ARTICLE IN PRESS

Advances in Engineering Software xxx (xxxx) xxx–xxx

Contents lists available at ScienceDirect

Advances in Engineering Software

journal homepage: www.elsevier.com/locate/advengsoft

Research paper

MATLAB 2D higher-order triangle mesh generator with finite element applications using subparametric transformations

T.V. Smitha, K.V. Nagaraja*, Sarada Jayan

Department of Mathematics, Amrita University, Bengaluru, 560035, India

ARTICLE INFO

Keywords:
Higher order triangular elements
Mesh generation
Subparametric transformations
Finite element method
Parabolic arcs
Curved boundary

ABSTRACT

This paper presents a novel automated higher-order (HO) unstructured triangular mesh generation of the two-dimensional domain. The proposed HO scheme uses the nodal relations obtained from subparametric transformations with parabolic arcs, especially for curved geometry. This approach is shown to drastically simplify the computational complexities involved in the HO finite element (HOFE) formulation of any partial differential equation (PDE). The prospective generalised MATLAB 2D mesh generation codes, *HOmesh2d* for the regular domain and *CurvedHOmesh2d* for a circular domain are based on the MATLAB mesh generator *distmesh* of Persson and Strang. As an input, the code takes a signed distance function of the domain geometry and the desired order for the triangular elements and as outputs, the code generates an HO triangular mesh with element connectivity, node coordinates, and boundary data (edges and nodes). The working principle of HOFE scheme, using subparametric transformations with the proposed HO automated mesh generator is explained. The simplicity, efficiency, and accuracy of the HOFE method, with the proposed HO automated mesh generator up to 28-noded triangular elements, are illustrated with elliptic PDE. The proposed techniques are applied to some electromagnetic problems. The use of higher order elements from the proposed mesh generator is shown to increase the accuracy and efficiency of the numerical results. Also, with the proposed HOFE scheme it is verified that HO elements significantly decrease the numbers of degrees of freedom, and elements required to achieve a specific level of accuracy compared to lower order elements. Numerical results show that the HO elements outperform the lower order elements in terms of efficiency and accuracy of the numerical results.

1. Introduction

In real life situation, most of the engineering models have complex curved geometries and finding an approximate solution using any numerical technique has always been challenging. Finite Element Analysis has become very popular to the modern engineer due to the increase in computational resources, for solving such complex problems [1–8]. Mesh generation, which is the first step, is a crucial prerequisite in Finite element method (FEM) as well as in many other applications (like computer graphics, scientific computing).

The objective of the present work is to provide a simple automated mesh generation code in MATLAB using HO triangular elements for any regular or curved geometries. This approach will be very useful to solve large-scale engineering problems in Mechanical, Aerospace, Civil, Biomechanics, and Electromagnetic etc. According to Ergatoudis et al. [9] and Babuska et al. [10], mesh with HO elements is proved to accelerate the accuracy, stability, and efficiency of computational processes for solving various applications in FEM. Numerical accuracy and efficiency of HO elements for the analysis of models containing curved boundaries using subparametric transformations are demonstrated in [11–19].

In [11] the isoparametric point transformations by the parabolic or cubic curves in the standard triangle are developed. Later, a better point transformations called the subparametric mapping (parabolic arcs) to match curved boundaries for HO triangular elements was used in [14–17]. Further, these developments were put into practical use in [15–19]. Though the earlier works proved the efficiency and accuracy of using HO elements in FEM especially for curved geometries, an HO automatic mesh generator for curved boundaries remained a challenge. All the available mesh generating codes are very complex and uses either linear or quadratic triangular elements. Therefore, the available codes are difficult to integrate with other FEM codes. Although triangular elements up to quadratic order have been used extensively in the literature, the application of HO elements has received much less attention, presumably because of the complexities related to the mesh optimization, and computational difficulties which arise in finite

* Corresponding author.
E-mail address: kv_nagaraja@blr.amrita.edu (K.V. Nagaraja).

http://dx.doi.org/10.1016/j.advengsoft.2017.10.012
Received 1 February 2017; Received in revised form 18 September 2017; Accepted 18 October 2017
0965-9978/

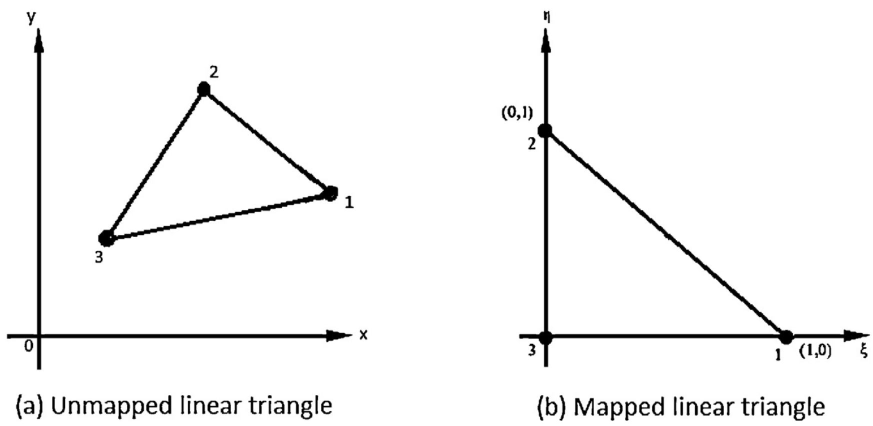

Fig. 1. 3-node linear triangular element mapped into the standard (master) triangle.

element formulations of a PDE. In the FEM formulation of a second order linear PDE using one sided curved triangular elements, we come across integrals of rational functions with denominators that are high order bivariate polynomials. While using isoparametric transformation for HO one side curved triangular elements, we get the Jacobian of the transformation as a bivariate polynomial of order ≥ 2. Hence, the integrand that arises in the finite element formulation of the PDE will be complex to evaluate. To avoid this difficulty, we use the subparametric transformation with parabolic arcs, the Jacobian of which will always be a bivariate linear polynomial for any HO one sided curved triangular element (cubic, quartic, quintic, sextic etc.)[14–18]. Thus, finite element applications can be efficiently and easily carried out by using subparametric transformations with the proposed simple automated HO mesh generator in MATLAB with unstructured triangles for any regular or curved geometry.

The proposed MATLAB functions for two-dimensional mesh generation using HO triangular elements are *HOmesh2d* for regular domains and *CurvedHOmesh2d* for circular domains. The MATLAB function *CurvedHOmesh2d* can be easily modified for any irregular or curved domain.

The proposed code is based on a simple and efficient MATLAB mesh generation code *distmesh* developed by Persson and Strang [20]. Their algorithm typically produces high-quality meshes and compared to other meshing techniques, it is shorter and simpler. Koko [21] has proposed a simple unstructured mesh generator based on *distmesh*, and a fast refinement procedure *kmg2d* and *kmg2dref* for finite element applications in MATLAB. *HOmesh2d* and *CurvedHOmesh2d* work with either *distmesh* or *kmg2d*. The geometrical description of the domain has to be simplified by the signed distance functions as seen in [20]. We have provided the complete proposed MATLAB functions, *HOmesh2d* and *CurvedHOmesh2d* in Appendix A.2 and A.3.

In this paper, the automated Lagrange interpolation functions generation as per the required node distribution proposed for the mesh generator is described in Section 2, and the developed complete MATLAB function *Gen_LagSF* is given in Appendix A.1. In Section 3, we present the detailed explanation of the MATLAB fragment of the proposed 2D HO mesh generator *HOmesh2d* and *CurvedHOmesh2d* for the quartic case. Mesh generation of the other HO triangular elements is obtained likewise. Later in Section 4, we have illustrated in a flowchart, the finite element application of the proposed HO mesh generator. This flowchart will facilitate any MATLAB user to develop generalized and efficient code to solve easily any PDE using HO elements. The results of the proposed procedure are provided for elliptic PDE with few numerical examples focused on the application in Electromagnetics. It is shown that using the proposed HO 2D mesh generators, *HOmesh2d* and *CurvedHOmesh2d* any PDE can be comfortably solved using unstructured triangular elements for finite element applications, which significantly improves accuracy and efficiency of the method by using subparametric transformation with parabolic arcs for curved geometries.

2. Automatic Lagrange shape functions generation

In finite element analysis, interpolation functions are used to express the variation of the field variable within an element by its nodal values as in [14]:

$$u = \sum_{k=1}^{\frac{(n+1)(n+2)}{2}} N_k^{(n)}(\xi,\ \eta) u_k^e \tag{1}$$

where u is the field variable at any point within the element, u_k^e are the nodal values of u for each element and the interpolation functions (N_k) is referred to triangular element shape functions or basis polynomials of order n at the node k. In FEM, we need to transform each triangular element to a standard (or master) triangular element in order to simplify the numerical integrations, for which the expression for numerical integration is feasible. The standard triangle is an isosceles right-angled triangle with equal sides being unity. This element is shown in Fig. 1. The coordinates (ξ, η) are called as natural coordinates and the coordinates (x, y) are called Cartesian coordinates.

Lagrange interpolation functions are widely used in practice. In Eq. (1), the assumed functions take on the same values as the given functions at specified points. For HO elements simpler and suitable shape functions can be derived using Lagrange interpolation polynomials. For one-dimensional case, Lagrange interpolation functions at node i is defined by,

$$\begin{aligned} L_i(x) &= \prod_{k=1,\ k\neq i}^{n} \frac{x - x_k}{x_i - x_k},\ \textit{in Cartesian coordinate} \\ L_i(\xi) &= \prod_{k=1,\ k\neq i}^{n} \frac{\xi - \xi_k}{\xi_i - \xi_k},\ \textit{in Natural coordinate.} \end{aligned} \tag{2}$$

It is clear from (2) that for an element with n nodes, $L_i(x)$ or $L_i(\xi)$ will have $n - 1$ degrees of freedom (DOF). Similarly, Lagrange interpolation functions for two-dimensional elements can be derived as the product of one-dimensional Lagrange interpolation functions. Thus, for two-dimensional elements, Lagrange shape functions are denoted by $N_i(x, y)$ or $N_i(\xi, \eta)$ and are defined as

$$N_i(\xi,\ \eta) = L_i(\xi) L_i(\eta).$$

We have developed an automated Lagrange shape functions generation MATLAB function *Gen_LagSF*. The distribution of the nodes over the area of the sextic order triangle in the Cartesian $x - y$ or natural $\xi - \eta$ plane is as shown in Fig. 2 below:

Gen_LagSF.m generates and displays the generalised Lagrange coefficients and shape functions for triangular elements up to octic order in anticlockwise sequence for the nodes distributed as shown in Fig. 2 (see Appendix A.1).