Relaxed error control in shape optimization that utilizes remeshing

Daniel N. Wilke[1,*,†], Schalk Kok[2] and Albert A. Groenwold[3]

[1]*Department of Mechanical and Aeronautical Engineering, University of Pretoria, Pretoria, 0002, South Africa*
[2]*Modelling and Digital Science, Council for Scientific and Industrial Research, Scientia, Pretoria, 0001, South Africa*
[3]*Department of Mechanical and Mechatronic Engineering, University of Stellenbosch, Stellenbosch, 7602, South Africa*

SUMMARY

Shape optimization strategies based on error indicators usually require strict error control for every computed design during the optimization run. The strict error control serves two purposes. Firstly, it allows for the accurate computation of the structural response used to define the shape optimization problem itself. Secondly, it reduces the discretization error, which in turn reduces the size of the step discontinuities in the objective function that result from remeshing in the first place. These discontinuities may trap conventional optimization algorithms, which rely on both function and gradient evaluations, in local minima. This has the drawback that multiple analyses and error computations are often required per design to control the error.

In this study we propose a methodology that relaxes the requirements for strict error control for each design. Instead, we rather control the error as the iterations progress. Our approach only requires a single analysis and error computation per design. Consequently, large discontinuities may initially be accommodated; their intensities reduce as the iterations progress. To circumvent the difficulties associated with local minima due to remeshing, we rely on gradient-only optimization algorithms, which have previously been shown to be able to robustly overcome these discontinuities.

KEY WORDS: error indicator; remeshing; r-refinement; shape optimization; radial basis function; analytical sensitivities; gradient-only optimization; discontinuities

1. INTRODUCTION

Shape optimization may be considered a natural companion to *a posteriori* adaptive finite element (FE) mesh refinement, because both techniques share the computational burden of multiple analyses [1–4]. A large portion of the computational burden associated with *a posteriori* adaptive FE mesh refinement in once-off analyses is already accommodated for during shape optimization. *A posteriori* error indicators (or estimators) can be incorporated into FE based shape optimization environments using two distinctly different approaches [1]—the difference being whether changes in mesh *topologies* are allowed as the iterations progress, or not.

Approaches that require the mesh topology to remain fixed between optimization iterations only update the mesh topology after the optimization update, as depicted in Figure 1(a). Hence, a single finite element analysis (FEA) is required for each candidate shape design if the error remains sufficiently small after a shape design update. Otherwise, multiple analyses are required per candidate shape design to reduce the discretization error. Because the function values computed with different mesh topologies differ for the same candidate shape design, the optimization update may

*Correspondence to: Daniel N. Wilke, Department of Mechanical and Aeronautical Engineering, University of Pretoria, Pretoria, 0002, South Africa.
†E-mail: nico.wilke@up.ac.za

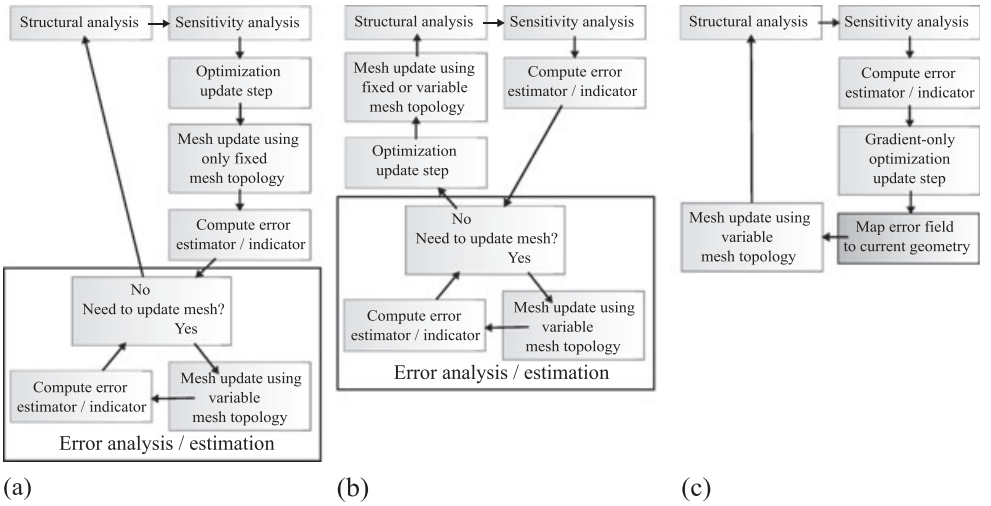

Figure 1. Finite element-error indicator integration into optimization.

have to be repeated with the updated mesh topology [1]. This strategy may be efficient when design changes are small and no error control is required between design updates.

Alternatively, the mesh topology may be updated to control the discretization error during each optimization update, as depicted in Figure 1(b). Multiple FEAs may be required for each candidate shape design to control the discretization error, that is, to limit the size of the jump discontinuities resulting from different mesh topologies, allowing for the efficient use of classical gradient based algorithms [1].

In this study, we implement a new approach that allows for relaxation of the *requirement* that strict error control is done when using classical optimization algorithms that rely on both function and gradient information. This new approach is depicted in Figure 1(c). We are able to relax the requirement for strict discretization error control for each candidate shape design, because gradient-only optimization allows us to robustly and efficiently optimize the resulting discontinuous cost functions, as we have previously demonstrated [5] (but at that time without any error indicators whatsoever).

Instead of obtaining a converged error for each candidate shape design, we allow the error to converge as the iterations progress. A loosely related strategy was implemented by Van Keulen *et al.* [6]. The benefit of our proposed approach is enhanced computational efficiency. Note, however, that the use of gradient-only optimization is an essential ingredient to enable relaxed error control. Conventional gradient based algorithms are not able to solve these optimization problems robustly [5,7], unless the magnitudes of the discontinuities are controlled to be sufficiently small by enforcing strict error control. This is so because classical methods rely on function value information, which contain numerical local minima in the presence of remeshing.

We demonstrate our approach by extending the previously proposed uniform remeshing shape optimization strategy [7] on the basis of the efforts of Persson and Strang [8]. We extend the strategy by including the Zienkiewicz–Zhu (ZZ) error indicator [9], with the objective of improving the accuracy of the global energy norm of the structure for a fixed number of DOFs. This ensures that the computational cost for each design iteration remains more or less the same. In addition, (semi) analytical sensitivities are made available for use in gradient-only optimization approaches. We can indeed incorporate error indicators freely, because we are able to robustly and efficiently optimize the discontinuous objective functions resulting from changes in the mesh topology. Although we choose to keep the number of DOFs constant, the technique proposed herein is generic and can be combined with any error estimator to control errors within user specified tolerances when allowing for a varying number of DOFs.

In summary, the contribution of our proposed strategy is that it requires only a single FEA for each candidate shape design. This is achieved by mapping the computed error indicator of a

given shape geometry to the geometry obtained after an iteration of our gradient-only optimization algorithm. The mapping of the error indicator field between two shape geometries is merely a relocation of the nodal positions of the error indicator mesh from one shape geometry to the next using radial basis functions (RBFs), as opposed to a linearization of the error indicator field between two shape geometries [10]. This allows for a generic mapping of data between two geometries and may be used to map additional or alternative information as opposed to the error indicator mapped in this study. An additional advantage of this mapping is that the required number of computations are fewer than that required to compute a linearized error indicator field, or the actual error indicator field, which would require a full FEA.

We do not address convergence herein in a rigorous fashion. In engineering optimization, this is not uncommon. Many efficient algorithms for engineering optimization simply rely on the accuracy of the approximations used to drive the iteration sequence to termination; this is so because there often is a computational penalty to be paid when convergence is enforced using, for example, conservatism, or trust region methods [11]. In turn, the motivation for this is that the simulations used in engineering optimization, for example, FE or CFD simulations are expensive enough as they are. Popular algorithms in this class include the convex linearization method (CONLIN) of Fleury and Braibant [12] and its generalization, the method of moving asymptotes of Svanberg [13]. Having said this, the use of unconstrained gradient-only optimization methods in certain classes of piecewise discontinuous functions have been proven to be convergent [14]. What is more, currently, we at least have empirical evidence from our previous studies [5, 14] with a uniform remeshing strategy that our approach seems promising. Nevertheless, an investigation into the convergence properties of the methodologies we use herein is deserving of much attention.

Before proceeding, some clarification regarding the computation of gradients is provided. Given a candidate shape design, this shape is discretized using some meshing algorithm. A structural analysis is then performed, as well as a (semi) analytical sensitivity analysis. The sensitivity analysis is performed while disallowing any mesh topology changes. This computation can be performed for *every imaginable* shape design. We denote the result of the sensitivity analysis as the gradient of the objective function. It is this gradient that has to be provided to gradient-only optimization algorithms [5, 14]. This usage of the term gradient is *only* inconsistent with the strict mathematical definition if the computation occurred *exactly* at the threshold of a change in mesh topology. Because it is not necessary to identify such a case when using gradient-only optimization algorithms, our definition of the gradient suffices for the purposes considered herein.

Our paper is arranged as follows. Firstly, we present the gradient-only shape optimization problem in Section 2. Thereafter, we briefly outline the gradient-only optimization algorithm used in this study in Section 3. We then discuss the structural analysis, including the *a posteriori* error indicator and mesh refinement strategy, in Section 4. Our adaptive mesh generation strategy is presented in Section 5, followed in Section 6 by a sensitivity analysis. Section 7 contains the numerical results, which includes a convergence study and two example problems. Additional examples may be found in Wilke [15]. Some conclusions are offered in Section 8.

2. SHAPE OPTIMIZATION PROBLEM

The problem under consideration is the equality constrained shape optimization problem, for which the Lagrangian is given by

$$L(\boldsymbol{x}, \boldsymbol{\lambda}) = \mathcal{F}(\Omega(\boldsymbol{x})) + \sum_{j=1}^{m} \lambda_j g_j(\boldsymbol{x}),\ \boldsymbol{x} \in X \subseteq \mathbb{R}^n \text{ and } \boldsymbol{\lambda} \in \mathbb{R}^m, \tag{1}$$

where the objective function $\mathcal{F}(\Omega(\boldsymbol{x}))$ is a scalar function that depends on the geometry Ω of the structure, which in turn depends on the control variables $\boldsymbol{x}$ that describe the geometrical boundary $\partial\Omega$. The equality constraints $g_j(\boldsymbol{x}) = 0,\ j = 1, 2, \cdots, m$ are scalar functions of the control variables $\boldsymbol{x}$. For the sake of brevity, the cost function and the constraints will respectively be denoted by $\mathcal{F}(\boldsymbol{x})$ and $\boldsymbol{g}(\boldsymbol{x})$; this notation will however imply dependency on $\Omega(\boldsymbol{x})$. We choose to represent the

geometrical boundary $\partial\Omega$ by a simple piecewise linear interpolation between the control variables. However, Bezier curves or B-splines, and others, may of course also be used.

Normally, the saddle point of (1) is solved for using the dual formulation

$$\max_{\boldsymbol{\lambda}} \left\{ \min_{\boldsymbol{x}} L(\boldsymbol{x}, \boldsymbol{\lambda}) \right\}. \tag{2}$$

We however solve (1) using the gradient-only dual formulation [5]

$$\max_{\boldsymbol{\lambda}}^{g} \left\{ \min_{\boldsymbol{x}}^{g} L(\boldsymbol{x}, \boldsymbol{\lambda}) \right\}, \tag{3}$$

with $\max_{\boldsymbol{\lambda}}^{g}$ defined as follows: find $\boldsymbol{\lambda}$, such that

$$\nabla_{\boldsymbol{\lambda}}^{\mathrm{T}} L(\boldsymbol{x}, \boldsymbol{\lambda} + \gamma_v \boldsymbol{v}) \boldsymbol{v} \leqslant 0, \ \forall \ \boldsymbol{v} \in \mathbb{R}^m. \tag{4}$$

Similarly, $\min_{\boldsymbol{x}}^{g}$ is defined as follows: find $\boldsymbol{x}$, such that

$$\nabla_{\boldsymbol{x}}^{\mathrm{T}} L(\boldsymbol{x} + \delta_u \boldsymbol{u}, \boldsymbol{\lambda}) \boldsymbol{u} \geqslant 0, \forall \ \boldsymbol{u} \in \mathbb{R}^n \text{ such that } \boldsymbol{x} + \delta_u \boldsymbol{u} \in X, \tag{5}$$

with X as the convex set of all possible solutions, $\nabla_{\boldsymbol{x}}$ the partial derivatives w.r.t. $\boldsymbol{x}$, $\nabla_{\boldsymbol{\lambda}}$ the partial derivatives w.r.t. $\boldsymbol{\lambda}$, and where δ_u and γ_v are real positive scalars. Note that we have only exploited gradient information of $L(\boldsymbol{x}, \boldsymbol{\lambda})$.

Following a similar approach, we may pose the problem as an inequality constrained problem by imposing positive bounds on the Lagrangian multipliers. However, because our proposed approach is currently limited to *smooth* constraint functions, we currently cannot consider displacements, strains and stresses as constraints, but we hope to do so in the near future.

3. OPTIMIZATION ALGORITHM

We will use the gradient-only sequential spherical approximation algorithm presented by Wilke *et al.* [5] to optimize the discontinuous shape optimization problem. For the sake of completeness and brevity, we merely outline the algorithm here (for details on the algorithm, and a motivation for using gradient-only optimization methods in the first place, the reader is referred to [5]):

1. *Initialization*: Select real constants $\epsilon > 0$, $\alpha > 1$, initial curvature $c^{\{0\}} > 0$ and initial point $\left[\boldsymbol{x}^{\{0\}} \ \ \boldsymbol{\lambda}^{\{0\}} \right]$. Set $k := 1$, $s := 0$.
2. *Gradient evaluation*: Compute $\nabla_{\boldsymbol{x}}^{\mathrm{T}} L \left(\left[\boldsymbol{x}^{\{k\}} \ \ \boldsymbol{\lambda}^{\{k\}} \right] \right)$.
3. *Approximate optimization*: Construct the local gradient-only approximate subproblem

$$\nabla \tilde{f}^{\{k\}}(\boldsymbol{x}) = \nabla_{\boldsymbol{x}}^{\mathrm{T}} L \left(\left[\boldsymbol{x}^{\{k\}} \ \ \boldsymbol{\lambda}^{\{k\}} \right] \right) + \boldsymbol{H}^{\{k\}} \left(\boldsymbol{x} - \boldsymbol{x}^{\{k\}} \right) \tag{6}$$

   at $\boldsymbol{x}^{\{k\}}$, using $\boldsymbol{H}^{\{k\}} = c^{\{k\}} \boldsymbol{I}$, where $\boldsymbol{I}$ is the identity matrix and

$$c^{\{k\}} = \frac{\left(\boldsymbol{x}^{\{k-1\}} - \boldsymbol{x}^{\{k\}} \right)^{\mathrm{T}} \left(\nabla_{\boldsymbol{x}}^{\mathrm{T}} L \left(\left[\boldsymbol{x}^{\{k-1\}} \ \ \boldsymbol{\lambda}^{\{k\}} \right] \right) - \nabla_{\boldsymbol{x}}^{\mathrm{T}} L \left(\left[\boldsymbol{x}^{\{k\}} \ \ \boldsymbol{\lambda}^{\{k\}} \right] \right) \right)}{\left(\boldsymbol{x}^{\{k-1\}} - \boldsymbol{x}^{\{k\}} \right)^{\mathrm{T}} \left(\boldsymbol{x}^{\{k-1\}} - \boldsymbol{x}^{\{k\}} \right)}. \tag{7}$$

   (In an inner loop, use $c^{\{k\}}$ as calculated in step 6(b)). Solve this subproblem analytically, to arrive at $\boldsymbol{x}^{\{k*\}}$ by setting (6) equal to $\boldsymbol{0}$.
4. *Evaluation*: Compute $\nabla_{\boldsymbol{x}}^{\mathrm{T}} L \left(\left[\boldsymbol{x}^{\{k*\}} \ \ \boldsymbol{\lambda}^{\{k\}} \right] \right)$.

5. *Test if* $\boldsymbol{x}^{\{k*\}}$ *is acceptable*: if

$$\nabla_{\boldsymbol{x}}^{\mathrm{T}} L\left(\left[\boldsymbol{x}^{\{k*\}}\ \boldsymbol{\lambda}^{\{k\}}\right]\right)\left(\boldsymbol{x}^{\{k*\}}-\boldsymbol{x}^{\{k\}}\right) \leqslant \nabla^{\mathrm{T}} \tilde{f}\left(\boldsymbol{x}^{\{k*\}}\right)\left(\boldsymbol{x}^{\{k*\}}-\boldsymbol{x}^{\{k\}}\right)=0 \tag{8}$$

   go to step 7.
6. *Initiate an inner loop to effect conservatism*:
   (a) Set $s := s+1$.
   (b) Set $c^{\{k\}} := \alpha c^{\{k\}}$.
   (c) Go to step 3.
7. *Move to the new iterate*: Set $\boldsymbol{x}^{\{k+1\}} := \boldsymbol{x}^{\{k*\}}$.
8. *Update multiplier*: Set $\boldsymbol{\lambda}^{\{k+1\}} := \boldsymbol{\lambda}^{\{k\}} + \boldsymbol{\lambda_s}^{\{k+1\}}$, with $\boldsymbol{\lambda_s}^{\{k+1\}}$ the multiplier update step.
9. *Convergence test*: if $\left\|\left[\Delta \boldsymbol{x}^{\{k\}}\ \Delta\lambda^{\{k\}}\right]\right\| < \epsilon$, OR $\left\|\Delta \boldsymbol{x}^{\{i\}}\right\| < \epsilon,\ \forall\ i = \{k-4, k-3, \ldots, k\}$, OR $k = k_{\max}$, stop[‡].
10. *Initiate an additional outer loop*: Set $k := k+1$ and go to step 2.

We elaborate as follows: in step 5, a step is deemed acceptable when the projection of the step onto the directional derivative is negative, but this need not be optimal; superior strategies may well exist. Again, we hope to investigate this in the near future.

4. STRUCTURAL ANALYSIS

In shape optimization, the cost function $\mathcal{F}(\boldsymbol{x}) = \mathcal{F}(\boldsymbol{u}(\boldsymbol{\mathcal{X}}(\boldsymbol{x})))$ is an explicit function of the nodal displacements $\boldsymbol{u}$, which in turn is a function of the discretized geometrical domain $\boldsymbol{\mathcal{X}}$. The discretized geometrical domain $\boldsymbol{\mathcal{X}}$ is described by the control variables $\boldsymbol{x}$, which represent the geometrical boundary $\partial\Omega$.

The nodal displacements $\boldsymbol{u}$ are obtained by solving the approximate FE equilibrium equations for linear elasticity

$$\boldsymbol{K}\boldsymbol{u} = \boldsymbol{f}, \tag{9}$$

where $\boldsymbol{K}$ represents the assembled structural stiffness matrix and $\boldsymbol{f}$ the consistent structural nodal loads, from which the unknown displacements $\boldsymbol{u}$ can be computed. From $\boldsymbol{u}$, we can then locally compute elemental stress fields

$$\hat{\boldsymbol{\sigma}}_e = \boldsymbol{C}\,\boldsymbol{B}_e \boldsymbol{u}_e, \tag{10}$$

with constitutive relationship $\boldsymbol{C}$, element kinematic relation $\boldsymbol{B}_e$ and element displacement $\boldsymbol{u}_e$. By combining the local stress fields $\hat{\boldsymbol{\sigma}}_e$ of adjacent elements, we obtain a global discontinuous stress field $\hat{\boldsymbol{\sigma}}$ over the entire structure, because inter-elemental stress continuity is not enforced. As the true stress field is continuous, error indicators may be recovered from the discontinuous stress field [16].

As said, shape optimization and *a posteriori* adaptive FE mesh refinement may naturally compliment each other, because both imply multiple FEAs. Instead of only conducting a FEA for each candidate shape design, we also recover error indicators from these FEAs. The recovered error from a given shape design is then used to discretize an updated shape design, causing the refinement strategy to converge as the shape design converges.

4.1. Recovery-based global error indicator and refinement procedure

Although many recovery-based error indicators exist, which range from so-called global to local indicators [17], we will herein opt for only the well-known ZZ global error indicator [9]. We do so merely for the sake of brevity and simplicity—other indicators, be it global or local, may also be used.

Using the computed error, we seek a refinement strategy to indicate spatial refinement (respectively de-refinement) of the mesh. For this, we modify the refinement procedure of Zienkiewicz and

[‡]The notation used is $\Delta\boldsymbol{\phi}^{\{k\}} = \boldsymbol{\phi}^{\{k+1\}} - \boldsymbol{\phi}^{\{k\}}$.

Zhu [9] to suit our r-refinement strategy: for the ith element at the kth iteration, we compute the refinement ratio $\xi_i^{\{k\}}$ as

$$\xi_i^{\{k\}} = \frac{\left\| \boldsymbol{e}_i^{\{k\}} \right\|}{\bar{e}^{\{k\}}}, \tag{11}$$

where $\left\| \boldsymbol{e}_i^{\{k\}} \right\|$ represents the approximated energy norm and $\bar{e}^k$ the average element error. In turn, we use the refinement ratio $\xi_i^{\{k\}}$ to compute the ideal element length[§]

$$\hat{h}_i^{\{k\}} = \frac{h_i^{\{k-1\}}}{\left(\xi_i^{\{k\}}\right)^{1/p}}, \tag{12}$$

with $h^{\{0\}}$ chosen as the ideal element length of a uniform mesh for the first iteration. This also defines the initial number of nodes for our r-refinement strategy. Here, p is usually selected as the polynomial order of the shape functions away from singularities, and adjusted near singularities [9]. Our mesh generator naturally generates linear strain triangle elements, for which we found experimentally that $p < 4$ may result in oscillatory behavior. Hence, we have somewhat arbitrarily selected $p = 5$ herein [6]. We then smooth the discrete elemental scalar field $\hat{h}^{\{k\}}$ using nodal averaging to obtain a piecewise continuous ideal element field $\tilde{h}^{\{k\}}$ described by a FE interpolation. Finally, we normalize the continuous ideal element field $\tilde{h}^{\{k\}}$ to obtain

$$\check{h}^{\{k\}} = \frac{\tilde{h}^{\{k\}}}{\bar{h}^{\{k\}}} \kappa h^{\{0\}}, \tag{13}$$

with $\bar{h}^{\{k\}}$ the average continuous ideal element length and κ a scaling factor. We select κ as constant, but allowing κ to vary (as some function of the initial area, current area, number of initial boundary nodes and current boundary) may well be beneficial. Finally, we limit the minimum ideal element length $\check{h}^{\{k\}}$ as follows:

$$\check{h}^{\{k\}} = \max\left(h_{\min}, \check{h}^{\{k\}}\right), \tag{14}$$

with $h_{\min}$ prescribed.

5. ADAPTIVE MESH GENERATOR

Our mesh generator solves for the equilibrium position of a truss structure [8], which doubles as the FE mesh, using the ideal element length field $h^{\{k\}}, k = 1, 2, 3, \ldots$ as the unloaded truss lengths, as opposed to directly optimizing the mesh according to some optimality criterion [18, 19]. It has been demonstrated [5, 7, 8] that this approach generates 'good' meshes.

We start with an initial uniform mesh $\boldsymbol{\mathcal{X}}^{\{0\}}$ at $k = 0$, using a uniform ideal element length field $h^{\{0\}}$ [7]. After each analysis, we compute the ideal element length field $\check{h}^{\{k\}}$ using the refinement strategy described in Section 4.1. Recall that we then merely relocate the nodal positions of the computed ideal element length field to the new candidate shape design obtained from the optimization step, to avoid multiple analyses per candidate shape design, as illustrated in Figure 1(c). (We will describe the details of the mapping of the error field in Section 5.2.)

As with our previous mesh generator [7], we partition the mesh $\boldsymbol{\mathcal{X}}^{\{k\}}$ along the interior nodes $\boldsymbol{\mathcal{X}}^{\{k\}\Omega}$ and boundary nodes $\boldsymbol{\mathcal{X}}^{\{k\}\partial\Omega}$, which allows for the independent treatment of the boundary nodes $\partial\Omega^{\{k\}}$ and interior nodes $\Omega^{\{k\}}$. Superscript k denotes the iteration counter, which we will omit for the sake of brevity, unless we explicitly want to highlight the dependency on k.

[§]Ideal element length refers to the 'unloaded' truss lengths of the truss members in our truss structure analogous mesh generator [7]—also see Section 5.

The boundary nodes $\boldsymbol{\mathcal{X}}^{\partial\Omega}$ are seeded according to the ideal element length field $\breve{h}^{\{k\}}$ along the geometrical boundary $\partial\Omega$, with nodes explicitly placed on the control variable locations $\boldsymbol{x}$. This ensures accurate representation of the defined geometrical domain Ω. Therefore, $\boldsymbol{x} \subset \boldsymbol{\mathcal{X}}^{\partial\Omega}$, with $\partial\Omega$ described by a piecewise linear interpolation of $\boldsymbol{x}$. The boundary nodes $\boldsymbol{\mathcal{X}}^{\partial\Omega}$ remain fixed during the current iteration of the mesh generation process. We therefore *only* solve for $\boldsymbol{\mathcal{X}}^{\Omega}$ in finding the equilibrium of the truss structure

$$\boldsymbol{F}_{\Omega}\left(F(\boldsymbol{\mathcal{X}}^{\Omega})\right) = \boldsymbol{0}. \tag{15}$$

The equilibrium of the truss structure is related to the interior nodes $\boldsymbol{\mathcal{X}}^{\Omega}$ via the force function

$$F(\boldsymbol{\mathcal{X}}^{\Omega}) = F\left(\boldsymbol{l}(\boldsymbol{\mathcal{X}}^{\Omega}), \boldsymbol{l}_0\left(\breve{h}^{\{k\}}, \boldsymbol{\mathcal{X}}^{\Omega}\right)\right) = \mathcal{K}(\boldsymbol{l}_0 - \boldsymbol{l}), \tag{16}$$

which depends on the constant spring stiffness $\mathcal{K}$, the length of the truss members $\boldsymbol{l}(\boldsymbol{\mathcal{X}}^{\Omega})$ and the undeformed truss lengths $\boldsymbol{l}_0\left(\breve{h}^{\{k\}}, \boldsymbol{\mathcal{X}}^{\Omega}\right)$. The undeformed truss lengths $\boldsymbol{l}_0\left(\breve{h}^{\{k\}}, \boldsymbol{\mathcal{X}}^{\Omega}\right)$ depend on the ideal element length field $\breve{h}^{\{k\}}$, as well as the interior nodes $\boldsymbol{\mathcal{X}}^{\Omega}$, because the ideal element length field $\breve{h}^{\{k\}}$ is evaluated at the midpoint of each truss member.

However, the ideal element length field $\breve{h}^{\{k\}}(\boldsymbol{e})$ is taken as a constant background field [6, 10, 20, 21] during the mesh generation process, that is, we do not recompute the error field or linearize the error field when the interior nodes $\boldsymbol{\mathcal{X}}^{\Omega}$ of the mesh vary. Consequently, the dependence of the ideal element length field $\breve{h}^{\{k\}}(\boldsymbol{e})$ on the spatially varying *a posteriori* error field $\boldsymbol{e}$ is constant and the sensitivity zero. Lastly, the displacement field $\boldsymbol{u}(\boldsymbol{\mathcal{X}}^{\Omega})$ depends on the interior nodes $\boldsymbol{\mathcal{X}}^{\Omega}$.

The reduced truss system in (15) is solved directly via the quadratically convergent Newton's method, which is given by

$$\frac{\partial \boldsymbol{F}_{\Omega}}{\partial \boldsymbol{\mathcal{X}}^{\Omega}} \Delta \boldsymbol{\mathcal{X}}^{\Omega} = -\boldsymbol{F}_{\Omega}. \tag{17}$$

The update of the nodal coordinates is given by

$$\boldsymbol{\mathcal{X}}^{\Omega}_{n+1} = \boldsymbol{\mathcal{X}}^{\Omega}_{n} + \Delta \boldsymbol{\mathcal{X}}^{\Omega}, \tag{18}$$

and for a constant ideal element length background field, the consistent tangent $\frac{\partial \boldsymbol{F}_{\Omega}}{\partial \boldsymbol{\mathcal{X}}^{\Omega}}$ is given by

$$\frac{\partial \boldsymbol{F}_{\Omega}}{\partial \boldsymbol{\mathcal{X}}^{\Omega}} = \frac{\partial \boldsymbol{F}_{\Omega}}{\partial \boldsymbol{l}} \frac{\partial \boldsymbol{l}}{\partial \boldsymbol{\mathcal{X}}^{\Omega}} + \frac{\partial \boldsymbol{F}_{\Omega}}{\partial \boldsymbol{l}_0} \frac{\partial \boldsymbol{l}_0}{\partial \boldsymbol{\mathcal{X}}^{\Omega}}. \tag{19}$$

We obtain quadratic convergence using Newton's method.

5.1. Boundary nodes

A first approach to accommodate a spatially varying ideal element length field may be to merely change the ideal element length and boundary spacing, whereas the number of boundary nodes and interior nodes follow from our previous uniform remeshing strategy [7]. However, limited improvements are achieved using this naive strategy.

In this study, we let the number of boundary nodes follow from the spatially varying ideal element length field by first placing nodes along the boundary $\boldsymbol{\mathcal{X}}^{\partial\Omega}$. We first compute the required number of boundary nodes per piecewise linear boundary section by dividing the physical length of each section by the ideal length of each section. To obtain an integer value for the required number of nodes per section, we round the result down. For each section, the boundary nodes are then explicitly placed such that the integral of the error between the boundary nodes are the same. Because we aim to keep the number of nodes constant, the remaining nodes are seeded in the interior domain $\boldsymbol{\mathcal{X}}^{\Omega}$. Although this strategy may seem simplistic, we found other strategies to determine the number

of boundary nodes, for example, using ratios of the circumference to interior area together with the error along the circumference to the interior, susceptible to oscillations.

As stated previously, the boundary nodes remain fixed once they have been placed along the boundary, which reduces the size of the Newton system when solving for the truss equilibrium. Consequently, an increase in the number of boundary nodes $\boldsymbol{\mathcal{X}}^{\partial\Omega}$ results in a decrease in the number of interior nodes $\boldsymbol{\mathcal{X}}^{\Omega}$ and vice versa, because we want to keep the total number of nodes fixed. To reduce the computational effort, we use the nodes that describe the ideal element length field $\check{h}^{\{k\}}$ as an initial guess for our interior nodes. To keep the total number of nodes constant, we need to either remove or add nodes to those nodes used in representing $\check{h}^{\{k\}}$. We do so by ranking the nodes and elements according to their error densities. We remove nodes by starting with nodes with smaller error densities, and add nodes by introducing nodes at the centroids of elements with higher element error densities.

5.2. Mapping the error field between candidate shape designs

The ideal element length field $\check{h}^{\{k\}}$ for the kth iteration is obtained by mapping the computed ideal element length field after the analysis of the $(k-1)$th iteration to the candidate shape design for the kth iteration. We achieve this by merely mapping the nodal positions, without requiring connectivity information, using RBFs. We therefore have total flexibility on whether we want to use or discard the nodal connectivity of the previous geometry. In this study, we pay the computational penalty of retriangulation at every iteration to keep our strategy unsophisticated, while allowing for large shape changes.

The details of the mapping strategy using RBFs are outlined in the Appendix.

6. SENSITIVITY ANALYSIS

Recall that the cost function $\mathcal{F}(\boldsymbol{u}(\boldsymbol{\mathcal{X}}(\boldsymbol{x})))$ is an explicit function of the nodal displacements. Specifically, in all the examples herein, the cost function $\mathcal{F}$ is the nodal displacement at the point where a point load F is applied, expressed as

$$\mathcal{F}(\boldsymbol{u}(\boldsymbol{x})) = \boldsymbol{u}_{\mathrm{F}}(\boldsymbol{x}). \tag{20}$$

The displacement field $\boldsymbol{u}(\boldsymbol{x})$ depends on the discretized geometrical domain $\boldsymbol{\mathcal{X}}$, which is obtained by solving for the nodal positions of a truss structure at equilibrium. The ideal element lengths $\check{h}^{\{k\}}$ of the truss structure are obtained from the error indicator discussed in Section 4. Then, the sensitivity of the displacement $\boldsymbol{u}_{\mathrm{F}}$ w.r.t. the control variables $\boldsymbol{x}$ is obtained by computing

$$\frac{\mathrm{d}\boldsymbol{u}_{\mathrm{F}}}{\mathrm{d}\boldsymbol{x}} = \frac{\mathrm{d}\boldsymbol{u}_{\mathrm{F}}}{\mathrm{d}\boldsymbol{\mathcal{X}}}\frac{\mathrm{d}\boldsymbol{\mathcal{X}}}{\mathrm{d}\boldsymbol{x}}. \tag{21}$$

The computation of $\frac{d\boldsymbol{u}_{\mathrm{F}}}{d\boldsymbol{\mathcal{X}}}$ is obtained by direct differentiation of the FE equilibrium equations $\boldsymbol{K}\boldsymbol{u} = \boldsymbol{f}$, given by

$$\frac{\mathrm{d}\boldsymbol{K}}{\mathrm{d}\boldsymbol{\mathcal{X}}}\boldsymbol{u} + \boldsymbol{K}\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}\boldsymbol{\mathcal{X}}} = \frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{\mathcal{X}}}. \tag{22}$$

For the constant external loads $\boldsymbol{f}$ we restrict ourselves to in this study, $\frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{\mathcal{X}}} = \boldsymbol{0}$ and (22) reduces to

$$\boldsymbol{K}\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}\boldsymbol{\mathcal{X}}} = -\frac{\mathrm{d}\boldsymbol{K}}{\mathrm{d}\boldsymbol{\mathcal{X}}}\boldsymbol{u}. \tag{23}$$

We compute $\frac{\mathrm{d}\boldsymbol{K}}{\mathrm{d}\boldsymbol{\mathcal{X}}}$ by direct differentiation of the analytical stiffness matrices for the linear strain triangle elements [7, 22], which allows to solve for $\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}\boldsymbol{\mathcal{X}}}$, to then obtain $\frac{\mathrm{d}\boldsymbol{u}_{\mathrm{F}}}{\mathrm{d}\boldsymbol{\mathcal{X}}}$.

The sensitivities of the nodal coordinates $\boldsymbol{\mathcal{X}}$ w.r.t. the control variables $\boldsymbol{x}$, $\frac{\mathrm{d}\boldsymbol{\mathcal{X}}}{\mathrm{d}\boldsymbol{x}}$, are obtained by differentiating the truss structure equilibrium equations from the mesh generation in Section 5. Recall that we partitioned $\boldsymbol{\mathcal{X}}$ along the interior nodes $\boldsymbol{\mathcal{X}}^{\Omega}$ and boundary nodes $\boldsymbol{\mathcal{X}}^{\partial\Omega}$. Also, recollect that

the boundary nodes $\boldsymbol{\mathcal{X}}^{\partial\Omega}$ are seeded according to the ideal element length field $\breve{h}^{\{k\}}$ along the geometrical boundary $\partial\Omega$, and that they remain fixed during the mesh generation process. Hence, the equilibrium of the truss structure $\boldsymbol{F}_\Omega$ only depends implicitly on the interior nodes $\boldsymbol{\mathcal{X}}^{\Omega}$.

Consider the dependency of the equilibrium equations $\boldsymbol{F}_\Omega$ on $\boldsymbol{x}$:

$$\boldsymbol{F}_\Omega\left(\boldsymbol{l}(\boldsymbol{\mathcal{X}}^{\Omega}(\boldsymbol{x}),\boldsymbol{\mathcal{X}}^{\partial\Omega}(\breve{h}^{\{k\}}(\boldsymbol{x}))),\boldsymbol{l}_0(\breve{h}^{\{k\}}(\boldsymbol{x}),\boldsymbol{\mathcal{X}}^{\Omega}(\boldsymbol{x}),\boldsymbol{\mathcal{X}}^{\partial\Omega}(\boldsymbol{x}))\right)=\boldsymbol{0}. \tag{24}$$

The equilibrium equations $\boldsymbol{F}_\Omega$ depend on the deformed lengths $\boldsymbol{l}(\boldsymbol{\mathcal{X}}^{\Omega},\boldsymbol{\mathcal{X}}^{\partial\Omega})$ and undeformed lengths $\boldsymbol{l}_0\left(\breve{h}^{\{k\}}(\boldsymbol{x}),\boldsymbol{\mathcal{X}}^{\Omega}(\boldsymbol{x}),\boldsymbol{\mathcal{X}}^{\partial\Omega}(\boldsymbol{x})\right)$ of the truss members. The truss member lengths $\boldsymbol{l}(\boldsymbol{\mathcal{X}}^{\Omega},\boldsymbol{\mathcal{X}}^{\partial\Omega})$ in turn depend on the interior nodes $\boldsymbol{\mathcal{X}}^{\Omega}$ and boundary nodes $\boldsymbol{\mathcal{X}}^{\partial\Omega}$. The undeformed lengths $\boldsymbol{l}_0\left(\breve{h}^{\{k\}},\boldsymbol{\mathcal{X}}^{\partial\Omega}\right)$ are evaluated at the midpoints of the truss members, which depend on the interior nodes $\boldsymbol{\mathcal{X}}^{\Omega}$ and the boundary nodes $\boldsymbol{\mathcal{X}}^{\partial\Omega}$. In addition, the ideal element length field $\breve{h}^{\{k\}}(\boldsymbol{x})$ changes as a function of $\boldsymbol{x}$ because of the RBF mapping. By taking the derivative of (24) w.r.t. to the control variables $\boldsymbol{x}$ we obtain

$$\begin{aligned}\frac{\mathrm{d}\boldsymbol{F}_\Omega}{\mathrm{d}\boldsymbol{x}}=&\frac{\partial\boldsymbol{F}_\Omega}{\partial\boldsymbol{l}}\frac{\partial\boldsymbol{l}}{\partial\boldsymbol{\mathcal{X}}^{\Omega}}\frac{\partial\boldsymbol{\mathcal{X}}^{\Omega}}{\partial\boldsymbol{x}}+\frac{\partial\boldsymbol{F}_\Omega}{\partial\boldsymbol{l}}\frac{\partial\boldsymbol{l}}{\partial\boldsymbol{\mathcal{X}}^{\partial\Omega}}\frac{\partial\boldsymbol{\mathcal{X}}^{\partial\Omega}}{\partial\breve{h}^{\{k\}}}\frac{\partial\breve{h}^{\{k\}}}{\partial\boldsymbol{x}}\\&+\frac{\partial\boldsymbol{F}_\Omega}{\partial\boldsymbol{l}_0}\frac{\partial\boldsymbol{l}_0}{\partial\breve{h}^{\{k\}}}\frac{\partial\breve{h}^{\{k\}}}{\partial\boldsymbol{x}}+\frac{\partial\boldsymbol{F}_\Omega}{\partial\boldsymbol{l}_0}\frac{\partial\boldsymbol{l}_0}{\partial\boldsymbol{\mathcal{X}}^{\Omega}}\frac{\partial\boldsymbol{\mathcal{X}}^{\Omega}}{\partial\boldsymbol{x}}+\frac{\partial\boldsymbol{F}_\Omega}{\partial\boldsymbol{l}_0}\frac{\partial\boldsymbol{l}_0}{\partial\boldsymbol{\mathcal{X}}^{\partial\Omega}}\frac{\partial\boldsymbol{\mathcal{X}}^{\partial\Omega}}{\partial\boldsymbol{x}}=\boldsymbol{0},\end{aligned} \tag{25}$$

to give

$$\begin{aligned}\left(\frac{\partial\boldsymbol{F}_\Omega}{\partial\boldsymbol{l}}\frac{\partial\boldsymbol{l}}{\partial\boldsymbol{\mathcal{X}}^{\Omega}}+\frac{\partial\boldsymbol{F}_\Omega}{\partial\boldsymbol{l}_0}\frac{\partial\boldsymbol{l}_0}{\partial\boldsymbol{\mathcal{X}}^{\Omega}}\right)\frac{\partial\boldsymbol{\mathcal{X}}^{\Omega}}{\partial\boldsymbol{x}}=&-\left(\frac{\partial\boldsymbol{F}_\Omega}{\partial\boldsymbol{l}}\frac{\partial\boldsymbol{l}}{\partial\boldsymbol{\mathcal{X}}^{\partial\Omega}}\frac{\partial\boldsymbol{\mathcal{X}}^{\partial\Omega}}{\partial\breve{h}^{\{k\}}}+\frac{\partial\boldsymbol{F}_\Omega}{\partial\boldsymbol{l}_0}\frac{\partial\boldsymbol{l}_0}{\partial\breve{h}^{\{k\}}}\right)\frac{\partial\breve{h}^{\{k\}}}{\partial\boldsymbol{x}}\\&-\frac{\partial\boldsymbol{F}_\Omega}{\partial\boldsymbol{l}_0}\frac{\partial\boldsymbol{l}_0}{\partial\boldsymbol{\mathcal{X}}^{\partial\Omega}}\frac{\partial\boldsymbol{\mathcal{X}}^{\partial\Omega}}{\partial\boldsymbol{x}}.\end{aligned} \tag{26}$$

From (26), we may solve for $\frac{\partial\boldsymbol{\mathcal{X}}^{\Omega}}{\partial\boldsymbol{x}}$, when $\left(\frac{\partial\boldsymbol{F}_\Omega}{\partial\boldsymbol{l}}\frac{\partial\boldsymbol{l}}{\partial\boldsymbol{\mathcal{X}}^{\Omega}}+\frac{\partial\boldsymbol{F}_\Omega}{\partial\boldsymbol{l}_0}\frac{\partial\boldsymbol{l}_0}{\partial\boldsymbol{\mathcal{X}}^{\Omega}}\right)$ and the right-hand side of (26) are known. We compute the right-hand side with a finite difference perturbation, and recall that $\left(\frac{\partial\boldsymbol{F}_\Omega}{\partial\boldsymbol{l}}\frac{\partial\boldsymbol{l}}{\partial\boldsymbol{\mathcal{X}}^{\Omega}}+\frac{\partial\boldsymbol{F}_\Omega}{\partial\boldsymbol{l}_0}\frac{\partial\boldsymbol{l}_0}{\partial\boldsymbol{\mathcal{X}}^{\Omega}}\right)$ is available from the Newton update, as noted in Section 5. Once we have solved for $\frac{\partial\boldsymbol{\mathcal{X}}^{\Omega}}{\partial\boldsymbol{x}}$, we obtain $\frac{\partial\boldsymbol{\mathcal{X}}}{\partial\boldsymbol{x}}$ as the union between $\frac{\partial\boldsymbol{\mathcal{X}}^{\partial\Omega}}{\partial\boldsymbol{x}}$ and $\frac{\partial\boldsymbol{\mathcal{X}}^{\Omega}}{\partial\boldsymbol{x}}$, where $\frac{\partial\boldsymbol{\mathcal{X}}^{\partial\Omega}}{\partial\boldsymbol{x}}$ is obtained numerically. We have verified that the (semi) analytical sensitivities are computed correctly by comparing them with numerical forward finite difference sensitivities obtained with a range of perturbations. Calculation of the finite difference sensitivities is done without Delaunay triangulation steps, to avoid the introduction of discontinuities (due to the addition or removal of nodes) in the finite difference calculations.

7. NUMERICAL STUDY

We will consider two remeshing strategies. Firstly, we use a remeshing strategy, which we will refer to as *uniform*, which uses an ideal element length field that is spatially uniform, and kept constant as the optimization iterations progress [7]. Secondly, we will use the newly developed remeshing strategy presented herein, which is characterized by a spatially varying ideal element field that changes as the optimization iterations progress, to which we will refer as *adaptive*.

The parameters used in the remeshing strategies, the FEAs and the optimization algorithm are as follows. For the adaptive remeshing strategy, we set the mesh refinement parameter p in (12) to 5, whereas the minimum element length $h_{\min}$ is selected as $0.1h^{\{0\}}$, unless otherwise stated. The material considered is an orthotropic Boron-Epoxy in a tape lay-out, that is, the fibers are all aligned in a single direction, aligned with the global x-axis. In other words, we do not determine an optimal fiber orientation. We assume plane stress conditions and use classical laminate theory. The material

properties used are a longitudinal Young's modulus of $E_1 = 228$ GPa, a transverse Young's modulus of $E_2 = 145$ GPa, and a shear modulus of $G_{12} = 48$ GPa. The last independent parameter in classical laminate theory is Poisson ratio $\nu_{12} = 0.23$, because ν_{21} follows from the symmetry relation $E_1\nu_{21} = E_2\nu_{12}$.

The selected parameters for the gradient-only conservative sequential spherical approximation algorithm [5] are the curvature factor $\alpha = 2$, initial curvature $c^{\{0\}} = 1$, convergence tolerance $\epsilon = 10^{-4}$, and a maximum number of outer iterations $k_{\max} = 300$. The Lagrange multiplier update step is selected as $\boldsymbol{\lambda}_s{}^{\{k+1\}} = \nabla^{\mathrm{T}}_{\boldsymbol{\lambda}} L\left(\left[\boldsymbol{x}^{\{k+1\}}\ \boldsymbol{\lambda}^{\{k\}}\right]\right)$. We also limit the maximum step size to 1, which was experimentally found to result in good convergence behavior, but this value will, in general, of course, strongly depend on the scaling of the problem. Before we proceed, we study the convergence behavior of the meshing strategies.

7.1. Convergence behavior

The convergence characteristics of the ZZ error indicator with established refinement strategies are well known, but we are herein interested in an indication of the effects of our own remeshing strategy in a shape optimization setting on convergence, if only qualitatively. We start the demonstration by considering a fixed boundary, being indicative of a converged shape design.

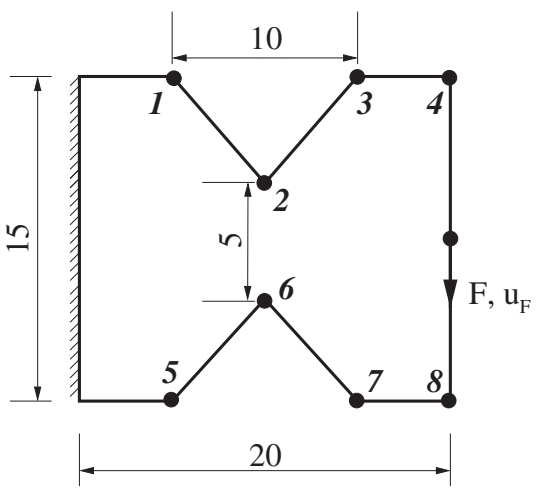

Figure 2. Bow-tie structure used to study the convergence behavior of the remeshing strategies.

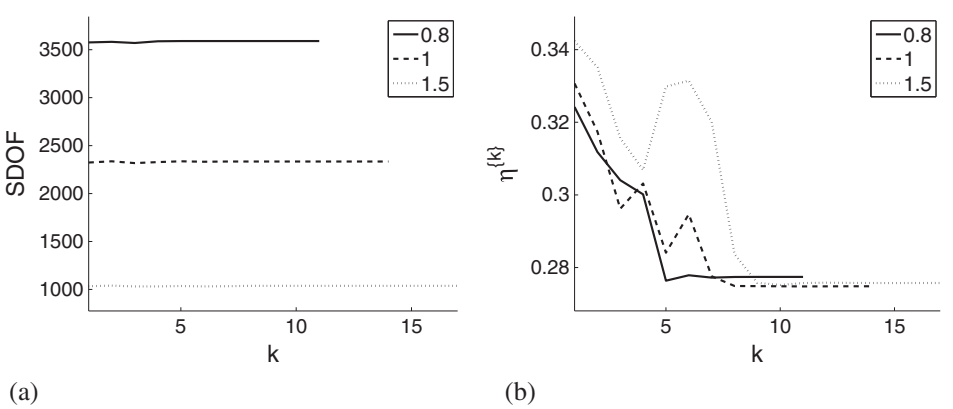

Figure 3. (a) System degrees of freedom (SDOF) and (b) global error $\eta^{\{k\}}$ for the mesh convergence study on the bow-tie structure for initial uniform element lengths $h_0 = \{1.5, 1, 0.8\}$.

Consider the bow-tie structure depicted in Figure 2. The structure is meshed using three initial undeformed truss lengths $h_0 = \{1.5, 1, 0.8\}$. The convergence criterion for the error indicator is given by

$$\frac{\eta^{\{k\}} - \eta^{\{k-1\}}}{\eta^{\{k\}}} < 10^{-6}. \tag{27}$$

Results are presented in Figure 3, with the system DOF (SDOF) depicted in Figure 3(a) and the global error indicator depicted in Figure 3(b). From Figure 3(a), it is clear that the SDOFs remain practically constant as the iterations progress. Figure 3(b) reveals that the final global error $\eta^{\{\max(k)\}}$ is lower than the global error of the initial uniform mesh $\eta^{\{1\}}$, for each of the three initial undeformed truss length choices. We also observe that the required number of refinement iterations reduces as the SDOFs increase. Note that the global errors $\eta^{\{k\}}$ for the various undeformed truss lengths cannot be compared with each other, because each uses a different $\breve{\boldsymbol{\sigma}}$ field. For each of the initial uniform element lengths $h_0 = \{1.5, 1, 0.8\}$, we depict the initial meshes in Figures 4(a)–(c), the final meshes in Figures 4(d)–(f) and the ideal element length fields in Figures 4(g)–(i).

We depict the convergence of the displacements of the bow-tie structure in Figure 5. To do so, we approximate the analytical solution u_{F}^{*} using Richardson's extrapolation method [23], because an analytical solution is not available for this problem. For Richardson's extrapolation method, we have used the initial uniform element lengths $h_0 = \{1.6, 0.8, 0.4\}$ to compute uniform meshes, with h_0 representative of the average element length. To approximate the asymptotic convergence rate, we fit a straight line in a least squares sense through the four data points with the highest SDOFs.

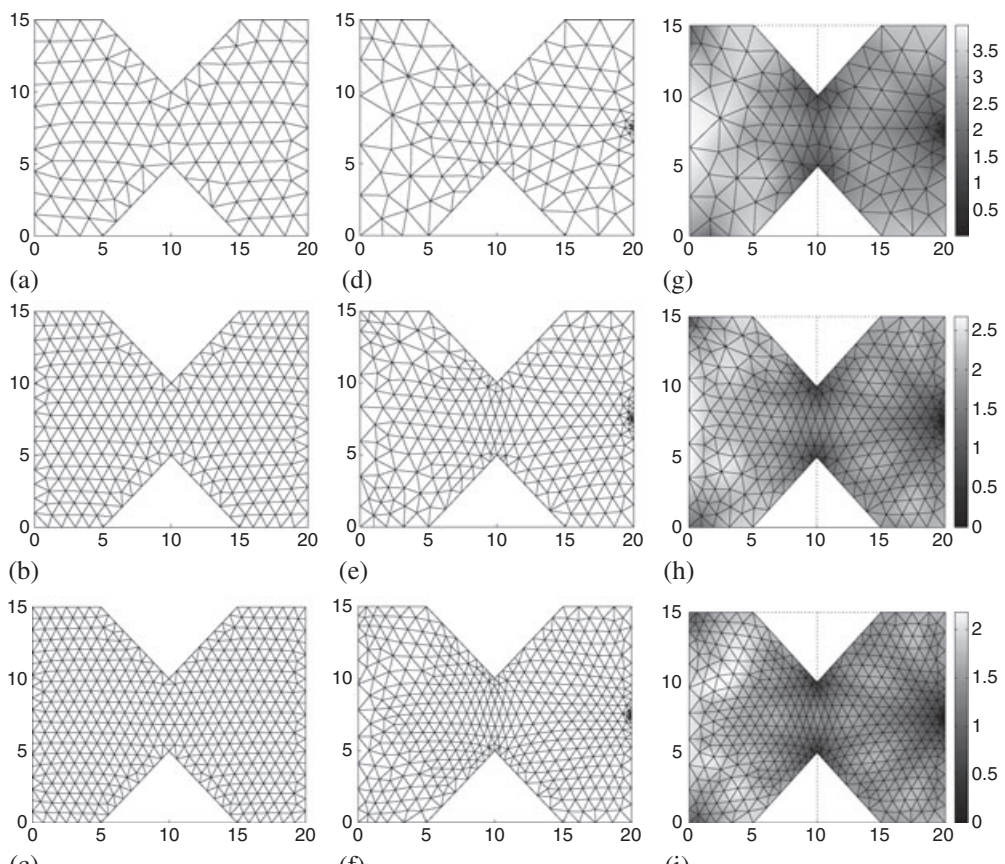

Figure 4. Convergence study showing (a)–(c) the initial mesh, (d)–(f) the final mesh and (g)–(i) the final ideal element length field of the bow-tie structure for various initial uniform element lengths $h_0 = \{1.5, 1, 0.8\}$.

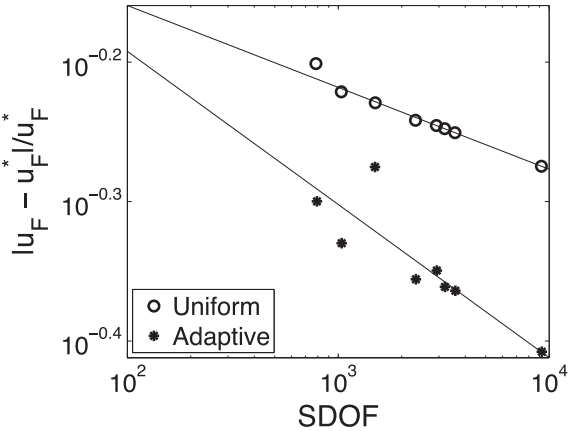

Figure 5. Approximated displacement convergence rate for the bow-tie structure problem using the uniform and adaptive mesh generators.

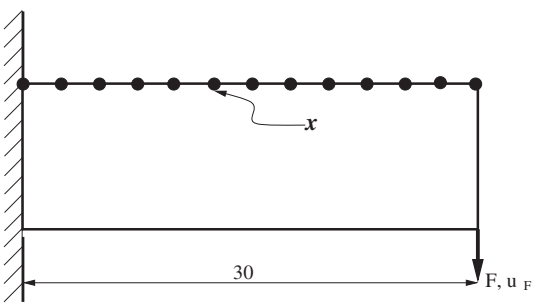

Figure 6. Initial geometry of the cantilever beam using 13 control points $\boldsymbol{x}$.

As shown, the error of the adaptive mesh is less than that of the uniform mesh for a given number of DOFs, whereas the convergence rate is superior.

Having offered some numerical evidence that our remeshing strategy seemingly converges for a fixed boundary, we now proceed to illustrate the same when both the boundary and error field are allowed to change simultaneously. We consider equality constrained example problems using both uniform and adaptive strategies. Note that the initial design is not required to satisfy the equality constraint (and indeed does not).

7.2. Simple cantilever beam

Next, we progress to the equality constrained design of the orthotropic cantilever beam depicted in Figure 6. The structure has a predefined length of 30 mm and a thickness of 1 mm. A point load F of 10 N acts at the bottom right corner of the structure. The boundary of the structure is controlled by the 13 control points or design variables $\boldsymbol{x}$ that can only move vertically. The boundary is interpolated linearly between the control points. We minimize the displacement u_{F} at the point of load application, subject to an equality constraint on volume, expressed as $V(\boldsymbol{x}) = V_0$, with $V_0 = 150$ mm^3, the prescribed volume of the structure.

Convergence histories for the value of the Lagrangian $L\left(\boldsymbol{x}^{\{k\}}, \lambda^{\{k\}}\right)$, the constraint function $\left|g(\boldsymbol{x}^{\{k\}})\right|$, the Lagrange multiplier $\lambda^{\{k\}}$ and the SDOFs are depicted in Figure 7(a)–(d) for the uniform and adaptive mesh generators. (We have used an initial ideal element length of 1.05 for the uniform mesh generator and one for the adaptive mesh generator to get a comparable number of SDOFs for the converged shapes.)

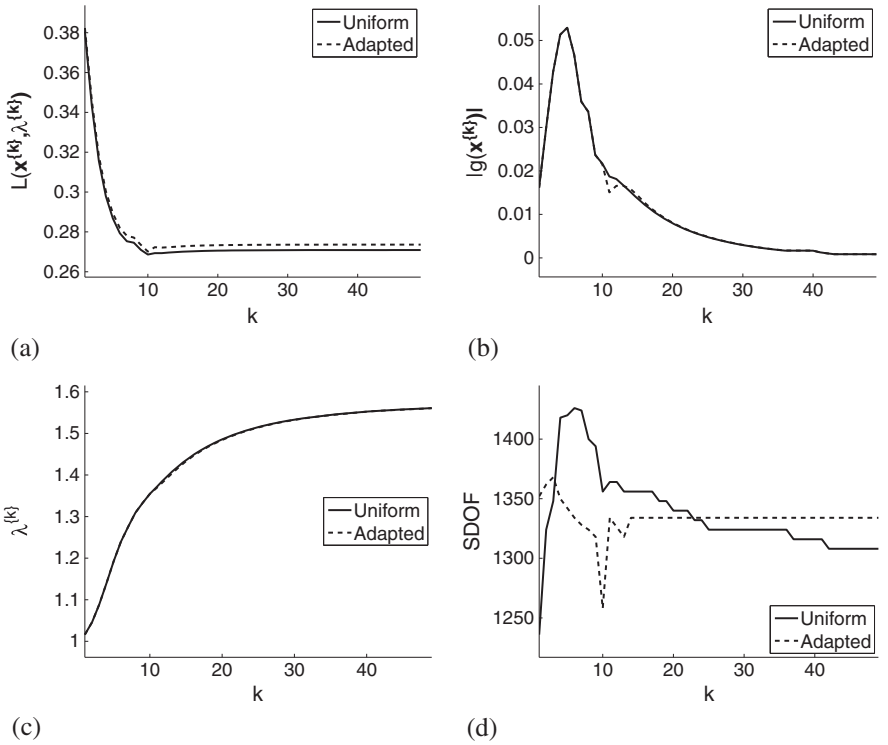

Figure 7. The cantilever beam convergence histories of (a) the Lagrangian $L\left(\boldsymbol{x}^{\{k\}}, \lambda^{\{k\}}\right)$, (b) absolute value of the constraint function $g\left(\boldsymbol{x}^{\{k\}}\right)$, (c) Lagrange multiplier $\lambda^{\{k\}}$, and (d) system degrees of freedom (SDOF) for a uniform and adapted mesh using initial ideal element lengths h_0 of respectively 1.05 and 1.

The required number of iterations and final designs are comparable. The SDOFs of the uniform remeshing strategy changes as the geometry varies, because the defined geometrical domain changes. The number of SDOFs of our adaptive remeshing strategy is roughly constant; the small variations present being the result of nodes being eliminated during convergence of the mesh generator. The interesting aspects of this example are depicted in Figure 8. The initial and final designs are depicted in Figure 8(a) and (b), and Figure 8(c) and (d), respectively, with the final ideal element length fields depicted in Figure 8(e) and (f), for the uniform and adaptive mesh generators, respectively—note the superiority of the latter mesh. The converged designs depicted in Figure 8(c) and (d) reflect the parabolic shape known as the analytical solution of the equivalent beam problem.

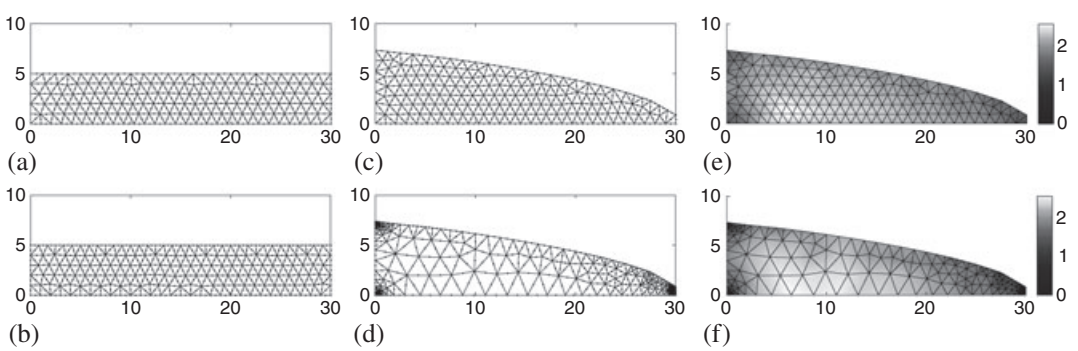

Figure 8. Initial (a) and (b) and final (c) and (d) designs of the cantilever beam with the associated final ideal element length field (e) and (f), for a uniform and adapted mesh using initial ideal element lengths h_0 of respectively 1.05 and 1.

7.3. Spanner design

Finally, we consider the shape design of the full spanner problem presented in Figure 9, which is subjected to multiple load cases. The objective is to minimize $\frac{1}{2}(u_{\mathrm{FA}} - u_{\mathrm{FB}})$, with u_{FA} and u_{FB} the vertical displacements at the point of load application, for the two independent load cases F_A and F_B, respectively. The spanner is subjected to an equality constraint on volume, expressed as $V(\boldsymbol{x}) = V_0$, with $V_0 = 70$ mm^3, the prescribed volume of the structure.

The upper and lower boundaries of the geometry are described using 11 control points each. The structure has a predefined length of 24 mm and thickness of 1 mm. The magnitude of the point loads F_A and F_B is 1 N each. Symmetry is *not* enforced; deviations from symmetry may be used to qualitatively evaluate the obtained designs because this problem should result in a symmetric geometry. The ideal element length field $\check{h}^{\{k\}}$ for the mesh is obtained by nodal averaging of the ideal element length fields obtained from the different load cases.

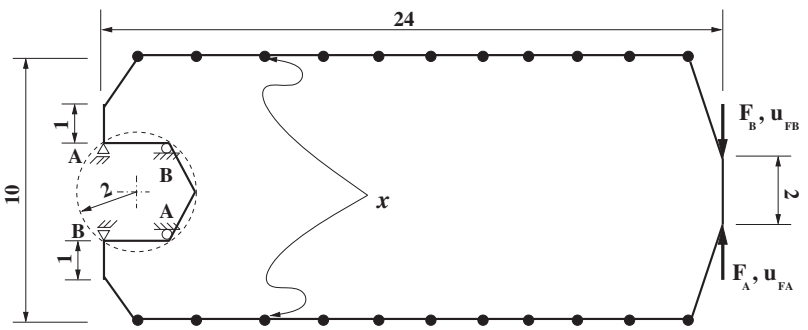

Figure 9. Initial geometry and loads of the full spanner problem using 22 control points $\boldsymbol{x}$.

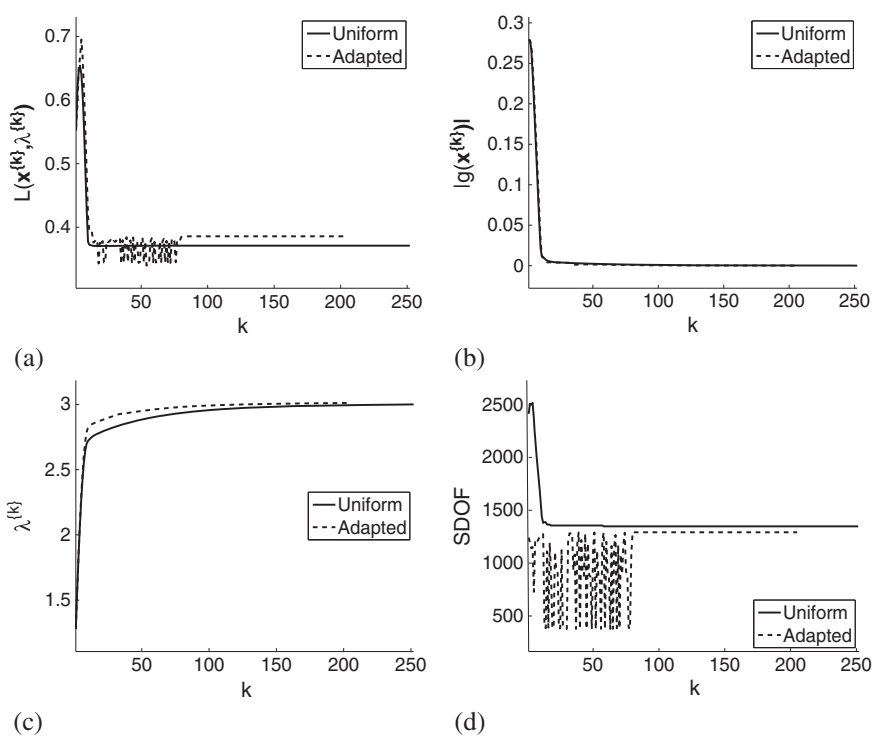

Figure 10. The full spanner convergence histories of (a) the Lagrangian $L\left(\boldsymbol{x}^{\{k\}}, \lambda^{\{k\}}\right)$, (b) absolute value of the constraint function $g\left(\boldsymbol{x}^{\{k\}}\right)$, (c) Lagrange multiplier $\lambda^{\{k\}}$, and (d) system degrees of freedom for a uniform and adaptive mesh using initial ideal element lengths h_0 of respectively 0.7 and 1.

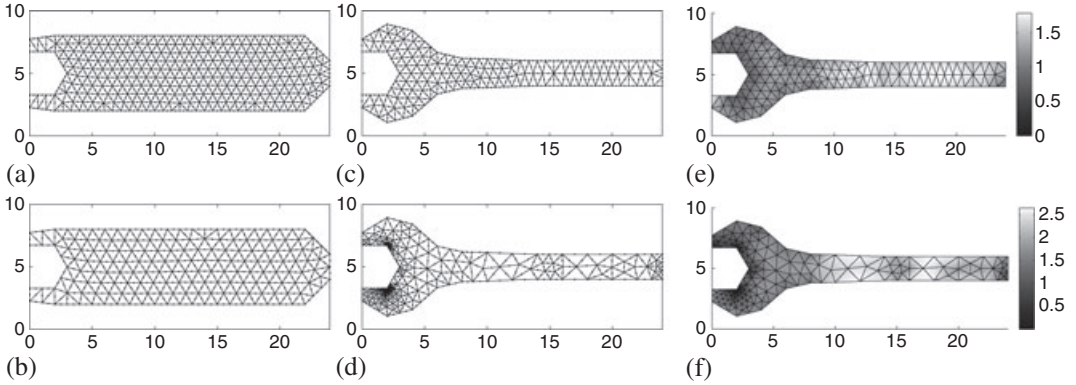

Figure 11. Initial (a) and (b) and final (c) and (d) designs of the full spanner with the associated final ideal element length field (e) and (f), for a uniform and adapted mesh using initial ideal element lengths h_0 of respectively 0.7 and 1.

Convergence histories for the value of the Lagrangian $L\left(\boldsymbol{x}^{\{k\}}, \lambda^{\{k\}}\right)$, the constraint function $\left|g\left(\boldsymbol{x}^{\{k\}}\right)\right|$, the Lagrange multiplier $\lambda^{\{k\}}$ and the SDOFs are depicted in Figure 10(a)–(d) for the uniform and adaptive mesh generators. This time, the required number of iterations for the uniform mesh generator is slightly more than that required for the adaptive mesh generator. Again, the SDOFs of the uniform remeshing strategy changes as the geometry varies, whereas the SDOFs of our adaptive remeshing strategy remains roughly constant after an initial unstable 80 iterations. The results depicted in Figure 11 compare well with results obtained in previous studies [24, 25]. Due to changes in the SDOF, as depicted in Figure 10(d), some oscillatory behavior is observed in the Lagrangian $L\left(\boldsymbol{x}^{\{k\}}, \lambda^{\{k\}}\right)$ within the first 70 iterations, as seen in Figure 10(a).

8. CONCLUSIONS

We have proposed a remeshing shape optimization methodology based on error estimates that only requires a single analysis and error calculation per iteration or shape design. This is in contrast to conventional approaches that often require multiple analyses per shape design. We have exploited gradient-only optimization to efficiently incorporate error indicators and refinement strategies, because we only require a single FEA followed by *a posteriori* error computation for each candidate shape design, without sacrificing optimization robustness.

We have demonstrated the proposed methodology by extending an existing uniform remeshing strategy [7] by incorporation of the very well known ZZ global error indicator. We improve significantly on the accuracy of the results obtained with uniform meshes. In addition, we have demonstrated convergence and efficiency of our strategy on two equality constrained example problems, albeit that we have not developed a theoretical framework that guarantees convergence at this stage.

In a follow-up study, we plan to linearize the error indicator field and investigate whether this improves the convergence of the gradient-only optimization strategies, and the implications for the development of theoretical frameworks for convergence.

APPENDIX A

The radial basis function $s(z)$ with $z \in \mathbb{R}^2$ for the two dimensional case, is given by

$$s(z) = \sum_{j=1}^{n_b} \alpha_j \phi\left(\left\|z - \boldsymbol{\chi}_j^{\partial\Omega}\right\|\right) + p(z), \tag{A.1}$$

with p a polynomial, n_b the number of boundary nodes and ϕ a given basis function with respect to the norm $\|z\|$. The coefficients α_j and the polynomial p are determined by the interpolation conditions

$$s\left(\boldsymbol{\mathcal{X}}_i^{\partial\Omega}\right) = \boldsymbol{d}_i^{\partial\Omega},\ i = 1, 2, \ldots, n_b \tag{A.2}$$

with $\boldsymbol{d}_i^{\partial\Omega}$ the displacement of the ith boundary node. In addition, it is required that

$$\sum_{j=1}^{n_b} \alpha_j q\left(\boldsymbol{\mathcal{X}}_j^{\partial\Omega}\right) = 0, \tag{A.3}$$

for all polynomials q with a degree less or equal than that of polynomial p. We rewrite (A.2) into a matrix form as

$$\boldsymbol{d}^{\partial\Omega} = \boldsymbol{M}\boldsymbol{\alpha} + \boldsymbol{P}\boldsymbol{\beta}, \tag{A.4}$$

where $\boldsymbol{M}$ is an $n_b \times n_b$ matrix with the ith row and jth column containing the evaluation of the basis function $\phi\left(\left\|\boldsymbol{\mathcal{X}}_i^{\partial\Omega} - \boldsymbol{\mathcal{X}}_j^{\partial\Omega}\right\|\right)$. For two dimensional interpolations $\boldsymbol{P}$ is an $n_b \times 3$ matrix with the ith row given by $\begin{bmatrix} 1 & \boldsymbol{\mathcal{X}}_{ix}^{\partial\Omega} & \boldsymbol{\mathcal{X}}_{iy}^{\partial\Omega} \end{bmatrix}$, where $\boldsymbol{\mathcal{X}}_{ix}^{\partial\Omega}$ and $\boldsymbol{\mathcal{X}}_{iy}^{\partial\Omega}$ are respectively the x and y coordinates of the ith boundary node. Similarly, (A.3) can be written in a matrix form

$$\boldsymbol{P}^{\mathrm{T}}\boldsymbol{\alpha} = \boldsymbol{0}. \tag{A.5}$$

We therefore need to solve the two systems of linear equations (A.4) and (A.5). We start by rewriting (A.4) to obtain

$$\boldsymbol{\alpha} = \boldsymbol{M}^{-1}\boldsymbol{d}^{\partial\Omega} - \boldsymbol{M}^{-1}\boldsymbol{P}\boldsymbol{\beta}, \tag{A.6}$$

which we substitute into (A.5) to obtain

$$\boldsymbol{P}^{\mathrm{T}}\left(\boldsymbol{M}^{-1}\boldsymbol{d}^{\partial\Omega} - \boldsymbol{M}^{-1}\boldsymbol{P}\boldsymbol{\beta}\right) = \boldsymbol{0}.$$

We then solve for $\boldsymbol{\beta}$ from

$$\boldsymbol{P}^{\mathrm{T}}\boldsymbol{M}^{-1}\boldsymbol{P}\boldsymbol{\beta} = \boldsymbol{P}^{\mathrm{T}}\boldsymbol{M}^{-1}\boldsymbol{d}^{\partial\Omega}, \tag{A.7}$$

and $\boldsymbol{\alpha}$ from (A.6). After solving for $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, the RBF $s(z)$ is defined and can be used to update the interior nodes $\boldsymbol{\mathcal{X}}^{\Omega}$ as the geometry changes. The boundary displacements $\boldsymbol{d}^{\partial\Omega}$ are obtained from the control variables $\boldsymbol{x}$ and piecewise linear boundary interpolation.

ACKNOWLEDGEMENT

This study was funded by the South African National Research Foundation (NRF) with contract/grant number 2769.

REFERENCES

1. Schleupen A, Maute K, Ramm E. Adaptive FE-procedures in shape optimization. *Structural and Multidisciplinary Optimization* 2000; **19**(4):282–302.
2. Ding Y. Shape optimization of structures: a literature survey. *Computers & Structures* 1986; **24**(6):985–1004.
3. Haftka RT, Grandhi RV. Structural shape optimization—a survey. *Computer Methods in Applied Mechanics and Engineering* 1986; **57**(1):91–106.
4. Kikuchi N, Chung KY, Torigaki T, Taylor JE. Adaptive finite element methods for shape optimization of linearly elastic structures. *Computer Methods in Applied Mechanics and Engineering* 1986; **57**(1):67–89.
5. Wilke DN, Kok S, Groenwold AA. The application of gradient-only optimization methods for problems discretized using non-constant methods. *Structural and Multidisciplinary Optimization* 2010; **40**(1–6):433–451.
6. Van Keulen F, Polynkine AA, Toropov VV. Shape optimization with adaptive mesh refinement: target error selection strategies. *Engineering and Optimization* 1997; **28**(1–2):95–125.

7. Wilke DN, Kok S, Groenwold AA. A quadratically convergent unstructured remeshing strategy for shape optimization. *International Journal for Numerical Methods in Engineering* 2006; **65**(1):1–17.
8. Persson PO, Strang G. A simple mesh generator in MATLAB. *SIAM Review* 2004; **46**(2):329–345.
9. Zienkiewicz OC, Zhu JZ. A simple error estimator and adaptive procedure for practical engineerng analysis. *International Journal for Numerical Methods in Engineering* 1987; **24**(2):337–357.
10. Bugeda G, Onate E. A methodology for adaptive mesh refinement in optimum shape design problems. *Computing Systems in Engineering* 1994; **5**(1):91–102.
11. Groenwold AA, Etman LFP. On the conditional acceptance of iterates in SAO algorithms based on convex separable approximations. *Structural and Multidisciplinary Optimization* 2010; **42**:165–178.
12. Fleury C, Braibant V. Structural optimization: a new dual method using mixed variables. *International Journal for Numerical Methods in Engineering* 1986; **23**:409–428.
13. Svanberg K. The method of moving asymptotes – a new method for structural optimization. *International Journal for Numerical Methods in Engineering* 1987; **24**:359–373.
14. Wilke DN, Kok S, Snyman JA, Groenwold AA. Gradient-only approaches to avoid spurious local minima in unconstrained optimization. *Optimization and Engineering* 2012. In press, available from Online First (DOI 10.1007/s11081-011-9178-7).
15. Wilke DN. Approaches to accommodate remeshing in shape optimization. *Ph.D*, University of Pretoria, Pretoria, South Africa, August 2010.
16. Zienkiewicz OC, Taylor RL, Zhu JZ. *The Finite Element Method*. Butterworth-Heinemann: Oxford, 2005.
17. Zienkiewicz OC. The background of error estimation and adaptivity in finite element computations. *Adaptive Modeling and Simulation* 2006; **195**(4–6):207–213.
18. Martinez R, Samartin A. Two-dimensional mesh optimization in the finite element method. *Computers & Structures* 1991; **40**(5):1169–1175.
19. Turcke D, Mcneice GM. Guidelines for selecting finite element grids based on an optimization study. *Computers & Structures* 1974; **4**(3):499–519.
20. Hinton E, Ozakca M, Rao NVR. An integrated approach to structural shape optimization of linearly elastic structures. Part II: shape definition and adaptivity. *Computing Systems in Engineering* 1991; **2**(1):41–56.
21. Sienz J, Hinton E. Reliable structural optimization with error estimation, adaptivity and robust sensitivity analysis. *Computers & Structures* 1997; **64**(1–4):31–63.
22. Subramanian G, Bose JC. Convenient generation of stiffness matrices for the family of plane triangular elements. *Computers & Structures* 1982; **15**(1):85–89.
23. Joyce DC. Survey of extrapolation processes in numerical analysis. *SIAM Review* 1971; **13**(4):435–490.
24. Herskovits J, Dias G, Santos G, Mota Soares CM. Shape structural optimization with an interior point nonlinear programming algorithm. *Structural and Multidisciplinary Optimization* 2000; **20**(2):107–115.
25. Wall WA, Frenzel MA, Cyron C. Isogeometric structural shape optimization. *Computer Methods in Applied Mechanics and Engineering* 2008; **197**(33–40):2976–2988.