Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Engineering 00 (2015) 000–000

Procedia Engineering

24th International Meshing Roundtable (IMR24)

Mesh smoothing: an MMPDE moving mesh approach

Weizhang Huang[a], Lennard Kamenski[b,*], Hang Si[b]

[a] *Department of Mathematics, The University of Kansas, Lawrence, Kansas, U.S.A.*
[b] *Weierstrass Institute, Berlin, Germany*

Abstract

We study a mesh smoothing algorithm based on the moving mesh PDE (MMPDE) moving mesh method. For the MMPDE, we employ a simple and efficient direct geometric discretization of the underlying meshing functional on simplicial meshes. The nodal mesh velocities can be expressed in a simple, analytical matrix form, which makes the implementation of the method easy, simple, and efficient. Numerical examples are provided.

Peer-review under responsibility of organizing committee of the 24th International Meshing Roundtable (IMR24).

Keywords: mesh smoothing, mesh optimization, moving mesh method, tetrahedral meshes

1. Introduction

Meshes generated with automatic tools often contain poorly shaped elements especially near domain boundaries and their quality needs to be improved before they can be used in the numerical solution of PDEs or other applications. Mesh smoothing improves the mesh quality by moving vertex locations. It is often effective in eliminating extremal dihedral angles in the mesh. Commonly used mesh smoothing methods are Laplacian smoothing and its variants (Field [4] and Lo [17]), where a vertex is moved to the geometric center of its neighboring vertices. While economic, easy to implement, and often effective, Laplacian smoothing does not guarantee improvements of the mesh quality.

Alternatives are optimization-based methods that are effective for a variety of mesh quality measures. For example, Bank [1] uses the ratio of the area to the sum of the squared edge lengths for triangular meshes, Shephard and Georges [18] employ the ratio of the volume to a power of the sum of the squared face areas for tetrahedral meshes, Freitag and Ollivier-Gooch [7] use various angle-based measures, and Freitag and Knupp [6] utilize the condition number of the Jacobian matrix of the affine mapping between the reference element and physical elements. Knupp [13,15] proposes several shape quality measures based on the Jacobian matrix and Canann et al. [2] propose a distortion metric for triangles and quadrilaterals. A parallel algorithm that solves a sequence of independent subproblems is proposed by Freitag et al. [5] for a class of local mesh-smoothing techniques.

* Corresponding author. Tel.: +49-30-20372-450 ; fax: +49-30-20372-317.
E-mail address: kamenski@wias-berlin.de

Peer-review under responsibility of organizing committee of the 24th International Meshing Roundtable (IMR24).

In this project, we study a mesh smoothing algorithm based on a moving mesh method and move the mesh continuously in time by means of a moving mesh PDE, defined as the gradient flow equation of a meshing functional. The key point of the approach is a simple and efficient direct geometric discretization of the meshing functional on simplicial meshes, which was recently introduced by Huang and Kamenski [9]. Most importantly, the nodal mesh velocities can be expressed in a simple, analytical matrix form, which makes the implementation of the method relatively easy and simple. It also makes the method amenable to the development of efficient solvers and parallel computation.

The new method has the following advantages: it is based on a continuous functional for which the existence of minimizers is known, the functional has a clear geometric meaning for controlling mesh shape and size quality, mesh velocities have a simple, analytical matrix form, which makes programming relatively easy, the resulting ODE systems can be solved locally or globally, and it is amenable to parallel computation.

2. MMPDE moving mesh method

In the moving mesh context, it is common to describe a moving mesh in terms of coordinate transformations. Assume that we are given a computational domain Ω_c, which can be the same as the physical domain $\Omega \subset \mathbb{R}^d$, $d \geq 1$. A moving mesh on Ω is the image of a computational mesh on Ω_c under a time dependent coordinate transformation $\boldsymbol{x} = \boldsymbol{x}(\boldsymbol{\xi}, t)\colon \Omega_c \to \Omega$. The coordinate transformation in the MMPDE method is determined as the solution of the gradient flow equation of a meshing functional (i.e., an objective functional to optimize). Many existing functionals can be written in the form

$$I[\boldsymbol{\xi}] = \int_\Omega G(\mathbb{J}, \det(\mathbb{J}), \mathbb{M}, \boldsymbol{x}) dx, \tag{1}$$

where $\boldsymbol{\xi} = \boldsymbol{\xi}(\boldsymbol{x}, t)\colon \Omega \to \Omega_c$ is the inverse coordinate transformation of $\boldsymbol{x} = \boldsymbol{x}(\boldsymbol{\xi}, t)$, G is a sufficiently smooth function in all of its arguments, $\mathbb{J} = \frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{x}}$ is the Jacobian matrix of $\boldsymbol{\xi} = \boldsymbol{\xi}(\boldsymbol{x}, t)$, and $\mathbb{M}$ is the metric tensor used for controlling mesh concentration (assumed to be symmetric and uniformly positive definite on Ω). Note that (1) is formulated in terms of the inverse coordinate transformation since functionals formulated this way are known to be less likely to produce singular coordinate transformations [3].

A number of meshing functionals have been developed in the past based on error estimates and physical and geometric considerations, e.g., see [12,14,16] and references therein. (For a detailed numerical comparison of various functionals see [10].)

Example 2.1 (Generalized Winslow's functional [20])**.** *This functional reads as*

$$I[\boldsymbol{\xi}] = \int_\Omega \operatorname{tr}(\mathbb{J}\mathbb{M}^{-1}\mathbb{J}^T)\, d\boldsymbol{x}, \tag{2}$$

where $\operatorname{tr}(\cdot)$ *is the trace of a matrix and* $\mathbb{M}^{-1}$ *serves as the diffusion matrix. The functional is coercive and convex and has a unique minimizer [12, Example 6.2.1].*

Since our interest is in the MMPDE method as a mesh smoothing scheme (in the Euclidean metric), we consider the simplest case $\mathbb{M} = I$*, for which the functional becomes*

$$I[\boldsymbol{\xi}] = \int_\Omega \operatorname{tr}(\mathbb{J}\mathbb{J}^T)\, d\boldsymbol{x}. \tag{3}$$

For this functional, we have $G = \operatorname{tr}(\mathbb{J}\mathbb{J}^T)$*.*

Example 2.2 (Huang's functional [8])**.** *This functional is based on* $\mathbb{M}$*-uniformity (requiring the mesh to be uniform in the metric* $\mathbb{M}$*),*

$$I[\boldsymbol{\xi}] = \theta \int_\Omega \sqrt{\det(\mathbb{M})}\big(\operatorname{tr}(\mathbb{J}\mathbb{M}^{-1}\mathbb{J}^T)\big)^{\frac{dp}{2}}\, d\boldsymbol{x} + (1-2\theta)d^{\frac{dp}{2}} \int_\Omega \sqrt{\det(\mathbb{M})}\left(\frac{\det(\mathbb{J})}{\sqrt{\det(\mathbb{M})}}\right)^p d\boldsymbol{x}, \tag{4}$$

where $0 \le \theta \le 1$ and $p > 0$ are dimensionless parameters. The $\mathbb{M}$-uniformity is mathematically equivalent to the so-called alignment (first term) and equidistribution (second term) conditions combined. For $0 < \theta \le \frac{1}{2}$, $dp \ge 2$, and $p \ge 1$, the functional is coercive and polyconvex and has a minimizer [12, Example 6.2.2]. For the case $\mathbb{M} = I$, (4) becomes

$$I[\boldsymbol{\xi}] = \theta \int_\Omega \left(\mathrm{tr}(\mathbb{J}\mathbb{J}^T)\right)^{\frac{dp}{2}} d\boldsymbol{x} + (1-2\theta) d^{\frac{dp}{2}} \int_\Omega \det(\mathbb{J})^p \, d\boldsymbol{x} \tag{5}$$

with $G = \theta\left(\mathrm{tr}(\mathbb{J}\mathbb{J}^T)\right)^{\frac{dp}{2}} + (1-2\theta) d^{\frac{dp}{2}} \det(\mathbb{J})^p$. The first term in (5) (alignment) with $\mathbb{M} = I$ represents a shape regularity measure while the second term (equidistribution) represents a volume uniformity measure (when $p > 1$). If $\theta = \frac{1}{2}$ and $dp = 2$ then the functional reduces to the Winslow's functional (3).

The MMPDE is defined as the gradient flow equation of $I[\boldsymbol{\xi}]$, i.e.,

$$\frac{\partial \boldsymbol{\xi}}{\partial t} = -\frac{P}{\tau} \frac{\delta I}{\delta \boldsymbol{\xi}},$$

where $\frac{\delta I}{\delta \boldsymbol{\xi}}$ is the functional derivative of I, τ is a positive constant used for adjusting the time scale of mesh movement, and $P = P(\boldsymbol{x}, t)$ is a positive function used for preserving certain scaling invariances; in our computation we choose $P = 1$. For the functional (1), this becomes

$$\frac{\partial \boldsymbol{\xi}}{\partial t} = \frac{P}{\tau} \nabla \cdot \left(\frac{\partial G}{\partial \mathbb{J}} + \frac{\partial G}{\partial \det(\mathbb{J})} \det(\mathbb{J}) \mathbb{J}^{-1} \right). \tag{6}$$

Using the identity

$$\frac{\partial \boldsymbol{x}}{\partial t} = -\mathbb{J}^{-1} \frac{\partial \boldsymbol{\xi}}{\partial t},$$

we have

$$\frac{\partial \boldsymbol{x}}{\partial t} = -\frac{P \mathbb{J}^{-1}}{\tau} \nabla \cdot \left(\frac{\partial G}{\partial \mathbb{J}} + \frac{\partial G}{\partial \det(\mathbb{J})} \det(\mathbb{J}) \mathbb{J}^{-1} \right). \tag{7}$$

After exchanging the roles of dependent and independent variables $\boldsymbol{\xi}$ and $\boldsymbol{x}$ on the right-hand side, the equation can be discretized on a computational mesh Ω_c and a set of mesh equations for the nodal mesh velocities can be obtained. Then, the mesh equations can be integrated with proper boundary conditions for the vertex locations of the underlying moving mesh.

3. The MMPDE mesh smoothing scheme

A simple approach was proposed in [9] for the implementation of variational mesh generation and adaptation. This approach can also be used for implementing the MMPDE moving mesh method described in the previous section. More specifically, we denote by $\mathcal{T}_h$ and $\mathcal{T}_{h,c}$ simplicial meshes on Ω and Ω_c, respectively, and assume that they have the same numbers of elements and vertices and the same connectivity. We also assume that $\mathcal{T}_{h,c}$ has been chosen to be almost uniform in the sense that all of its elements have almost the same size and are almost equilateral. We will see below that Ω_c and $\mathcal{T}_{h,c}$ are used only as an intermediate step and do not appear in the final formulation.

With the approach, the functional (1) is first approximated on $\mathcal{T}_h$ using a midpoint quadrature rule with $\mathbb{J}$ being approximated by the Jacobian matrix of the affine mapping between elements in $\mathcal{T}_h$ and their counterparts in $\mathcal{T}_{h,c}$; the Jacobian matrix can be computed using the edge matrices. The discretized functional is a function of the locations of the vertices of $\mathcal{T}_h$. By assumption, $\mathcal{T}_{h,c}$ is known and so are the locations of its vertices and, thus, the discretized functional is a function of the locations of the vertices of $\mathcal{T}_h$ only. The mesh equation for the vertex locations for $\mathcal{T}_h$ is obtained as the gradient equation of the discretized functional with respect to the locations. The derivatives of the discretized functional with respect to the locations can be expressed in a simple, analytical matrix form; the interested reader is referred to [9,11] for details

on the derivation. Notice that the functionals (2) and (4) are invariant under translations and rotations of the computational coordinate $\boldsymbol{\xi}$, which implies that the elements in $\mathcal{T}_{h,c}$, which are assumed to be almost equilateral, can be made to be similar to the master element $\hat{K}$ by translations and rotations. Moreover, they can be made almost identical to $\frac{1}{N}\hat{K}$, where N is the number of the mesh elements, since they have almost the same size. Thus, as long as the functional (1) is invariant under translations and rotation of $\boldsymbol{\xi}$, the computational elements appearing in the final formulation of the mesh equation can be replaced by $\frac{1}{N}\hat{K}$ where $\hat{K}$ is the master element assumed to be chosen as a regular simplex with the unitary volume.

Denote the locations of the vertices of $\mathcal{T}_h$ by $\boldsymbol{x}_i$, $i = 1, \ldots, N_v$, the element patch associated with vertex $\boldsymbol{x}_i$ by ω_i, the generic element in $\mathcal{T}_h$ by K, and the volume of K by $|K|$. Then the mesh equation is given by

$$\frac{d\boldsymbol{x}_i}{dt} = \frac{1}{\tau} \sum_{K \in \omega_i} |K| \boldsymbol{v}_{i_K}^K, \quad i = 1, \ldots, N_v, \tag{8}$$

where i_K is the local index of vertex $\boldsymbol{x}_i$ on K and the local mesh velocities are given by

$$\begin{bmatrix} (\boldsymbol{v}_1^K)^T \\ \vdots \\ (\boldsymbol{v}_d^K)^T \end{bmatrix} = -GE_K^{-1} + E_K^{-1} \frac{\partial G}{\partial \mathbb{J}} \hat{E} E_K^{-1} + \frac{\partial G}{\partial \det(\mathbb{J})} \frac{\det(\hat{E})}{\det(E_K)} E_K^{-1}, \qquad (\boldsymbol{v}_0^K)^T = -\sum_{j=1}^{d} (\boldsymbol{v}_j^K)^T.$$

$\hat{E}$ and E_K are the edge matrices of $\hat{K}$ and K and G, $\frac{\partial G}{\partial \mathbb{J}}$, and $\frac{\partial G}{\partial \det(\mathbb{J})}$ are evaluated at $\mathbb{J} = \hat{E}E_K^{-1}$ and $\det(\mathbb{J}) = \det(\hat{E})/\det(E_K)$.

The mesh velocities need to be modified for boundary vertices. For example, if $\boldsymbol{x}_i$ is a fixed boundary vertex, we can replace the corresponding equation by $\frac{\partial \boldsymbol{x}_i}{\partial t} = 0$. When $\boldsymbol{x}_i$ is allowed to move on a boundary curve (in 2D) or surface (in 3D) represented by $\phi(\boldsymbol{x}) = 0$, then the mesh velocity $\frac{\partial \boldsymbol{x}_i}{\partial t}$ needs to be modified such that its normal component along the curve or surface is zero, i.e., $\nabla\phi(\boldsymbol{x}_i) \cdot \frac{\partial \boldsymbol{x}_i}{\partial t} = 0$.

The mesh equation (8) can be integrated using explicit or implicit ODE solvers. It can also be first discretized in time and the linear system is solved globally (such as a Krylov-subspace method with preconditioning) or locally (such as Gauss-Seidel or Jacobi iteration). We can also just solve the balance equations (i.e., set velocities to zero) using global or local iterative methods.

In the following we provide a quick example for MMPDE-based smoothing to improve the mesh quality: smoothing of a randomly perturbed 2D mesh (Fig. 1) and of a tetrahedral meshe obtained by *TetGen* [19] for a cami1a part (Fig. 2). For our computation, we use Huang's functional (Example 2.2 with $\theta = 1/3$ and $p = 2$), which provides a better control of the mesh element sizing (equidistribution) than Winslow's functional. For the 3D mesh we compare the dihedral angle statistics of the original *TetGen* mesh with the dihedral angle statistics after a mesh smoothing iteration.

The proposed smoothing significantly reduces the number of small (0°–20°) as well as large (150°–180°) dihedral angles (Fig. 2b, marked with blue color). These first results look quite promising. More work will be done to investigate the potential of the MMPDE moving mesh approach for the mesh smoothing. Especially the combination of mesh smoothing with a reconnection step can provide a further improvement.

References

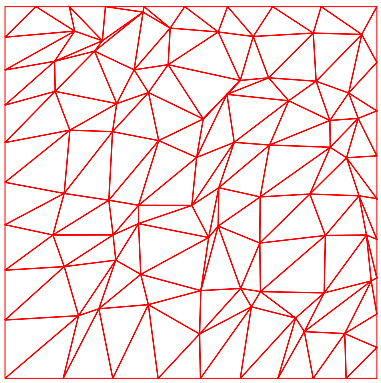
[1] R. E. Bank. *PLTMG: A Software Package for Solving Elliptic Partial Differential Equations. Users' Guide 7.0*. Frontiers Appl. Math. 15. SIAM, 1994.

[2] S. A. Canann, J. R. Tristano, and M. L. Staten. An approach to combined laplacian and optimization-based smoothing for triangular, quadrilateral, and quad-dominant meshes. In *Proceedings, 7th International Meshing Roundtable*, Sandia National Laboratories, Albuquerque, NM, 1998.

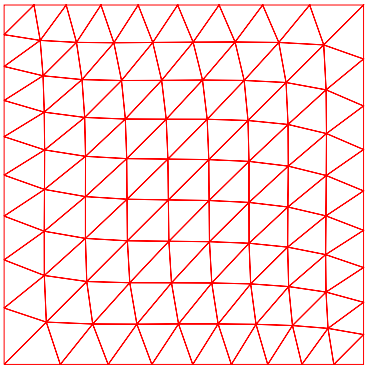
[3] A. S. Dvinsky. Adaptive grid generation from harmonic maps on Riemannian manifolds. *J. Comput. Phys.*, 95:450–476, 1991.

[4] D. A. Field. Laplacian smoothing and Delaunay triangulations. *Comm. Appl. Num. Meth.*, 4:709–712, 1988.

[5] L. Freitag, M. Jones, and P. Plassmann. A parallel algorithm for mesh smoothing. *SIAM J. Sci. Comput.*, 20, 1999.

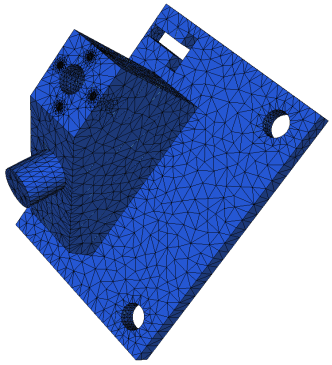


(a) initial, randomly perturbed mesh



(b) final, smoothed mesh

Figure 1: Smoothing of a randomly perturbed 2D mesh



(a) mesh example

angle	before	after	angle	before	after
0 – 5	1	0	80 – 110	30 060	30 087
5 – 10	335	15	110 – 120	5 417	5 607
10 – 20	4 086	3 617	120 – 130	4 329	5 026
20 – 30	8 083	8 701	130 – 140	2 895	3 830
30 – 40	12 172	13 704	140 – 150	1 676	1 739
40 – 50	15 338	15 389	150 – 160	1 033	458
50 – 60	17 175	16 740	160 – 170	307	3
60 – 70	15 208	14 851	170 – 175	0	0
70 – 80	15 703	14 051	175 – 180	0	0

(b) statistics of dihedral angles before and after smoothing

Figure 2: cami1a *TetGen* mesh, 22 303 elements

[6] L. A. Freitag and P. M. Knupp. Tetrahedral mesh improvement via optimization of the element condition number. *Internat. J. Numer. Methods Engrg.*, 53:1377–1391, 2002.

[7] L. A. Freitag and C. Ollivier-Gooch. Tetrahedral mesh improvement using swapping and smoothing. *International Journal for Numerical Methods in Engineering*, 40(21):3979–4002, 1997.

[8] W. Huang. Variational mesh adaptation: isotropy and equidistribution. *J. Comput. Phys.*, 174:903–924, 2001.

[9] W. Huang and L. Kamenski. A geometric discretization and a simple implementation for variational mesh generation and adaptation. *J. Comput. Phys. (submitted)*, 2014. (arXiv:1410.7872).

[10] W. Huang, L. Kamenski, and R. D. Russell. A comparative numerical study of meshing functionals for variational mesh adaptation. *J. Math. Study.*, 48(2):168–186, 2015. (arXiv:1503.04709).

[11] W. Huang, L. Kamenski, and H. Si. Mesh smoothing: an MMPDE approach. WIAS Preprint Nr. 2130, 2015.

[12] W. Huang and R. D. Russell. *Adaptive Moving Mesh Methods*. Springer, New York, 2011. Applied Mathematical Sciences Series, Vol. 174.

[13] P. Knupp. Achieving finite element mesh quality via optimization of the Jacobian matrix norm and associated quantities. Part I – A framework for surface mesh optimization. *Int. J. Numer. Meth. Engng.*, 48:401–420, 2000.

[14] P. Knupp and S. Steinberg. *Fundamentals of Grid Generation*. CRC Press, Boca Raton, 1994.

[15] P. M. Knupp. Applications of mesh smoothing: copy, morph, and sweep on unstructured quadrilateral meshes. *Internat. J. Numer. Methods Engrg.*, 45:37–45, 1999.

[16] V. D. Liseikin. *Grid Generation Methods*. Springer, Berlin, 1999.

[17] S. H. Lo. A new mesh generation scheme for arbitaar planar domains. *Int. J. Numer. Meth. Engng.*, 21:1403–1426, 1985.

[18] M. Shephard and M. Georges. Automatic three-dimensional mesh generation by the finite octree technique. *Int. J. Numer. Meth. Engrg.*, 32:709–749, 1991.

[19] H. Si. `http://tetgen.org`.

[20] A. M. Winslow. Adaptive mesh zoning by the equipotential method. Technical Report UCID-19062, Lawrence Livemore Laboratory, 1981 (unpublished).