Journal of Computational Physics 301 (2015) 322–337

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

A geometric discretization and a simple implementation for variational mesh generation and adaptation ☆

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

[a] The University of Kansas, Department of Mathematics, Lawrence, KS 66045, USA
[b] Weierstrass Institute for Applied Analysis and Stochastics, Berlin, Germany

ARTICLE INFO

Article history:
Received 1 November 2014
Received in revised form 31 July 2015
Accepted 19 August 2015
Available online 31 August 2015

Keywords:
Variational mesh generation
Mesh adaptation
Moving mesh

ABSTRACT

We present a simple direct discretization for functionals used in the variational mesh generation and adaptation. Meshing functionals are discretized on simplicial meshes and the Jacobian matrix of the continuous coordinate transformation is approximated by the Jacobian matrices of affine mappings between elements. The advantage of this direct geometric discretization is that it preserves the basic geometric structure of the continuous functional, which is useful in preventing strong decoupling or loss of integral constraints satisfied by the functional. Moreover, the discretized functional is a function of the coordinates of mesh vertices and its derivatives have a simple analytical form, which allows a simple implementation of variational mesh generation and adaptation on computer. Since the variational mesh adaptation is the base for a number of adaptive moving mesh and mesh smoothing methods, the result in this work can be used to develop simple implementations of those methods. Numerical examples are given.

1. Introduction

The basic idea of the variational approach of mesh generation and adaptation is to generate an adaptive mesh as an image of a given reference mesh under a coordinate transformation determined by a functional (which will hereafter be referred to as a meshing functional). Typically, the meshing functional measures difficulties of the numerical approximation of the physical solution and involves a user-prescribed metric tensor or a monitor function to control the mesh adaptation. The advantage of the variational approach is the relative ease of incorporating mesh requirements such as smoothness, adaptivity, or alignment in the formulation of the functional [2]. The variational approach is commonly used to generate structured meshes but it can be employed to generate unstructured meshes as well [3]. Moreover, it is the base for a number of adaptive moving mesh methods [11,12,14,20].

A number of variational methods have been developed in the past; e.g., see Thompson et al. [22], Knupp and Steinberg [19], Liseikin [21], Huang and Russell [11] and references therein. Noticeably, Winslow [23] proposed an equipotential method based on variable diffusion. Brackbill and Saltzman [2] developed a method by combining mesh concentration, smoothness, and orthogonality. Dvinsky [5] used the energy of harmonic mappings as his meshing functional. Knupp [16] and Knupp and Robidoux [17] developed functionals based on the idea of conditioning the Jacobian matrix of the coordinate

☆ Supported in part by the National Science Foundation (USA) under Grant DMS-1115118.
* Corresponding author.
E-mail addresses: whuang@ku.edu (W. Huang), kamenski@wias-berlin.de (L. Kamenski).

http://dx.doi.org/10.1016/j.jcp.2015.08.032
0021-9991/

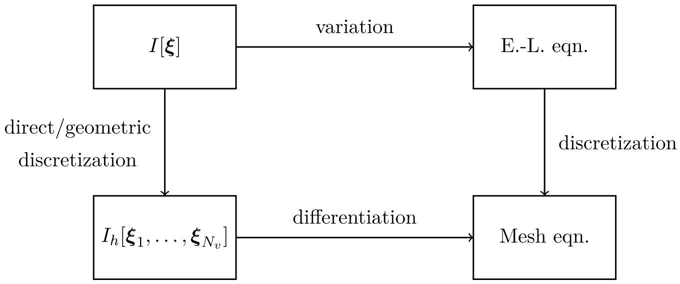

Fig. 1. Possible solution strategies for variational mesh generation and adaptation.

transformation. Huang [9] and Huang and Russell [11] developed functionals based on the so-called equidistribution and alignment conditions.

A common solution strategy for the existing variational methods is to first derive the Euler–Lagrange equation of the underlying meshing functional and then discretize it on either a physical or a computational mesh (cf. Fig. 1). If the discretization is done on a computational mesh, the Euler–Lagrange equation needs to be transformed by exchanging the roles of dependent and independent variables. Although this strategy works well for the most cases, the corresponding formulation can become complicated and its implementation requires a serious effort, especially in three dimensions; cf. [11, Chapt. 6]. Moreover, the geometric structure of the meshing functional can be lost in the process of spatial discretization of the Euler–Lagrange equation.

The objective of this paper is to study a new discretization and solution strategy. We consider simplicial meshes and approximate the underlying meshing functional directly. Although the direct discretization of variational problems is not new on itself, its employment in the context of variational mesh generation and adaptation is new. The Jacobian matrix of the coordinate transformation involved in the meshing functional is not discretized directly; instead, it is approximated by the Jacobian matrices of affine mappings between simplicial elements. The advantage of this geometric discretization is that it preserves the basic geometric structure of the continuous functional, which is useful in preventing strong decoupling or loss of integral constraints satisfied by the underling functional (cf. Castillo [4]). In particular, it preserves the coercivity and convexity for two examples of meshing functionals we consider (see Section 3.4). Moreover, as we will see in Section 3, the discretized functional is a function of the coordinates of vertices of the computational mesh and its derivatives have a simple analytical form. This allows a simple (and parallel) implementation of the corresponding variational meshing method.

The outline of the paper is as follows. Section 2 briefly describes the variational approach in mesh generation. Section 3 presents the direct discretization for meshing functionals and gives the analytical formula for the derivatives of the discretized functional with respect to the computational coordinates of mesh vertices. Several numerical examples are presented in Section 4, followed by conclusions and further remarks in Section 5. For completeness and for the convenience of users who prefer the physical coordinates as unknown variables, the derivatives of the discretized functional with respect to the physical coordinates are given in Appendix A.

2. The variational approach for mesh generation

Let Ω and Ω_c be the physical and computational domains in $\mathbb{R}^d$ $(d \ge 1)$, which are assumed to be bounded, simply connected, and polygonal/polyhedral. Generally speaking, Ω_c can be chosen to be the same as Ω but there are benefits to choose it to be convex, including that the to-be-determined coordinate transformation is less likely to be singular, see Dvinsky [5]. We also assume that we are given a symmetric and uniformly positive definite metric tensor $\mathbb{M} = \mathbb{M}(\boldsymbol{x})$ in Ω, which provides the information about the size and shape of mesh elements. Typically, $\mathbb{M}$ is defined in a mesh adaptation process based on the physical solution, solution error, or other physical considerations.

Denote the coordinates on Ω and Ω_c by $\boldsymbol{x}$ and $\boldsymbol{\xi}$, the corresponding coordinate transformation by $\boldsymbol{x} = \boldsymbol{x}(\boldsymbol{\xi})\colon \Omega_c \to \Omega$ and its inverse by $\boldsymbol{\xi} = \boldsymbol{\xi}(\boldsymbol{x})\colon \Omega \to \Omega_c$. Meshing functionals are commonly formulated in terms of the inverse coordinate transformation because the coordinate transformation determined in this way is less likely to be singular [5].

We consider a general meshing functional

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

where $\mathbb{J} = \frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{x}}$ is the Jacobian matrix of $\boldsymbol{\xi} = \boldsymbol{\xi}(\boldsymbol{x})$ and G is a given smooth function (with respect to all of its arguments). This form is very general and includes many existing meshing functionals, e.g., see Knupp and Steinberg [19], Liseikin [21], and Huang and Russell [11]. To be instructive, we consider two examples in the following. (For a detailed numerical comparison of various functionals see [15].)