meshes at the beginning, and a combination of refinement and coarsening later as DWR detects geometric singularities, such as corners, and sorts out whether they are genuine to the problem or just due to lack of numerical resolution. This aspect of our approach is a novel paradigm in adaptivity and is documented in §3.

Prior Work. The idea of coupling FEM, a posteriori error estimators and optimal design error estimators to efficiently solve shape optimization problems is not new. The pioneering work [3] presents an iterative scheme, where the Zienkiewicz-Zhu error indicator and the L^2 norm of the shape gradient are both used at each iteration to improve the PDE error and the Geometric error, respectively. However, the algorithm in [3] does not resort to any dynamically changing tolerance, that would allow, as it happens for ASQP, to produce coarse meshes at the beginning of the iteration and a combination of Geometric and PDE refinement/coarsening later on. Moreover, [3] does not distinguish between fake and genuine geometric singularities that may arise on the domain boundary during the iteration process, and does not allow the former to disappear. More recently, the use of adaptive modules for the numerical approximation of PDEs has been employed by several authors [1, 12, 11] to improve the accuracy of the solution of shape optimization problems. However, in these papers the critical issue of linking the adaptive PDE approximation with an adaptive procedure for the numerical treatment of the domain geometry is absent. We address this linkage below.

2 Drag Minimization for Stokes Flow

Let $\Omega\subset\mathbb{R}^d$, $d\ge 2$ be a bounded domain of $\mathbb{R}^d$. Let $\mathbf{u}:=\mathbf{u}(\Omega)$ and $p:=p(\Omega)$ solve the Stokes problem:

$$-\mathrm{div}\mathbf{T}(\mathbf{u},p)=0, \qquad \mathrm{div}\,\mathbf{u}=0, \qquad \text{in } \Omega, \tag{12}$$

with Dirichlet boundary condition $\mathbf{u}=\mathbf{v}_\infty$ on Γ_{in}, $\mathbf{u}=\mathbf{0}$ on $\Gamma_s\cup\Gamma_w$, and traction-free boundary condition $\mathbf{T}(\mathbf{u},p)\cdot\mathbf{n}=0$ on Γ_{out} (see Figure 1). Hereafter, $\mathbf{T}(\mathbf{u},p):=2\nu\varepsilon(\mathbf{u})-p\mathbf{I}$ is the stress tensor with $\varepsilon(\mathbf{u})=\frac{\nabla\mathbf{u}+\nabla\mathbf{u}^T}{2}$, and $\mathbf{v}_\infty=V_\infty\hat{\mathbf{v}}_\infty$, with $\hat{\mathbf{v}}_\infty$ being the unit vector directed as the incoming flow and V_∞ a scalar function.

The drag exerted by the fluid on the obstacle surrounded by Γ_s is given by the functional

$$J(\Omega)=J(\Omega,\mathbf{u},p):=-\int_{\Gamma_s}\hat{\mathbf{v}}_\infty\cdot\mathbf{T}(\mathbf{u},p)\cdot\mathbf{n}\,d\Gamma. \tag{13}$$

We consider the following shape optimization problem $\min_{\Omega\in\mathscr{U}_{ad}} J(\Omega)$ on the set $\mathscr{U}_{ad}$ of admissible configurations with given volume, obtained by perturbing only the boundary Γ_s of the obstacle [9].

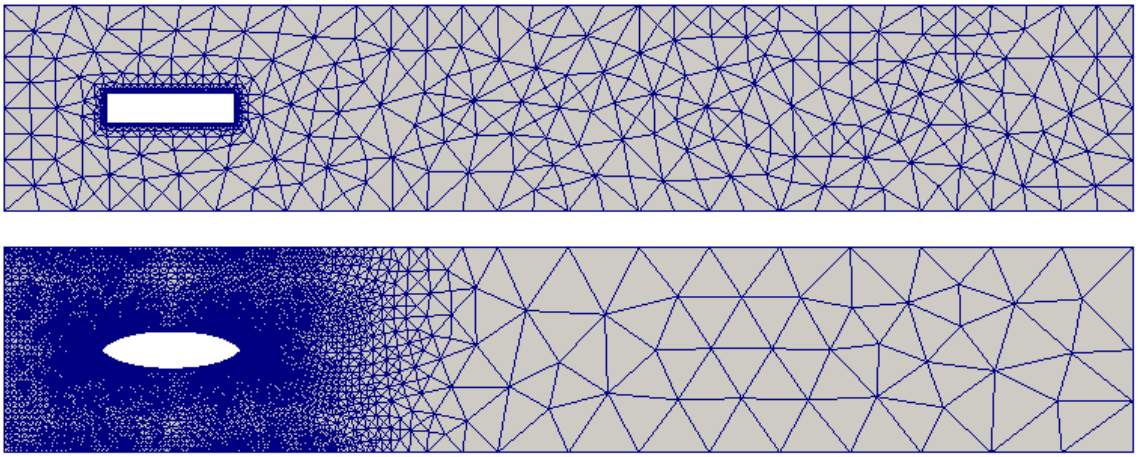

Fig. 1 Initial (top) and final (bottom) configuration: Γ_s is the deformable part of Ω, Γ_{in} the left-hand part, Γ_{out} the right-hand part and Γ_w the union of the upper and lower part. The algorithm obtains the optimal "rugby ball" shape [9]. The mesh refinement takes place mostly around Γ_s, whereas in the rest of Ω the mesh is rather coarse: this is related to DWR mesh refinement (and coarsening) and the particular expression (13) of the cost functional $J(\Omega)$.

It is possible to prove [7] that, for all sufficiently smooth vector fields $\mathbf{v}$ which are non-zero in a neighbourhood of Γ_s, the shape derivative of $J(\Omega)$ in the direction $\mathbf{v}$ is given by

$$dJ(\Omega;\mathbf{v}) = -2\nu \int_{\Gamma_s} \varepsilon(\mathbf{u}) : \varepsilon(\mathbf{z})\ \mathrm{v}\ d\Gamma, \tag{14}$$

with $\mathrm{v} = \mathbf{v}\cdot\mathbf{n}$ the normal velocity and $\mathbf{z}$ the solution to the adjoint problem

$$-\mathrm{div}\mathbf{T}(\mathbf{z},q) = 0, \qquad \mathrm{div}\ \mathbf{z} = 0, \qquad \text{in } \Omega, \tag{15}$$

subject to Dirichlet boundary conditions $\mathbf{z} = -\hat{\mathbf{v}}_\infty$ on Γ_s, $\mathbf{z} = \mathbf{0}$ on $\Gamma_w \cup \Gamma_{in}$, and traction-free condition $\mathbf{T}(\mathbf{z},q)\cdot\mathbf{n} = 0$ on Γ_{out}.

3 Numerical Experiment: Optimal Shape for Drag Minimization

In this section we briefly describe key aspects of the implementation of `ASQP` for the successful realization of simulations. A full description of the algorithm can be found in [7]. The implementation of `ASQP` was done using the toolbox ALBERTA [10], and the graphics were produced with ParaView [6].

Adaptivity. Adaptivity is carried out inside the modules `APPROXJ` and `DIRECTION`. In the module `APPROXJ`, adaptivity is performed using the goal-oriented Dual Weighted Residual estimator (DWR) driven

by approximation of the boundary functional $J(\Omega)$ [2]. Briefly, the goal-oriented DWR estimator determines where to refine/coarsen the mesh in Ω in order to improve the functional approximation, without imposing a small error in the global energy norm over the whole domain (see Figures 1 and 3).

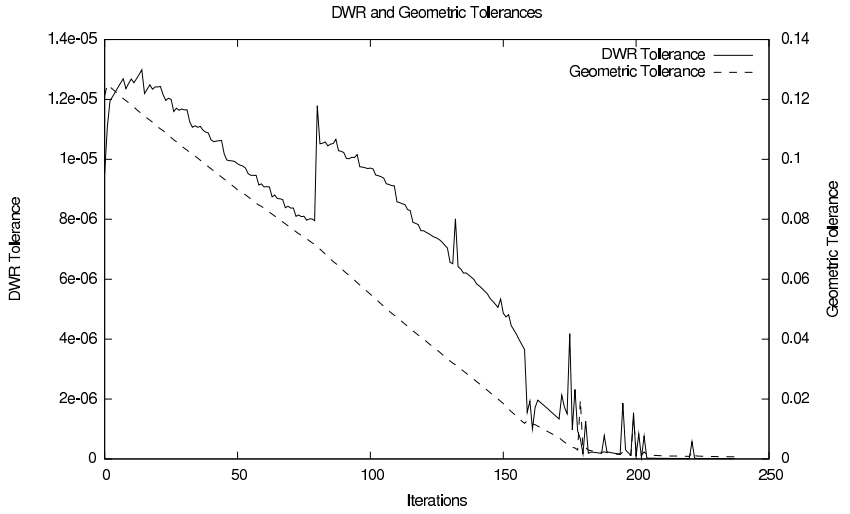

Fig. 2 Dynamic tolerance for both Geometric and PDE approximation: the adaptive SQP method produces coarse meshes at the beginning and a combination of Geometric and PDE refinement/coarsening later on (see Figure 3). The zig-zag behaviour in the tolerance is due to the combination of refinement/coarsening. Coarsening allows the tolerance to increase (see Table 3).

Iteration	0	1	81	88	128	150	153	160	161	163	173	175	177	179	181	189	196	200	202	204	213	222
DWR-ref	2000					218		523	2428	2112					7625			3786	4566	32372	1051	81657
DWR-coars	777							697	1284	1096	174	176	178	180	2312			1994	5355	3305		2234
LB-ref	75	44	4	21	5		25	523			29	65	56	819		191	35	1379		1002	0	924
LB-coars	88		22		11		14	49			44		4	54		104	11	57		113		1000

Table 1 Number of marked elements for refinement/coarsening according to Laplace-Beltrami (LB) and Dual Weighted Residual (DWR). The adaptive SQP method, with dynamically changing tolerance, alternates refinement/coarsening for LB and DWR. After the first two iterations, where refinement/coarsening takes place, the algorithm performs 80 iterations of optimization without changing the numerical resolution. Later on, the tolerance is modified by a sequence of DWR and LB refinement/coarsening.

The scalar velocity v_k obeys (5) with $\mathbb{V}(\Gamma_k) := H^1(\Gamma_k)$ and the bilinear form $b_{\Gamma_k}(\mathrm{v},\mathbf{w}) := \int_{\Gamma_k} \alpha_b \nabla_\Gamma v \cdot \nabla_\Gamma w + \beta_b v w$, where ∇_Γ denotes the surface gradient, and $\alpha_b = 10^{-3}$, $\beta_b = 1$. The module `DIRECTION` enforces the bound (7) on $\|V_k - \mathrm{v}_k\|_{\Gamma_k}$ using the a posteriori error estimators for the *Laplace-Beltrami* (LB) operator Δ_Γ developed in [8]. They are of residual type and estimate the energy error when solving $\Delta_\Gamma u = f$ on a known surface Γ. They consist of the usual PDE estimator and a new *geometric estimator* that accounts for the approximation of Γ by piecewise polynomials. Since Γ is unknown in this context, we mimic the W^1_∞ error between true and discrete surface by properly scaled jumps of the normal vector to the discrete surface. More precisely, the error indicator associated to element T of the k-th surface Γ_k is given by

$$\eta_{\Gamma_k}(T)^2 := h_T^2 \|\mathscr{R}(V_k)\|^2_{L^2(T)} + h_T \|\mathscr{J}(V_k)\|^2_{L^2(\partial T)} + \max_{S\subset\partial T} \mathscr{J}^2_{n,S} \|\nabla_\Gamma V_k\|^2_{L^2(T)},$$

where $\mathscr{R}(V_k) = -\alpha_b \Delta_\Gamma V_k + \beta_b V_k - g_k$ is the so-called *interior residual*, $\mathscr{J}(V_k)$ is the *jump residual*, namely jump of $\nabla_\Gamma V_k$ normal to the edge, and $\mathscr{J}_{n,S}$ is the *jump of the unit normal vector* (to the surface) across the interelement side S (see Figure 2 and Table 3).

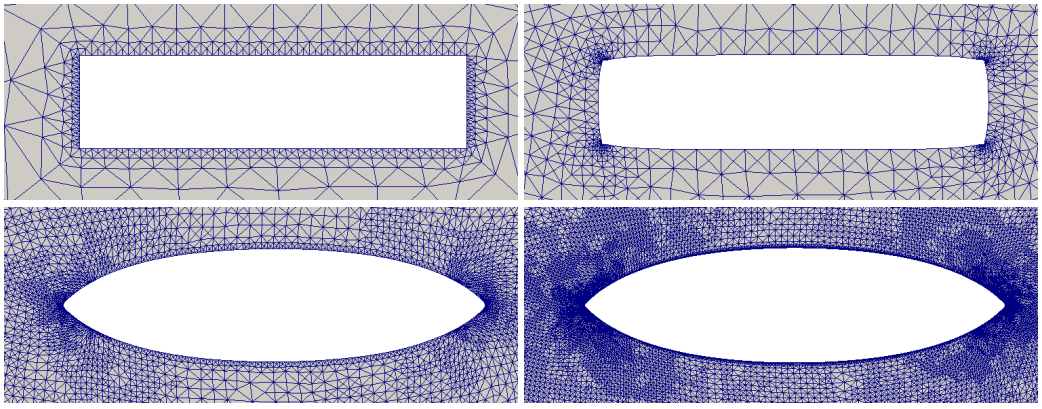

Fig. 3 Combination of DWR and LB refinement/coarsening. Evolution of the initial configuration Γ_s: iterations 0, 44, 184 and 217. The initially refined corners (top) are subsequently smoothed out and coarsened (see Figure 4). The new corners of the rugby ball, instead, are genuine singularities and are preserved and further refined by ASQP (bottom).

Geometrically Consistent Mesh Modification (GCMM). The presence of corners (or kinks) on the deformable boundary Γ_k is usually problematic. First, the scalar product $b_{\Gamma_k}(\cdot,\cdot)$ of (5) includes a LB regularization term ($\alpha_b > 0$) which stabilizes the boundary update but cannot remove kinks because V_k is smooth (see (14)). Second, DWR regards kinks as true singularities and tries to refine them accordingly. The combination of these two effects leads to numerical artifacts (ear formation) and halt of computations.

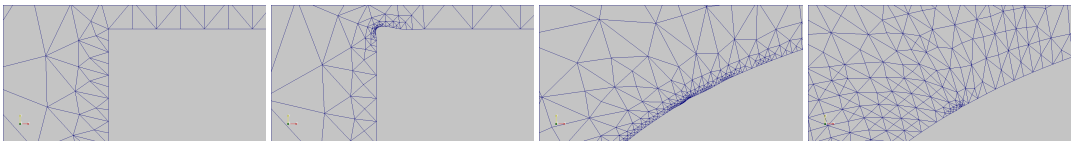

Fig. 4 Detection of genuine geometric singularities. Evolution of the initial upper-left corner of Γ_s (see top of Figures 1 and 3): snapshots of iterations 0, 1, 160 and 190. The adaptive SQP method is able to sort out whether geometric singularities are genuine to the problem or just due to lack of numerical resolution and to coarsen overrefined regions of the computational grid.

The GCMM method of [4] circumvents this issue; see Figure 4. Whenever the boundary mesh Γ_k is to be modified (refine, coarsen, or smooth out), then the discrete curvature H_k of Γ_k is interpolated and the new position X_k of the free boundary is determined from the fundamental geometric identity $-\Delta_{\Gamma_k} X_k = H_k$. This preserves geometric consistency, which is violated by simply interpolating Γ_k, as well as accuracy [4]. In addition, this computation rounds fake kinks (due to numerics) and preserves genuine kinks (see Figure 5).

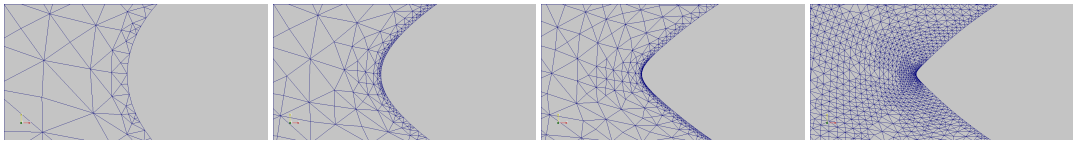

Fig. 5 Detection of genuine geometric singularities. Zoom on the evolution of the left-hand part of the initial configuration Γ_s (see top of Figure 1 and botton of Figure 3): snapshots of iterations 140,160,180 and 220. The adaptive SQP method is able to recognize the corner of the rugby ball as genuine singularity of the problem and to refine the mesh (combined use of LB and DWR error estimates) to improve both the PDE and the Geometric approximation.

Mesh Quality. The mesh is evolved by a prescribed discrete velocity of its boundary. To avoid mesh deterioration a mechanism to maintain good quality must be provided. Remeshing in each iteration is expensive and destroys the binary hierarchical data structure used for refinements and coarsenings [10]. Our approach is to use an optimization routine that works on stars and selectively reallocates the center node so as to improve the star quality and approximately preserve the local mesh size. It does not change the mesh topology so it is compatible with the binary data structure. In each star we minimize the SSU (Simultaneous Smoothing and Untangling) cost functional proposed in [14]. When optimization alone is not sufficient to maintain a good quality we remesh the domain. We refer to [7] for the effect of remeshing.

Time Step. Control of time step is required to satisfy the Armijo conditions (6) as well as to avoid node crossing when evolving the mesh [5]. The latter constraint sometimes dictates the time step, especially when the mesh is fine. We have found that remeshing ameliorates this issue upon drastically improving the mesh quality. In [7] we allow remeshing inside the Armijo condition.

Constraints. The area constraint that defines the class of admissible functions is enforced via a Lagrange multiplier. The algorithm, described in [5], guarantees volume conservation to machine precision in each time iteration and is well suited to be utilized inside the Armijo condition loop.

References

1. P. Alotto, P. Girdinio, P. Molfino and M. Nervi, *Mesh adaption and optimization techniques in magnet design*, IEEE Trans. Magnetics, 32, 4: 2954–2957, 1996.
2. W. Bangerth and R.Rannacher, *Adaptive Finite Element Methods for Differential Equations*, Birkhäuser, 2003
3. N.V. Banichuk, A. Falk and E. Stein, *Mesh refinement for shape optimization*, Structural Optim., 9:46–51, 1995.
4. A. Bonito and R.H. Nochetto and M.S. Pauletti, *Geometrically consistent mesh modification*, (submitted).
5. A. Bonito and R.H. Nochetto and M.S. Pauletti, *Parametric FEM for geometric biomembranes*, (submitted).
6. A. Henderson, *ParaView Guide, A Parallel Visualization Application* Kitware Inc., 2007.
7. P. Morin, R.H. Nochetto, M.S. Pauletti and M. Verani, *AFEM for shape optimization* (in preparation).
8. K. Mekchay, P. Morin, and R.H. Nochetto, *AFEM for Laplace Beltrami operator on graphs: Design and conditional contraction property* (submitted).
9. O. Pironneau, *On optimum profiles in Stokes flow*, J. Fluid Mech. 59: 117–128, 1973.
10. A. Schmidt and K.G. Siebert, *Design of Adaptive Finite Element Software. The Finite Element Toolbox ALBERTA*, Lecture Notes in Computational Science and Engineering 42, Springer, Berlin, 2005.
11. A. Schleupen, K. Maute and E. Ramm, *Adaptive FE-procedures in shape optimization*, Struct. Multidisc. Optim., 19: 282–302, 2000.
12. J.R. Roche, *Adaptive method for shape optimization*, 6th World Congresses of Structural and Multidisciplinary Optimization, Rio de Janeiro, 2005.
13. J. Sokołowski and J.-P. Zolésio, *Introduction to Shape Optimization*, Springer-Verlag, Berlin, 1992.
14. J. M. Escobar, E. Rodriguez, R. Montenegro, G. Montero and J. M. Gonzalez-Yuste, *Simultaneous untangling and smoothing of tetrahedral meshes*, Comput. Meths. Appl. Mech. Engn., 192 2775–2787, 2003.