Available online at www.sciencedirect.com

ScienceDirect

Procedia Engineering 163 (2016) 302 – 314

Procedia Engineering

www.elsevier.com/locate/procedia

25th International Meshing Roundtable

Tetrahedral mesh improvement using moving mesh smoothing and lazy searching flips

Franco Dassi[a,*], Lennard Kamenski[b], Hang Si[b]

[a]*Dipartimento di Matematica e Applicazioni, Università degli Studi di Milano Bicocca, Via Cozzi 53, 20125 Milano, Italy*
[b]*Weierstrass Institute for Applied Analysis and Stochastics, Mohrenstr. 39, 10117 Berlin, Germany*

Abstract

We combine the new *moving mesh smoothing*, based on the integration of an ordinary differential equation coming from a given functional, with the new *lazy flip* technique, a reversible edge removal algorithm for local mesh quality improvement. These strategies already provide good mesh improvement on themselves, but their combination achieves astonishing results not reported so far. Provided numerical comparison with some publicly available mesh improving software show that we can obtain final tetrahedral meshes with dihedral angles between 40° and 123°.

Peer-review under responsibility of the organizing committee of IMR 25

Keywords: mesh improvement, moving mesh, edge flipping, mesh quality, mesh smoothing

2010 MSC: 65N50, 65K10

1. Introduction

The two key operations for mesh improving are *smoothing* (which moves the mesh vertices) and *flipping* (which changes mesh topology without moving the mesh vertices). Previous work shows that the combination of these two operations achieves better results than if applied individually [1,2]. In this paper we combine a new smoothing and a new flipping methods to one mesh improvement scheme.

Flips are the most efficient ways to locally improve the mesh quality and they have been extensively addressed in the literature [1–4]. In the most simple cases, the basic flip operations, such as 2-to-3, 3-to-2, and 4-to-4 flips, are applied as long as the mesh quality can be improved. The more effective way is to combine several basic flip operations, such as the edge removal operation, which is an extension of the 3-to-2 and 4-to-4 flips. This operation removes an edge with $n \geq 3$ adjacent tetrahedra and replaces them by $m = 2n - 4$ new tetrahedra (the so-called an n-to-m flip). There are at most C_{n-2} possible cases, where $C_n = \frac{(2n)!}{(n+1)!\,n!}$ is the Catalan number. If n is small (e.g., $n < 7$), one can enumerate all the possible cases, compute the mesh quality for each of the individual cases, and then pick the optimal one. Another way is to use the dynamic programming to find the optimal configuration. However, the number of cases increases exponentially and finding the optimal solution with brute force is very time-consuming.

* Corresponding author.
E-mail address: franco.dassi@unimib.it, kamenski@wias-berlin.de, si@wias-berlin.de.

Peer-review under responsibility of the organizing committee of IMR 25
doi:10.1016/j.proeng.2016.11.065

In this paper, we propose the so-called *lazy searching flips*. The key idea is to automatically explore sequences of flips in order to remove a given edge in the mesh. If a sequence of flips leads to a configuration which doesn't improve the mesh quality, the algorithm reverses this sequence and explores another one (see section 3 and Figs 1a to 1c). Once an improvement is found, the algorithms stops the search and returns without exploring the remaining possibilities.

The *lazy searching flips* are accompanied with a smoothing procedure. Mesh smoothing improves the mesh quality by improving vertex locations, typically through Laplacian smoothing or some optimization-based algorithms. Most commonly used mesh smoothing methods are Laplacian smoothing and its variants [5,6], where a vertex is moved to the geometric center of its neighboring vertices. While economic, easy to implement, and often effective, Laplacian smoothing guarantees neither a mesh quality improvement nor the mesh validity.

Alternatives are optimization-based methods that are effective for a variety of mesh quality measures, e.g., for the ratio of the area to the sum of the squared edge lengths [7] or the ratio of the volume to a power of the sum of the squared face areas [8], the condition number of the Jacobian matrix of the affine mapping between the reference element and physical elements [9], or various other measures [1,10–12]. Most of the optimization-based methods are local and sequential, with Gauss-Seidel-type iterations being combined with location optimization problems over each patch. There is also a parallel algorithm that solves a sequence of independent subproblems [13].

In our scheme, we employ the moving mesh PDE (MMPDE) method, defined as the gradient flow equation of a meshing functional (an objective functional in the context of optimization) to move the mesh continuously in time. Such a functional is typically based on error estimation or physical and geometric considerations. Here, we consider a functional based on the equidistribution and alignment conditions [14] and employ the recently developed direct geometric discretization [15] of the underlying meshing functional on simplicial meshes.

Compared to the aforementioned mesh smoothing methods, the considered method has several advantages: it can be easily parallelized, it is based on a continuous functional for which the existence of minimizers is known, the functional controlling the mesh shape and size has a clear geometric meaning, and the nodal mesh velocities are given by a simple analytical matrix form. Moreover, the smoothed mesh will stay valid if it was valid initially [20].

In this paper we provide a detailed numerical study of a combination of the lazy searching flips with the MMPDE smoothing. More specifically, we compare the results of the whole algorithm with `Stellar` [2], `CGAL` [16] and `mmg3d` [17]. We also compare the lazy searching flips and the MMPDE smoothing with the flipping and smoothing procedures provided by `Stellar`.

2. The moving mesh PDE smoothing scheme

The key idea of this smoothing scheme is to move the mesh vertices via a moving mesh equation, which is formulated as the gradient system of an energy function (the MMPDE approach). Originally, the method was developed in the continuous form [18,19]. In this paper, we use its discrete form [15,20,21], for which the mesh vertex velocities are expressed in a simple, analytical matrix form, which makes the implementation more straightforward to parallelize.

2.1. Moving mesh smoothing

Consider a polygonal (polyhedral) domain $\Omega \subset \mathbb{R}^d$, $d \geq 1$, let the simplicial mesh under consideration be $\mathcal{T}_h$, and denote the numbers of its vertices and elements by $\#\mathcal{N}_h$ and $\#\mathcal{T}_h$. Let K be a generic mesh element and $\hat{K}$ the reference element taken as a regular simplex with the volume $|\hat{K}| = 1/\#\mathcal{T}_h$. Further, let F'_K be the Jacobian matrix of the affine mapping $F_K \colon \hat{K} \to K$ from the reference element $\hat{K}$ to a mesh element K. For notational simplicity, we denote the inverse of the Jacobia by $\mathbb{J}_K$, i.e., $\mathbb{J}_K \equiv (F'_K)^{-1}$. Then, the mesh $\mathcal{T}_h$ is uniform if and only if

$$|K| = \frac{|\Omega|}{\#\mathcal{T}_h} \quad \text{and} \quad \frac{1}{d}\,\mathrm{tr}\left(\mathbb{J}_K^T\mathbb{J}_K\right) = \det\left(\mathbb{J}_K^T\mathbb{J}_K\right)^{\frac{1}{d}} \quad \forall K \in \mathcal{T}_h. \tag{1}$$

The first condition requires all elements to have the same size and the second requires all elements to be shaped similarly to $\hat{K}$ (these conditions are the simplified versions of the equidistribution and alignment conditions [19,22]).

The corresponding energy function for which the minimization will result in a mesh satisfying (1) as closely as possible is

$$I_h = \sum_K |K|\; G\left(\mathbb{J}_K, \det \mathbb{J}_K\right) \quad \text{with} \quad G(\mathbb{J}, \det \mathbb{J}) = \theta\Big(\mathrm{tr}\left(\mathbb{J}\mathbb{J}^T\right)\Big)^{\frac{dp}{2}} + (1-2\theta)\, d^{\frac{dp}{2}} (\det \mathbb{J})^p, \tag{2}$$