Advanced Engineering Informatics 38 (2018) 357–369

Contents lists available at ScienceDirect

Advanced Engineering Informatics

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

Full length article

Feature-based intelligent system for steam simulation using computational fluid dynamics

Lei Li[a], Carlos F. Lange[a], Zhen Xu[b], Pingyu Jiang[c], Yongsheng Ma[a,*]

[a] Department of Mechanical Engineering, University of Alberta, Edmonton, Canada
[b] Department of Electrical and Computer Engineering, University of Alberta, Edmonton, Canada
[c] School of Mechanical Engineering, Xi'an Jiaotong University, Xi'an, China

ARTICLE INFO

Keywords:
Artificial intelligence
Feature-based modeling
Computational fluid dynamics
Robust simulation
Information consistency

ABSTRACT

In the development of products involving fluids, computational fluid dynamics (CFD) has been increasingly applied to investigate the flow associated with various product operating conditions or product designs. The batch simulation is usually conducted when CFD is heavily used, which is not able to respond to the changes in flow regime when the fluid domain changes. In order to overcome this defect, a rule-based intelligent CFD simulation system for steam simulation is proposed to analyze the specific product design and generate the corresponding robust simulation model with accurate results. The rules used in the system are based on physical knowledge and CFD best practices which make this system easy to be applied in other application scenarios by changing the relevant knowledge base. Fluid physics features and dynamic physics features are used to model the intelligent functions of the system. Incorporating CAE boundary features, the CFD analysis view is fulfilled, which maintains the information consistency in a multi-view feature modeling environment. The prototype software tool is developed by Python 3 with separated logics and settings. The effectiveness of the proposed system is proven by the case study of a disk-type gate valve and a pipe reducer in a piping system.

1. Introduction

As computational fluid dynamics (CFD) is gaining maturity rapidly, it is extensively applied in product development. In practice, CFD can be used to investigate the flow under various product operating conditions. Each operating condition is associated with a specific functioning of a component which affects the flow field. CFD can also be applied to improve the design in CFD-based optimization [1] in which metamodeling [2] is commonly used as the algorithm. Even though the design of experiments (DOE) [3] can be adopted to select a reasonable number of points to represent the design space and reduce the computational cost, a considerable number of CFD tests is still needed. In such kind of application scenarios, CFD is heavily used to process the simulations of different operating conditions or different designs. In order to eliminate the idle time, the simulations are usually conducted in batch mode in which the pre-defined solver configuration is applied to all the design points. When there are big changes in design which induce flow regime alteration, the solver would not be able to respond to the changes. As a result, wrong simulation results will be generated.

To conquer this deficiency, an intelligent CFD simulation system is proposed in this paper. This system is supposed to analyze each specific design and configure the solver with the best-fit physics models intelligently. An effective approach to achieve this is to embed knowledge into the system. The knowledge is represented as rules and coded into the system. Such kind of rule-based system is also known as the expert system which is the simplest form of artificial intelligence [4]. The rules are established using physical knowledge and CFD best practices, and the whole system is developed by feature modeling.

The review of the feature modeling technology and the CFD best practices is introduced in the next section. The structure, modeling, and implementation of the proposed intelligent CFD simulation system are given in Section 3. Following that, the case study of a disk-type gate valve and a pipe reducer in Section 4 is used to show how the proposed system works. The conclusion of the contribution and future work is made at last.

2. Literature review

2.1. Feature-based modeling

Feature-based modeling has been widely used in different aspects of engineering such as design, modeling, process control, and system

* Corresponding author.
E-mail address: yongsheng.ma@ualberta.ca (Y. Ma).

https://doi.org/10.1016/j.aei.2018.08.011
Received 19 March 2018; Received in revised form 9 August 2018; Accepted 16 August 2018
1474-0346/

integration [5]. In its early development, features are specifically designated as form features which are generic shapes for product development purposes [6]. For example, there may be form features like the hole, slot, pocket, and chamfer in a product model [7]. In practice, constructive solid geometry (CSG) and boundary representation (B-rep) are commonly used to represent geometry [8]. Among those two schemas, CSG represents the geometry at the implicit level while B-rep is an explicit representation scheme in which an object is formed by its boundary like faces, edges, and vertices [9].

Later, features are further developed to model the non-geometric product properties which are useful in different stages of the whole product lifecycle. Hence, the feature definition is usually driven by a specific application in product development [10]. In the conceptual design stage, the design intent is embedded in the customer's requirement for functions, which is a set of geometric and functional rules satisfied by the final product [11]. Cheng and Ma [12] propose functional features to interpret the design intent and provide modeling guidance during the detailed design stage. In the analysis stage using computer-aided engineering (CAE), CAE features are used to represent engineering analysis knowledge [10]. In the product assembly stage, an assembly feature is defined as a generic way to mate the components by relationships [13,14]. In the manufacturing stage, a machining feature can be defined as an object with geometric and topological characteristics which are associated with a set of machining operations [15]. As an early application of the feature technology, machining feature also extends its applicability in many aspects. For example, Liu and Ma [16] introduce 2.5D machining features into topology optimization to improve the manufacturability of the optimized product. The method was later extended to design for hybrid additive-subtractive manufacturing [17]. Further, associative optimization features are proposed to capture the optimization intent in the optimization stage [18]. Clearly, features have their specific definitions in different product development stages, which hinders the interoperability. Therefore, a generic feature definition is needed to associate product geometry and engineering knowledge in different applications [19]. Specifically, the generic feature is defined as the most basic feature entity template in an object-oriented software engineering approach to abstract the semantic patterns for different applications in engineering [20].

As this paper's focus is on CFD simulation, which is a typical application of CAE, the development of CAE feature needs to be reviewed in detail. In order to improve the efficiency of CAE simulation, the product modeled by computer-aided design (CAD) should be simplified in advance [21]. For this purpose, idealization features are introduced to facilitate the detail removal and dimension reduction of CAD model [22]. Similarly, in the work reported by Hamri et al. [23], the simplification features are also defined to remove certain form features in the CAD model. CAD-CAE features are proposed by Deng et al. to transform the CAD features into features in the analysis model [24]. In a CAD/CAE incorporate software framework presented by Xia et al., CAE features consist of geometry entities and analysis attributes which can be categorized as boundary conditions feature, material feature, mesh feature, rendering feature and so on [25]. Those reported works are more focused on the geometry conversion from CAD domain to CAE analysis. However, the analysis models which are a major constituent of CAE simulation have been paid less attention. Actually, how to adaptively improve the analysis model to increase accuracy is crucial to non-experts in simulation-based design [26]. In the authors' previous work [27], fluid physics features are put forward to analyze the product design and select the appropriate physics models for simulation. Meanwhile, dynamic physics features are developed to facilitate the generation of a robust simulation model for each design. The utilization of these two feature regimes should be the right approach to construct the intelligent CFD simulation system.

2.2. CFD best practices

CFD is a powerful tool to analyze the fluid flow problems. The solving space is usually a fluid domain which can be created by CAD software, and defeaturing is required for a good quality CFD solution [28]. Besides, the domain size will affect the simulation time and accuracy [29], which should be carefully tested especially for new problems.

The fluid domain needs to be discretized into elements for numerical calculation. There are two types of mesh, namely structured mesh and unstructured mesh [30]. Even though the unstructured mesh tends to have larger artificial diffusion and takes more time to solve, it is the most used mesh type because it is more efficient in preparing an adequate grid. The quality of the mesh is significant to the accuracy of the solution. Ideally, the mesh should provide evenly distributed levels of truncation error [31]. The truncation error is proportional to the grid spacing, which can be reduced by refining the mesh. The refinement over the entire domain is at high computational cost and not necessary because large error only appears in small regions in most cases. As a result, local mesh refinement is the right approach to increase the accuracy at specific locations. For example, mesh inflation should be applied along solid walls to have the grid surfaces aligned with the boundary layer flow approximately [32]. The mesh can be further refined by adaptive meshing, which is based on the solution [33].

In addition to discretization, the fluid domain should also be confined by boundary conditions which drive the flow inside the domain [34]. Boundary conditions are properties and values assigned to the fluid boundaries. A fluid boundary is an external surface of a fluid domain which supports the inlet, outlet, opening, wall, and symmetry boundary conditions. The inlet boundary condition is the most important one, where the fluid predominantly flows into the domain. Correspondingly, the outlet specifies the area where the fluid flows out of the domain. Either velocity or pressure can be set at those boundaries. Among all the possible combinations, the inlet with velocity assigned and outlet with static pressure assigned lead to the most robust boundary condition setup [35]. The opening boundary should be applied if the direction of the flow is uncertain, which means the fluid can flow in and out of the domain at the opening boundary simultaneously. It is suggested to use this boundary type only as part of the preliminary investigation because it introduces an increased uncertainty in the solution. The wall boundary defines the area where the fluid cannot penetrate. Especially for the no-slip wall boundary, the fluid has zero velocity relative to the boundary. If there is a plane that satisfies both geometric and physical symmetry, the symmetry boundary condition can be applied to this plane where the diffusive flux is zero [36].

Before the solving stage, appropriate physical models need to be specified for the solver. The flow regime, such as laminar or turbulent flow and flow compressibility should always be checked first to select the correct models. At the beginning of the simulation, instead of using higher order schemes and advanced turbulence models, first order schemes and k-ε turbulence model which is the most commonly used model in industry applications [37] should be chosen in favor of convergence.

After the pre-processing is done, the simulation can be started and it will stop when the convergence criteria are met or the maximum number of iterations is reached. On condition that there is no error occurring, the post-processing can be conducted to analyze and visualize the solution. If there is a convergence problem found after the solving stage, only one modification in the model configuration should be made to identify the key factor. In such situations, more robust schemes, such as upwind differencing scheme (UDS) [38] for advection, Euler Implicit [39] for time, k-ε for turbulence, should be considered. If a steady simulation diverged, switching to transient simulation helps to test whether the flow is unsteady.

Following the successful application of lower order schemes, higher order schemes like central differencing scheme (CDS) [38] are preferred