Advances in Engineering Software 90 (2015) 107–118

Contents lists available at ScienceDirect

Advances in Engineering Software

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

Volume element model mesh generation strategy and its application in ship thermal analysis

S. Yang [a,*], E. Dilay [a], J.C. Ordonez [a], J.V.C. Vargas [a], J. Chalfant [b]

[a] Department of Mechanical Engineering, Energy and Sustainability Center, and Center for Advanced Power Systems, Florida State University, Tallahassee, FL 32310, USA
[b] MIT Sea Grant Design Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139, USA

ARTICLE INFO

Article history:
Received 4 May 2015
Revised 16 July 2015
Accepted 9 August 2015
Available online 25 August 2015

Keywords:
Volume element model
Ship mesh generation
Ship thermal analysis
All-electric ships
Hexahedral mesh
Ray tracing

ABSTRACT

This paper introduces a mesh generation strategy devised and implemented for the volume element model (VEM), and elaborates key contributions of the strategy in enhancing the VEM as a prominent tool in ship thermal modeling and simulation. The VEM mesh generation strategy employs ray crossings and ray– triangle intersection algorithms developed in previous studies, and constructs sufficiently accurate geometric representations of the whole ship within permissible time frame using hexahedral meshes. In addition, this work demonstrates the strategy's practicality in thermal analysis of a notional all-electric ship, which is characterized by intricate structures and multiple internal components, i.e., thermal loads. Ship thermal solutions obtained in this assessment verify the proposed mesh generation strategy's ability to improve the overall computational efficiency of the VEM, by allowing it to obtain plausible thermal solutions with respect to time and space using a coarse independent mesh.

1. Introduction

Volume element model (VEM) is a three-dimensional (3D) dynamic reduced-order model developed for its application in thermal systems engineering [1]. The scheme discretizes the domain of interest (i.e., system under analysis) in three dimensions using a cell-centered finite-volume scheme [2]. Subsequently, fundamental laws of thermodynamics and heat transfer are applied to each cell in conjunction with empirical correlations in order to quantify the energy transfer between cells. The resulting system of nonlinear ordinary differential equations (ODEs) with respect to time is solved using an appropriate numerical method according to the problem under consideration. The scheme has been employed to study thermal responses of several dynamic systems [1,3,4], and it has been experimentally validated using Power Electronic Building Blocks [5] and an Off-Grid Zero Emissions Building [6].

The mesh generation strategy devised and implemented into the VEM plays a key role in the enhancement of the overall VEM computational efficiency, defined as accuracy achieved per unit execution time [2]. The proposed VEM mesh generation strategy creates a ship mesh composed of structured hexahedral elements within acceptable time frame, and it allows the VEM to obtain plausible numerical solutions using a coarse independent mesh. Furthermore, the strategy allows for the visualization and analysis of VEM computational grids and numerical results using VisIt visualization software [7].

In recent years, several studies have been published proposing methods to construct and refine hexahedral meshes [8–17]. Mesh generation techniques and algorithms developed in previous works created 3D meshes with great accuracy as a result of novel refinement techniques like the octree node-boundary adjustment. The technique presented by Ito et al. [13], for instance, is an effective mesh generation and refinement method which allows to construct computational grids of large and complex geometries with a small number of hexahedral elements. In addition, Ruiz et al. [16] combined grid-based and advancing front methods to generate unstructured hexahedral meshes of exterior domains and reproduce the shape of distorted boundaries.

According to [18], hexahedral meshes are often preferred over other meshes (e.g., tetrahedral), when available, for a variety of reasons. Typically, tetrahedral meshes require 4–10 times more elements than a hexahedral mesh to obtain the same level of accuracy [19,20]. Furthermore, in some types of numerical approximations (e.g., high deformation structural finite element analysis), tetrahedral elements are mathematically *stiffer* due to a reduced number of degrees of freedom associated with a tetrahedral element [21]. In their paper [18], Shepherd and Johnson also outline existing constraints of

* Corresponding author. Tel: +1- 937- 657- 4156.
E-mail addresses: syang@caps.fsu.edu (S. Yang), Emerson.dilay@up.com.br (E. Dilay), ordonez@caps.fsu.edu (J.C. Ordonez), jvargas@caps.fsu.edu (J.V.C. Vargas), chalfant@mit.edu (J. Chalfant).

http://dx.doi.org/10.1016/j.advengsoft.2015.08.003
0965-9978/

Nomenclature

c	specific heat (*J*/kg K)
k	thermal conductivity (*W*/m K)
L	length (m)
N	total number
n	number
$\vec{\mathbf{n}}$	normal
$\mathbf{P}$	array of volume element vertex coordinates
p	pressure (Pa)
$\dot{Q}$	heat transfer rate (W)
q	point
$\mathbf{R}$	ray
$\mathbf{T}$	triangle
T	temperature
t	time
V	volume (m^3); vertex
x, y, z	Cartesian coordinates

Greek letters

β	divider thickness (m)
δ	volume element length (m)
ε	mesh refinement relative error
$\boldsymbol{\pi}$	plane
ρ	density (kg/m^3)
φ	relative humidity
Φ	energy (J)
Ψ	solution vector

Subscripts

conv	convection
b	bottom
div	divider
e	east
eq	equivalent
gen	generation
i	VE index
j	VE face index
max	maximum
min	minimum
m	mesh
n	north
R	actual component weighted average
s	south
t	top
v	vapor
VE	volume element
vs	saturation pressure
w	west
x	*x*-direction
y	*y*-direction
z	*z*-direction

hexahedral meshes and propose possible solutions to overcome such limitations.

Hexahedral meshes have been employed to solve a variety of problems ranging from CFD to metal forming processes [9–11,17,19,21,22]. In [10], Kwak and Im studied the practicality of employing hexahedral elements for 3D metal forming simulations. Based on their assessment using distributions of effective strain-rate graident and *posteriori* error values, the authors construed that the proposed hexahedral mesh could be effectively used to simulate metal formations. De Santis et al. [17] developed a full-hexahedral structured meshing technique for image-based computational vascular modeling. The authors indicated that in solving the Navier– Stokes equations in a left coronary artery, tetrahedral (unstructured) meshes needed much higher resolution than structured hexahedral meshes to reach mesh independency, with higher computational costs. Such differences were also reported in other studies and were attributed to the misalignment of meshes with the flow direction, causing high numerical diffusion error associated with unstructured meshes [22,23].

Among numerous dynamic systems, ship thermal modeling and simulation, in particular, generally require an exorbitant amount of time to obtain mesh-independent solutions due to large and complex ship geometry (e.g., curved surfaces) and multiple internal components. As a result, numerous ship mesh generation techniques have been presented previously [24–27] in efforts to expedite mesh construction and refinement processes for finite element method. These techniques, however, are restricted to finite element analysis which usually demands high computing power to generate mesh and simulate large systems such as ships. Therefore, the objective of this paper is to present the effort to apply hexahedral mesh generation techniques and algorithms developed previously [13,28,29] to the VEM. In addition, the practicality of the proposed mesh generation strategy in ship thermal modeling and simulation is discussed with an example problem.

2. Mesh generation strategy

The computational domain in the VEM is discretized into a coarse structured mesh constituted of lumped control volumes, i.e., volume elements (VEs). The mesh generation strategy proposed in this paper consists of the following steps: (1) extraction of geometric information from a STL file; (2) generation of an enclosing mesh block that has sufficient size to comprise the entire CAD geometry and (3) intersection of the CAD geometry with the enclosing mesh block to produce the actual mesh. Furthermore, the strategy includes two additional steps in preparation for the numerical solver associated with the VEM: (i) assignment of physical properties to each corresponding volume element in the computational domain and (ii) writing output files for VisIt and the solver. The mesh generation algorithm is written in FORTRAN to facilitate its coupling with the VEM solver [30].

2.1. Extraction of CAD geometry from the Stereolithography file

The mesh generation process initiates by reading the geometric information from a STL file in ASCII format, then numbers each triangle and saves the coordinates of its vertex and unit normal vector into separate arrays. In addition, the algorithm reads an user-input text file including: the total desired number of VEs in *x*, *y*, and *z* directions, number of vertical (e.g., walls, bulkheads, etc.) and horizontal (e.g., decks) dividers with predefined thicknesses, and spatial location, physical dimensions, as well as component (e.g., ship equipment) properties. Ships, for instance, feature bulkheads and decks to represent distinct compartments, zones, etc., and they often prevent the energy transfer between these spaces via heat and/or mass. VEs representing dividers may or may not have smaller volume depending on their predefined thickness. Assignment of multiple ship components within the mesh will further be discussed in Section 2.4.

2.2. Hexahedral mesh block generation

The next step in the VEM mesh generation strategy is to create an enclosing mesh block of hexahedral elements as shown in Fig. 1. The extreme values of *x*, *y*, and *z* (x_{min}, x_{max}, y_{min}, y_{max}, z_{min}, and z_{max}) are identified in the array of vertices extracted from the STL file discussed in Section 2.1. Then the enclosing mesh block dimensions are determined as: $L_x = x_{max} - x_{min}$, $L_y = y_{max} - y_{min}$, and $L_z = z_{max} - z_{min}$ in the *x*, *y* and *z* directions, respectively.

In order to fit all VEs within the enclosing mesh block, the element size (i.e., volume) is computed based on L_x, L_y, L_z, the total number of