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
Available online 23 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

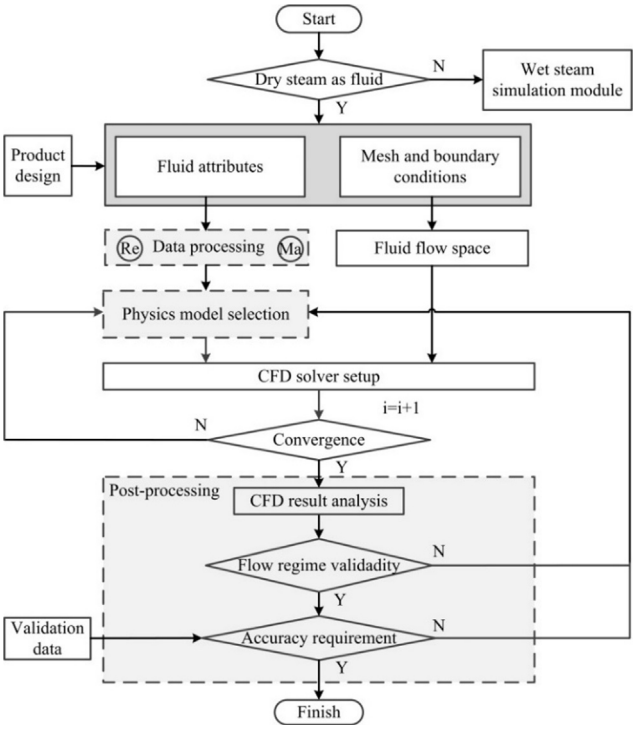

Fig. 1. Structure of the intelligent CFD simulation system.

to increase accuracy. Because there is no universal turbulence model, the effect of different turbulence models should be tested, if the flow is turbulent [40]. Before performing these actions, the mesh should be refined accordingly. At last, if an analytical or experimental result is available, the numerical solution should be validated by comparing with the existing solution [41].

3. Intelligent CFD simulation system

3.1. Structure of the system

According to the type of steam used, dry steam or wet steam simulation module will be entered separately. The details of the intelligent CFD simulation system are introduced in this subsection based on dry steam simulation. The wet steam simulation module will be described in the next subsection. The intelligent CFD simulation system has the overall structure shown in Fig. 1.

Based on the CAD model of the product, the fluid domain can be abstracted and discretized with boundary conditions attached, which is the input of the CFD solver. The fluid attributes are obtained from the product design. The other parameters that are essential to CFD simulation can be derived in the data processing module using the equations listed below:

$$\rho = \frac{p}{RT} \tag{1}$$

$$a = \sqrt{kRT} \tag{2}$$

$$\mathrm{Re} = \frac{\rho \bar{v} D}{\mu} \tag{3}$$

$$\mathrm{Ma} = \frac{\bar{v}}{a} \tag{4}$$

where ρ is the density of steam, p is the pressure of steam, R is the gas constant, T is the temperature of steam, a is the speed of sound in steam, k is the specific heat ratio of steam, Re is the Reynolds number, $\bar{v}$ is the average velocity of steam, D is the inner diameter of duct, μ is the dynamic viscosity of steam, and Ma is the Mach number. The non-dimensional Reynolds number and Mach number determine the flow regime, which is used to select the correct models in the physics model selection module.

Fig. 2 shows how the physics model selection module works. For flow in a pipe to be fully turbulent, the Reynolds number should be greater than 4000 [42], then a turbulence model will be selected. The compressibility of the fluid is judged by the Mach number. If the Mach number is bigger than 0.3, the compressibility effect is not negligible, then the total energy model will be selected. And at the same time, the reference pressure, as well as proper boundary conditions, should be configured to activate the compressible flow simulation. In the beginning of simulation when the index i (iteration) is small or at the time the simulation has convergence issues, lower order discretization schemes like UDS and the commonly used k-ε turbulence model for turbulent flow are preferred to facilitate convergence.

The root-mean-square (RMS) of normalized residuals is used as the convergence criteria for individual CFD runs (for example RMS $< 10^{-4}$). As shown in Fig. 1, the index i is updated after each simulation. All the solver configurations are recorded regardless of the convergence status. For a converged simulation, post-processing can be conducted to check if the solution meets the initial flow regime assumption and expected accuracy. Specifically, according to the peak value of the Mach number obtained from the simulation, the flow regime will be evaluated again to check whether the simulation model should be altered. If a simulation diverged, the solver configuration needs to be updated to achieve convergence. In the changing process, in order to obtain the sensitivity towards different physics models, only one change is allowed in the solver configuration each time when a new iteration starts. Human intervention is needed to diagnose the problem if the simulation still has convergence issues after successive runs.

As i increases, higher order schemes like the blended scheme which is a weighted average between UDS and CDS, as well as advanced turbulence models like k-ω, Reynolds stress models (RSM) can be applied to improve the simulation quality. By comparing with the validation data, the accuracy of the simulation result can be obtained, which is the criteria for stopping the loop in the post-processing module. Then the program can be started again for the next product design.

3.2. Wet steam simulation

In practice, wet steam commonly acts as a medium formed by a

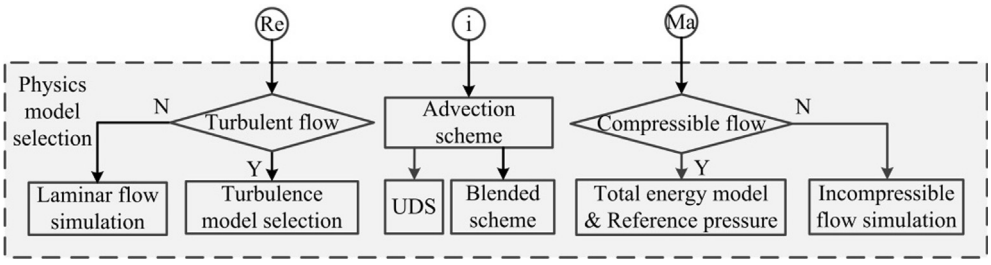

Fig. 2. Physics model selection process.

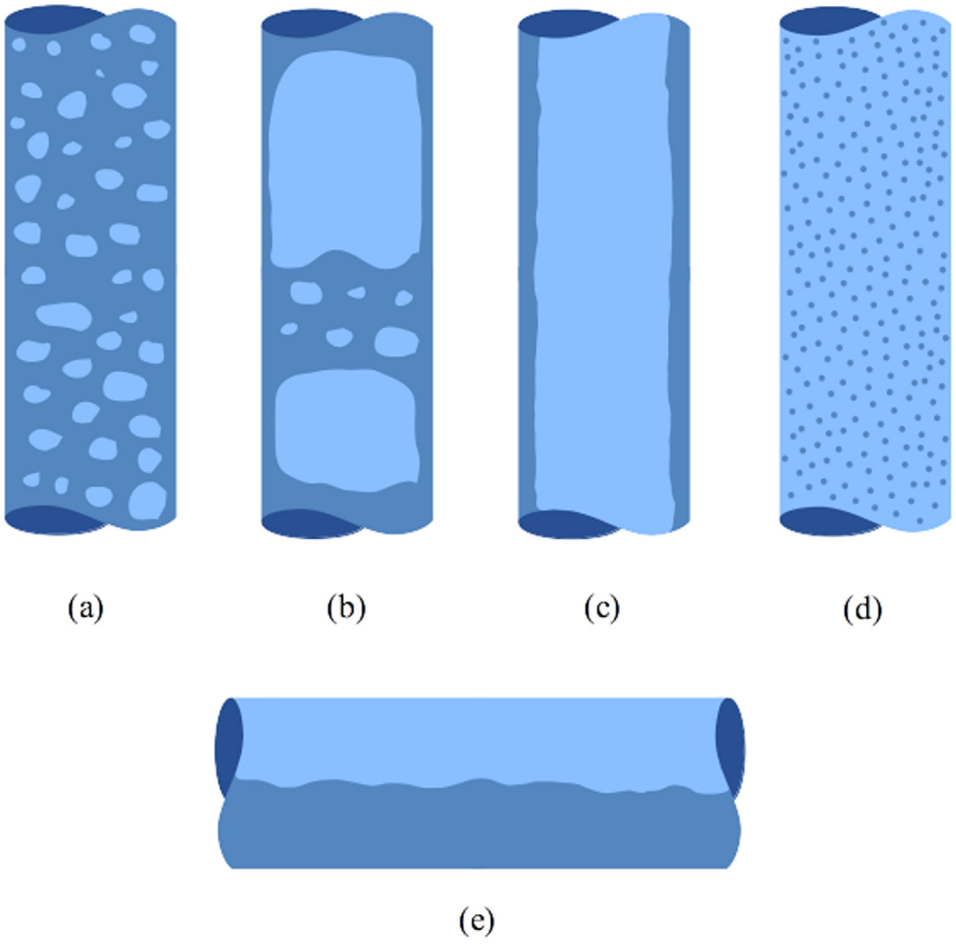

Fig. 3. Various regimes in wet steam: (a) bubbly flow, (b) slug flow, (c) annular flow, (d) mist flow, and (e) stratified flow.

mixture of water vapor and liquid water. This mixture represents two different thermodynamic phases of water in which both phases are at saturation temperature. So the simulation of wet steam falls into multiphase flow simulation. As shown in Fig. 3, wet steam exhibits various flow regimes depending on the relative concentration of the two phases and the flow rate. Each regime requires specific modeling approach. The dominant interactions between the liquid and vapor change their character from one regime to another.

In CFX specifically, there are two main multiphase models, namely the Lagrangian particle tracking model and the Eulerian-Eulerian model [35]. To model the phase change, both equilibrium and non-equilibrium phase change models are available. The equilibrium phase change model is a single fluid, multicomponent phase change model which assumes the phase change occurs instantaneously. Therefore, the two phases have the same temperature. This model is especially suitable for wet steam simulation with a small liquid mass fraction [35]. The non-equilibrium phase change model is available when using Eulerian multiphase and Particle Transport, which allows the two phases to go beyond the saturation dome. The focus of this wet steam simulation module lies in the equilibrium phase change model as an initial investigation attempt.

As shown in Fig. 4, if the wet steam simulation module is entered, the homogeneous binary mixture including water vapor and liquid water is created as a real gas which is treated as the fluid of the flow space. Similar to the dry steam simulation, the fluid domain is abstracted from the product design and meshed with boundary conditions attached. Again, by forward-chaining, the fluid attributes derived from the design are used to estimate the non-dimensional Reynolds number and Mach number to select suitable fluid physics models. Especially, the pre-defined steam quality χ is used as an additional boundary condition of the fluid flow space. The rules for configuring the solver are the same as the intelligent solver functions for dry steam simulation. If a simulation diverged, the solver setup should be tweaked to achieve convergence. Still, only one change is allowed in the solver configuration. If a simulation converged, post-processing will be executed. The physics models will be updated if the flow regime does not match the initial assumption. The program ends when the simulation result meets the accuracy requirement.

3.3. Feature modeling of the intelligent CFD simulation system

As described in the literature review, the fluid physics features and dynamic physics features are capable of selecting the proper physics model to configure the CFD solver and generate the robust simulation model, which should be used to model the intelligent CFD simulation system. As shown in Fig. 5, incorporating CAE boundary features [43], the CFD analysis view [44] is formed in a multi-view feature modeling environment [45] in which the information consistency is maintained properly.

Under the control of CAE boundary features, the fluid domain can be abstracted from the product design and discretized by specific mesh generation methods. Boundary conditions are assigned to the corresponding locations of the fluid domain, forming the fluid flow space. The fluid attributes derived from the design are processed by the equations in the fluid physics feature to evaluate the flow regime. Then, suitable physics models will be selected accordingly based on the rules set by CFD best practices. With the analysis model and fluid flow space, the CFD solver is fully configured and activated to run. After the

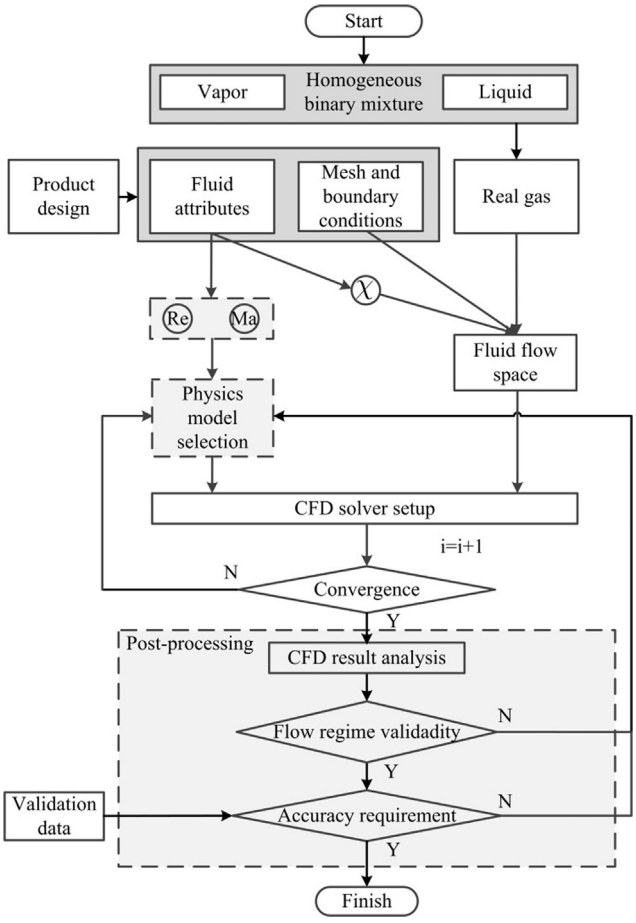

Fig. 4. Wet steam simulation function.

simulation is done, results can be obtained. The convergence status, as well as the analysis model used for the simulation, is recorded as parts of the dynamic physics features. By checking the non-dimensional number, the flow regime can be checked to see if there is any validity issue. The relative error is calculated by comparing with the validation data. The physics models will be re-selected if the simulation diverged or the check of flow regime or simulation accuracy failed. This is to guarantee the generation of the robust simulation model which is defined as the applicable CFD regime and simulation setup template with validated physics conditions and converges into physically reasonable and accurate results [27]. More specifically, the robustness of the simulation model means that the model can be used with confidence on a difficult problem and produce physically reasonable results besides the expected accuracy [46]. The generation of the robust simulation model of the current design indicates the completion of the analysis of this design point. Then, the simulation of next design point can be started by updating the design variables. This process iterates until all the design points are processed.

3.4. Implementation of the intelligent CFD simulation system

A prototype software tool is developed by Python 3 to fulfill the intelligent functions of the proposed CFD simulation system. This tool is designed to invoke ANSYS Workbench and execute different scripts based on the specific product design. It should be highlighted that the logics and settings of the tool are separated, and all possible extensions to be developed in the future can be saved in the settings, making it easily extendable.

The logics of the tool, shown as the pseudocode in Fig. A1 of the appendix, involves iterating the different product designs, calculating the Reynolds number and Mach number, updating ANSYS Workbench scripts accordingly, executing scripts, and finally checking the analysis results. Those actions are specified as follows:

- The calculation of Reynolds number and Mach number is based on the aforementioned Eqs. (1)–(4).
- Updating ANSYS Workbench scripts is to copy a given script as the source template and then change the value of design variable as well as select the correct physics models correspondingly in the copied script. This function is explained by the pseudocode in Fig. A2.
- Executing the script is done by calling the ANSYS command with arguments specifying the Workbench project.

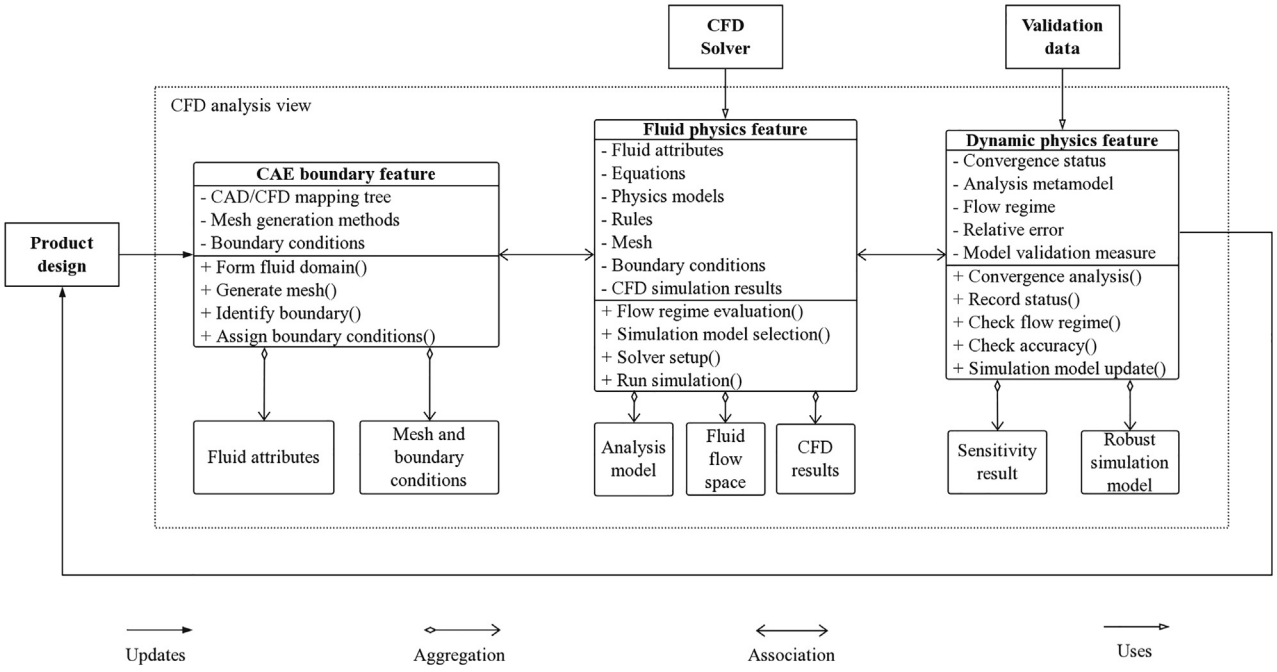

Fig. 5. Feature model of the CFD analysis view.

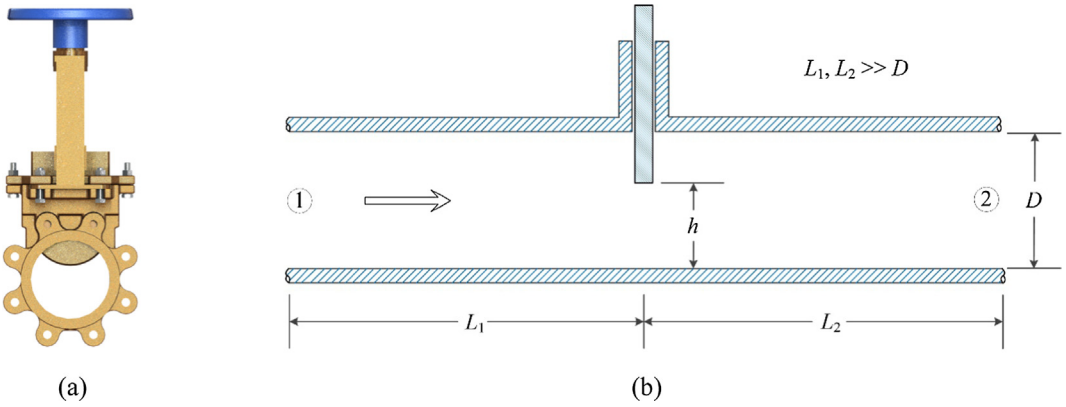

Fig. 6. Disk-type gate valve: (a) product design, and (b) abstracted cross-sectional view.

Table 1
The initial values of parameters.

Parameter	Value	Unit
D	0.2	m
k	1.327	N/A
p	1.013×10^5	Pa
R	461.5	J/(kg·K)
T	373.15	K
$\bar{v}$	40	m/s
μ	1.23×10^{-5}	kg/(m·s)

- Checking the analysis results is achieved by reading the ANSYS-generated comma-separated values (CSV) file to find out the latest values of the output parameter, as shown by the pseudocode in Fig. A3.

As shown in Fig. A4, settings of the tool contain models (the model names and the variable/range by which the corresponding models are determined), values of design variables, validation data to calculate the relative error, and threshold of error to finish looping the models, etc. Such values may change in the future development of other functions, but these changes will not affect the logic to process. The application of this tool will be illustrated by the case study in the next section.

4. Case study

4.1. Analysis of dry steam through a disk-type gate valve

Valves are widely used in the piping system and they are designed to control the flow and condition of a process fluid such as liquid and gas [47]. There are various types of valves available in the market, for example, plug valves, ball valves, butterfly valves, globe valves, gate valves, pinch valves, and diaphragm valves, etc. Among these valves, the gate valve is one of the most common valves in use. And it is selected as the case to validate the effectiveness of the proposed intelligent CFD simulation system because the pressure drop across the valve can be calculated by an empirical equation. Fig. 6(a) shows a typical disk-type gate valve design. The gate is usually driven by a manual wheel or an actuator to move perpendicular to the pipe and finally shut off the flow. The cross-sectional view of the disk-type gate valve is shown in Fig. 6(b) in which h is the opening-distance, L_1 and L_2 are the distance from pressure measurement location 1 and location 2 to the valve gate, respectively. It should be noted that the pressure measurement location must be far away from the gate to allow the flow to be fully developed. Based on experiments, the pressure drop through the valve can be calculated by the following equation [48]:

Table 2
The values of parameters calculated by the system.

Parameter	Value	Unit
a	478	m/s
ρ	0.588	kg/m^3
Re	3.8×10^5	N/A
Ma	0.08	N/A

$$\Delta p = \frac{\rho \bar{v}^2}{2}\left[K + \frac{f(L_1 + L_2)}{D}\right] \tag{5}$$

where K represents the loss coefficient, f indicates the friction factor for fully developed pipe flow. The value of K depends on the opening-distance ratio h/D [42].

The testing case has dry steam flowing into the valve through the inlet at 40 m/s. The pressure at the outlet is 101,325 Pa. The other initial physical parameters are collected in Table 1. Using Eqs. (1)–(4), the proposed system calculates the other parameters which are shown in Table 2. Judging from the Reynolds number and Mach number, the flow is assumed to be incompressible turbulent flow. According to the table for friction factor as a function of Reynolds number and relative roughness [48], the value of f in this case is 0.0138. Both L_1 and L_2 are ten times the inner diameter D. Using Eq. (5), the pressure drop corresponding to different opening-distance ratio can be calculated, which is shown in Table 3.

The starting case, which is used as the template for all the following cases to assign boundary conditions and create output parameters, uses the opening-distance ratio of 0.8. L_1 and L_2 in this testing case should be 2 m. So the length of the fluid domain is 5 m as shown in Fig. 7(a) in which the boundary conditions are also specified. The mesh generation and refinement method is the same for all the cases with different opening-distance ratios to purely test the effect of physics models. Fig. 7(b) shows a section of the meshed domain which has the inflation layer along the wall boundary and refinement made adjacent to the gate region.

When i = 1, the intelligent CFD simulation system selects the Isothermal as the heat transfer model because the derived Mach number

Table 3
Empirical pressure drop corresponding to opening-distance ratio [48].

h/D	K	Δp_1 (Pa)
0.8	0.17	209.8
0.7	0.44	336.81
0.6	0.98	590.82
0.5	2.1	1117.67
0.4	4.6	2293.67
0.3	10	4833.83

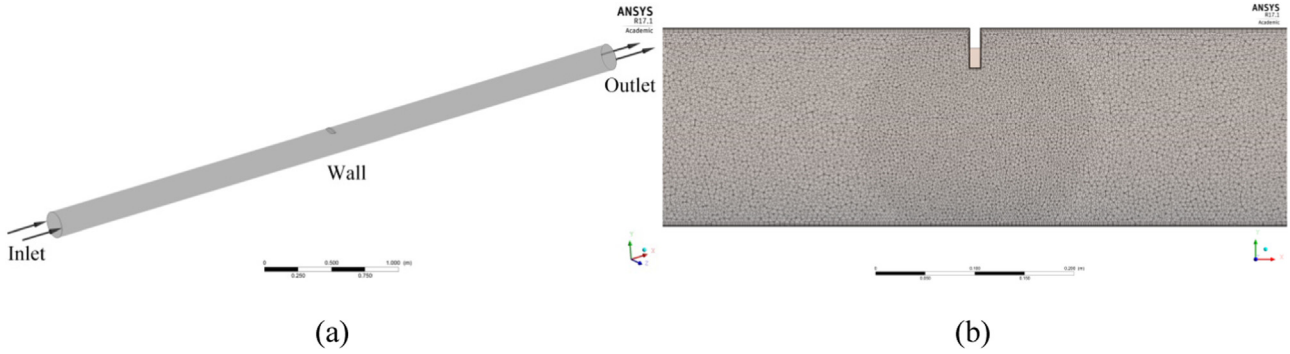

Fig. 7. Fluid domain of the case $h/D = 0.8$: (a) boundaries, and (b) partial view of mesh.

Table 4
Pressure drop, relative error, Mach number, total iteration and robust combination of models.

h/D	Δp_2 (Pa)	δ_1 (%)	Ma	i	Heat transfer	Advection scheme	Turbulence model
0.8	219.88	4.81	0.11	1	Isothermal	Upwind	k-epsilon
0.7	336.45	0.11	0.13	1	Isothermal	Upwind	k-epsilon
0.6	577.95	2.18	0.16	1	Isothermal	Upwind	k-epsilon
0.5	1166.44	4.36	0.21	3	Isothermal	High Resolution	SST k-omega
0.4	2284.49	0.4	0.26	4	Isothermal	High Resolution	SSG Reynolds Stress
0.3	4921.32	1.81	0.35	4	Total Energy	High Resolution	SST k-omega

is less than 0.3. The advection scheme selected is the lower order Upwind. Because the Reynolds number is much bigger than 4000, a turbulence model is needed and k-epsilon is chosen in this iteration. To control the convergence in the solver run, the residual target is set to be 2×10^{-5} and the maximum number of iterations is 300. After the simulation is done, by comparing with the validation result shown in Table 3, the relative error of the simulation result can be obtained. The stopping criterion for this case is the relative error should be less than 5%. Because the default combination of physical models is capable of producing accurate simulation result which has a relative error of 4.81%, the loop stops for the first design point and the loop for the next design point starts. At last, the intelligent CFD simulation system finishes the simulation of all the design points, and the results of pressure drop, relative error, Mach number, total iteration and the robust combination of physics models are collected in Table 4.

For the first three design points, it is obvious that the default combination of physics models is able to generate acceptable simulation results. Therefore, only one iteration is needed in each loop. As the gate approaches the bottom of the valve, the velocity of the steam will increase, which will create a larger recirculation zone. This is the reason why High Resolution and advanced turbulence models are needed for design point 4 and 5. After the initial simulation of the last design point, the Mach number is found to be bigger than 0.3. So the heat transfer model is switched to Total Energy when i = 2. Finally, there are 4 iterations used to generate the robust simulation model for the last design point. The detailed investigation of the first and last design point is selected to show why the changes in the physics models are necessary. Fig. 8(a) and (b) shows the streamlines when the opening-distance ratio equals 0.8 and 0.3, respectively. Clearly, in the last design point, the flow velocity is much higher and the flow reattachment point is much further in the downstream. This also confirms the necessity to use a long enough domain. The contours of density at the valve gate opening cross-section corresponding to those two design points are shown in Fig. 9. As can be seen, the change in density is negligible in the first design point. When the flow area is reduced dramatically in the last design point, the density varies by 7%, which needs the correct heat transfer model to simulate the compressible flow.

4.2. Analysis of wet steam in pipe reducer

As shown in Fig. 10(a), the reducer is used to reduce the pipe size to a smaller bore in a piping system. This corresponds to the contraction in fluid mechanics [42], which is shown in Fig. 10(b). When wet steam flows through a reducer that is not well insulated, the steam will condensate, which affects the steam quality. This case study is selected to test the wet steam simulation function of the proposed intelligent CFD simulation system.

The boundaries of the fluid domain are shown in Fig. 11(a), which have the conditions specified as follows:

- Inlet: flow velocity at 1 m/s, steam quality at 80%, temperature at 373.15 K;
- Outlet: relative pressure at - 40 Pa;
- Wall: no-slip wall at 273.15 K;
- Reference pressure: 101,325 Pa.

As shown in Fig. 11(b), the mesh is generated with inflation layer along the wall boundary and refinement made around the area where the two sections connect. The gas phase named H2ORKv and the liquid phase named H2ORKl are used to create the homogeneous binary mixture which is treated as the fluid of the domain. The saturation properties of this mixture are set to Redlich Kwong, which is a real gas model in CFX [35]. The Reynolds number and Mach number derived from the fluid attributes indicate that the flow is incompressible turbulent flow.

After several iterations, the robust simulation model with accurate results is obtained. As shown in Fig. 12(a), the streamlines exhibit the vena contracta [49] which is a typical flow behavior after a nozzle. The mass fraction of the H2ORKv is exactly the steam quality which is shown in Fig. 12(b). The steam quality is calculated by the intelligent CFD simulation system and the empirical method [50] at the connection area and at the outlet. The results are collected in Table 5. It is obvious that the relative error between the two methods is quite small at both locations. The error at the outlet is slightly bigger because of the entry effects for the second stage calculation. These results demonstrate the successful implementation of the wet steam simulation function in the current intelligent CFD simulation system.

4.3. Discussion of applicability

The disk-type gate valve and pipe reducer used in this case study are all typical products in a piping system, which are relatively simple. The advantage of using these simple cases is that the validation data is easy to obtain through calculation. As a result, the effectiveness of the

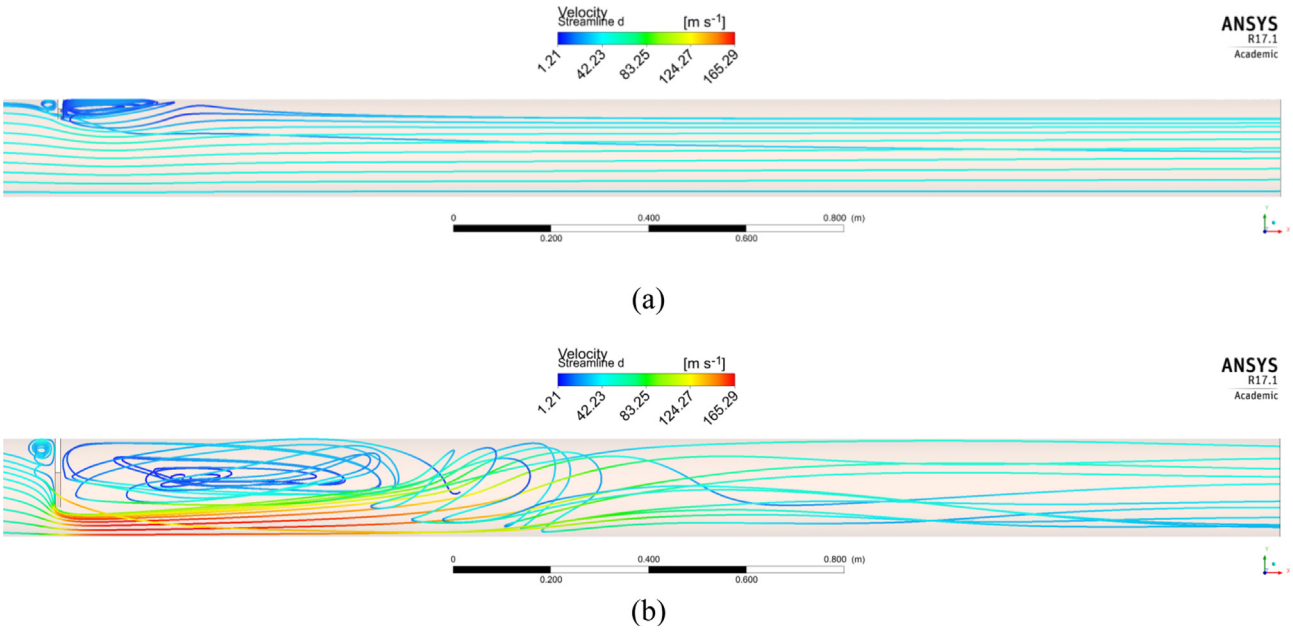

Fig. 8. Streamlines across the domain: (a) $h/D = 0.8$, and (b) $h/D = 0.3$.

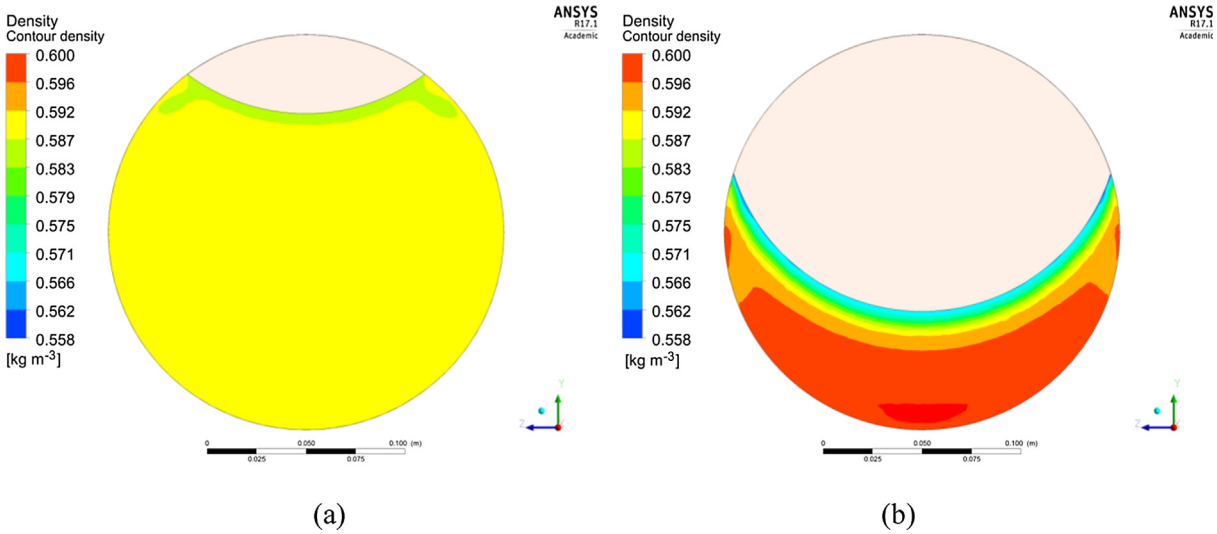

Fig. 9. Contour of density at the valve gate opening cross-section: (a) $h/D = 0.8$, and (b) $h/D = 0.3$.

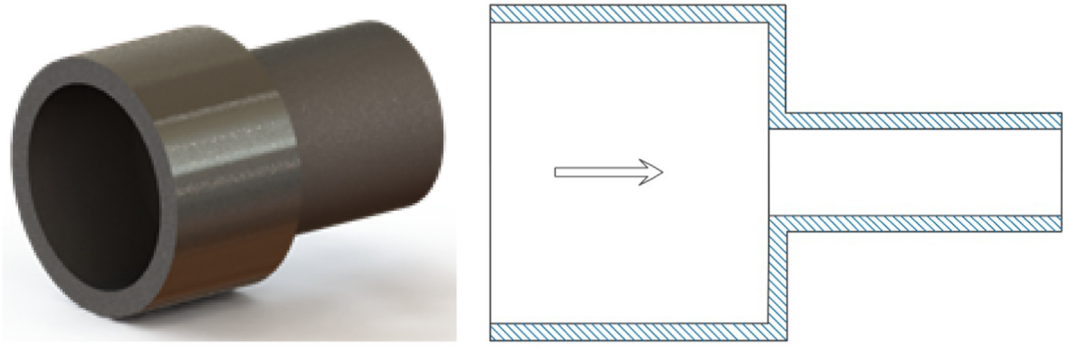

Fig. 10. (a) Pipe reducer, and (b) cross-sectional view of a contraction.

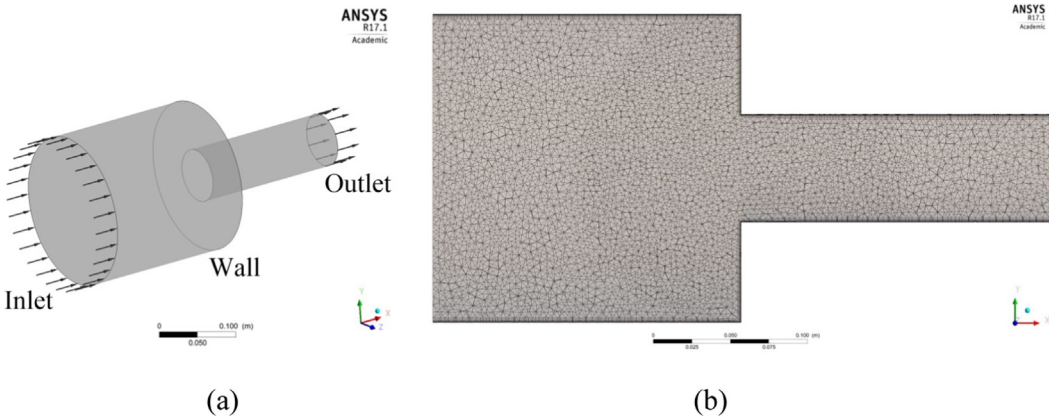

Fig. 11. Fluid domain of the pipe contraction: (a) boundaries, and (b) mesh.

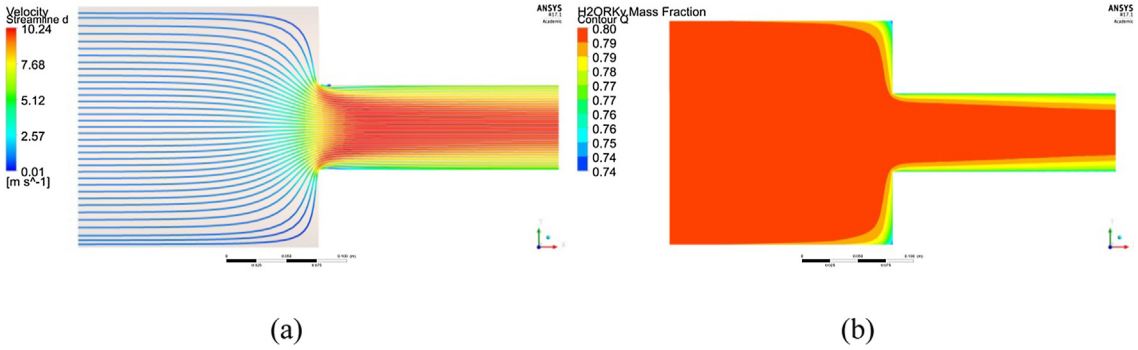

Fig. 12. Simulation result: (a) streamlines, and (b) contour of steam quality.

Table 5
Steam quality comparison.

Location	Intelligent system	Empirical calculation	δ_2
Inlet	80%	80%	–
Connection	79.7%	79.9%	0.25%
Outlet	78.6%	79.6%	1.26%

proposed intelligent CFD simulation system can be verified. However, it should be noted that the application of the system is not limited to simple cases only. In CFD-based optimization which is the other application scenario of CFD, this system can be used to efficiently generate convincible results for algorithms to improve the design of more complex product. For example, results obtained from the proposed system can be used to optimize the design of outflow control device and improve the heating efficiency in steam assisted gravity drainage (SAGD) [27].

Currently, the system is established in the field of steam simulation. Because the rules in this system are based on CFD best practices, this system can be used to simulate other kinds of gas or liquid flow by changing the relevant knowledge base of the specific fluid. Therefore, the proposed system is an extendable system which is also capable of complex simulations.

5. Conclusions

The development of an intelligent CFD simulation system is explored in this paper to tackle the problems existing in the CFD batch simulation. This system is able to simulate both dry and wet steam and it is built on the rules derived from physical knowledge and CFD best practices, making it capable of analyzing specific product design, selecting suitable physics models, and generating the corresponding robust simulation model. In order to realize the system, fluid physics features are used to model the data processing, physics model selection, and simulation execution while fluid dynamic physics features are applied to model the convergence analysis, post-processing, and generation of the robust simulation model. Along with CAE boundary features, the CFD analysis view is fulfilled in the product development process. The prototype is developed by Python 3 with separated logics and settings, making it easy to be extended. The case study of a disk-type gate valve is used to show the function of the whole system with various opening-distance ratios. It is shown that this intelligent system can generate the robust simulation models for each operating condition and output accurate results. In this process, the dynamic physics features support the analysis of the sensitivity toward different physics models, which is useful information for the simulation of other operating conditions. The pipe reducer case shows that the wet steam simulation module can capture the phase change and calculate the steam quality correctly. Overall, this paper provides a generic way for intelligent CFD simulation system development which can be applied to other complex scenarios by changing the relevant knowledge base.

At present, the intelligent CFD simulation system is built for steady flow simulation, and the wet steam module uses the equilibrium phase change model which is suitable for the specific mist flow. In the future, the function for transient simulation can be added to enhance its applicability. Other wet steam models will be investigated to facilitate the simulation of more complicated wet steam regimes. For complex simulations, the performance of the intelligent CFD simulation system can be further increased by running it in a high-performance cloud computing environment to dramatically reduce the computation time.

Acknowledgment

The authors would like to thank Natural Sciences and Engineering Research Council of Canada (NSERC), RGL Reservoir Management, China Scholarship Council (CSC), University of Alberta and Alberta Innovates Technology Futures (AITF) for the financial support.

Appendix A

See Figs. A1–A4.

```
Algorithm 1: intelligent CFD simulation system main()
  Input:
    DEFAULT_MODELS:     the default model series
    u, p, T, r, k, v:   parameters for calculation of "re" and "ma"
    h:                  value of "h" of a specific iteration
    D                   value of "D" of a specific iteration
    STANDARD_DP:        pressure drop value for validation
    THRESHOLD:          the value to determine if result is good enough
  Output:
    p2:                 pressure drop from simulation
    e:                  relative error of the simulation result
    ma:                 value of "ma" of the iteration

FOR h in {160, 140, 120, 100, 80, 60}
  models = copy(DEFAULT_MODELS)
  re, ma = Calculate(u, p, T, r, k, v, h / D)
  IF re < 4000
    FOR model in models
      model[TURBULENCE MODEL] = "None"
    ENDFOR
  ENDIF
  FOR model in models
    # Determine whether and how to use "Total energy"
    IF ma > 0.3 AND model[HEAT TRANSFER MODEL] != "Total energy"
      IF has_pervious(model)
        new_model = copy(previous(model))
      ENDIF
      models.add_new_before(new_model, model)         # Run "Total energy" with the previous model
      FOR model2 in sub_of_array_from(models, new_model)
        model2[HEAT TRANSFER MODEL] = "Total energy"  # Set the rest models to use "Total energy"
      ENDFOR
    ENDIF

    # Now "new_model" is "model"
    model_wbjn_script = update_script_with_new_model(h, model)    # Algorithm 2

    # Call ANSYS workbench to run the wbjn script
    results = call(ANSYS_WB, model_wbjn_script)

    # Check calculation results
    p2, ma = check_results(results)                               # Algorithm 3

    # Finsh up this round for the "h"
    e = |p2 - STANDARD_DP| / STANDARD_DP
    Output: p2, e, ma
    IF e < THRESHOLD                                  # Do not go further here
      BREAK
    ENDIF
  ENDFOR
ENDFOR
```

Fig. A1. Pseudocode of algorithm 1.

```
Algorithm 2: update ANSYS “wbjn” script with specific models
  Input:
    h:                  the value of "h"
    models:             the models for this iteration
  Output:
    model_wbjn_script:  the new script updated with models

# Heat transfer model is changed by choosing the right script template
IF models[HEAT TRANSFER MODEL] == "Total energy"
  model_wbjn_script = copy(wbjn_template_total_energy)
ELSE
  model_wbjn_script = copy(wbjn_template_iso_thermal)
ENDIF

FOR line in lines(model_wbjn_script)
  # Change value of "h"
  IF starts(line, "Expression=")
    replace(line, "\d+", h)                          # regular expression to match numbers
  ENDIF

  # Change value of models, except heat transfer model
  model = remove_trailing_symbol(line, ":")          # keywords of models all have a trailing colon
  IF model != "HEAT TRANSFER MODEL" and model in models
    value_line = find_next_line_starts("Option =")
    replace(value_line, model[model])
  ENDIF
ENDFOR

Output: model_wbjn_script
```

Fig. A2. Pseudocode of algorithm 2.

```
Algorithm 3: check ANSYS calculation results
  Input:
    results:            the result log file generated by ANSYS
  Output:
    p2:                 pressure drop from simulation
    ma:                 value of "ma" of the iteration

result = create_empty_set()
FOR line in lines(results)
  # skip empty lines and comments
  IF is_empty(line) or starts(line, "#")
    continue
  ENDIF

  # keys - variable names
  IF starts(line, " ")
    keys = split_by(line, ",")
    result.add_keys(keys)
    continue
  ENDIF

  # values - variable values
  values = split_by(line, ",")
  result.add_values(values)
ENDFOR

Output: result[p2], result[ma]
```

Fig. A3. Pseudocode of algorithm 3.

```
{
  "models": [
    {
      "ADVECTION SCHEME":           "Upwind",
      "TURBULENCE MODEL":           "k epsilon",
      "HEAT TRANSFER MODEL":        "Isothermal"
    },
    {
      "ADVECTION SCHEME":           "High Resolution",
      "TURBULENCE MODEL":           "k epsilon",
      "HEAT TRANSFER MODEL":        "Isothermal"
    },
    {
      "ADVECTION SCHEME":           "High Resolution",
      "TURBULENCE MODEL":           "SST",
      "HEAT TRANSFER MODEL":        "Isothermal"
    },
    {
      "ADVECTION SCHEME":           "High Resolution",
      "TURBULENCE MODEL":           "SSG Reynolds Stress",
      "HEAT TRANSFER MODEL":        "Isothermal"
    }
  ],
  "TURBULENCE MODEL": {
    "None":                   ["-Infinity", 4000]
  },
  "HEAT TRANSFER MODEL": {
      "Total energy":         [0.3,         "Infinity"]
  },
  "ansys_wb_path":    "C:\\ANSYS\\...",
  "root":             "D:\\Downloads",
  "source":           "valve_s",
  "h":                [160, 140, 120, 100, 80, 60],
  "parameters": {
    "u":  1.23e-5,
    "p":  1.013e5,
    "t":  373.15,
    "r":  461.5,
    "k":  1.327,
    "v":  40
  },
  "standard_dp": [
    209.8,
    336.81,
    590.82,
    1117.67,
    2293.67,
    4833.83
  ],
  "threshold":        0.05,
  "result_path":      "result.log"
}
```

Fig. A4. Settings file.

References

[1] O. Tonomura, S. Tanaka, M. Noda, M. Kano, S. Hasebe, I. Hashimoto, CFD-based optimal design of manifold in plate-fin microdevices, Chem. Eng. J. 101 (2004) 397–402, https://doi.org/10.1016/j.cej.2003.10.022.

[2] M.H.A. Bonte, A.H. van den Boogaard, J. Huétink, An optimisation strategy for industrial metal forming processes, Struct. Multidiscip. Optim. 35 (2008) 571–586, https://doi.org/10.1007/s00158-007-0206-3.

[3] D.C. Montgomery, Design and Analysis of Experiments, eighth ed., Wiley Global Education, 2012.

[4] C. Grosan, A. Abraham, Rule-based expert systems, Intell. Syst. Springer, Berlin, Heidelberg, 2011, pp. 149–185, , https://doi.org/10.1007/978-3-642-21004-4_7.

[5] C.G. Yin, Y.S. Ma, Parametric feature constraint modeling and mapping in product development, Adv. Eng. Informat. 26 (2012) 539–552, https://doi.org/10.1016/j.aei.2012.02.010.

[6] J.J. Shah, Conceptual development of form features and feature modelers, Res. Eng. Des. 2 (1991) 93–108, https://doi.org/10.1007/BF01579254.

[7] J. Liu, L. Cheng, A.C. To, Arbitrary void feature control in level set topology optimization, Comput. Methods Appl. Mech. Eng. 324 (2017) 595–618, https://doi.org/10.1016/j.cma.2017.06.021.

[8] J.J. Shah, M. Mäntylä, Parametric and Feature-based CAD/CAM: Concepts, Techniques, and Applications, Wiley, 1995.

[9] A.J.P. Gomes, J.G. Teixeira, Form feature modelling in a hybrid CSG/BRep scheme, Comput. Graph. 15 (1991) 217–229, https://doi.org/10.1016/0097-8493(91)90075-S.

[10] E.M. Sanfilippo, S. Borgo, What are features? An ontology-based review of the literature, Comput. Des. 80 (2016) 9–18, https://doi.org/10.1016/j.cad.2016.07.001.

[11] D. Mun, S. Han, J. Kim, Y. Oh, A set of standard modeling commands for the history-based parametric approach, Comput. Des. 35 (2003) 1171–1179, https://doi.org/10.1016/S0010-4485(03)00022-8.

[12] Z. Cheng, Y. Ma, A functional feature modeling method, Adv. Eng. Informat. 33 (2017) 1–15, https://doi.org/10.1016/J.AEI.2017.04.003.

[13] C.K. Chan, S.T. Tan, Generating assembly features onto split solid models, Comput. Des. 35 (2003) 1315–1336, https://doi.org/10.1016/S0010-4485(03)00062-9.

[14] M.R. Khabbazi, J. Wikander, M. Onori, A. Maffei, Object-oriented design of product assembly feature data requirements in advanced assembly planning, Assem. Autom. 38 (2018) 97–112, https://doi.org/10.1108/AA-07-2016-084.

[15] X. Yan, K. Yamazaki, J. Liu, Recognition of machining features and feature topologies from NC programs, Comput. Des. 32 (2000) 605–616, https://doi.org/10.1016/S0010-4485(00)00045-2.

[16] J. Liu, Y.-S. Ma, 3D level-set topology optimization: a machining feature-based approach, Struct. Multidiscip. Optim. 52 (2015) 563–582, https://doi.org/10.1007/s00158-015-1263-7.

[17] J. Liu, A.C. To, Topology optimization for hybrid additive-subtractive manufacturing, Struct. Multidiscip. Optim. 55 (2017) 1281–1299, https://doi.org/10.1007/s00158-016-1565-4.

[18] J. Liu, Z. Cheng, Y. Ma, Product design-optimization integration via associative optimization feature modeling, Adv. Eng. Informat. 30 (2016) 713–727, https://doi.org/10.1016/j.aei.2016.09.004.

[19] Y.-S. Ma, G. Chen, G. Thimm, Paradigm shift: unified and associative feature-based concurrent and collaborative engineering, J. Intell. Manuf. 19 (2008) 625–641, https://doi.org/10.1007/s10845-008-0128-y.

[20] S.-H. Tang, G. Chen, Y.-S. Ma, Fundamental concepts of generic features, Semant. Model. Interoperability Prod. Process Eng. Springer, London, 2013, pp. 89–115, , https://doi.org/10.1007/978-1-4471-5073-2_4.

[21] S. Gao, W. Zhao, H. Lin, F. Yang, X. Chen, Feature suppression based CAD mesh model simplification, Comput. Des. 42 (2010) 1178–1188, https://doi.org/10.1016/J.CAD.2010.05.010.

[22] S.H. Lee, A CAD–CAE integration approach using feature-based multi-resolution

and multi-abstraction modelling techniques, Comput. Des. 37 (2005) 941–955, https://doi.org/10.1016/j.cad.2004.09.021.
[23] O. Hamri, J.-C. Léon, F. Giannini, B. Falcidieno, Software environment for CAD/CAE integration, Adv. Eng. Softw. 41 (2010) 1211–1222, https://doi.org/10.1016/J.ADVENGSOFT.2010.07.003.
[24] Y.-M. Deng, Y.C. Lam, S.B. Tor, G.A. Britton, A CAD-CAE integrated injection molding design system, Eng. Comput. 18 (2002) 80–92, https://doi.org/10.1007/s003660200007.
[25] Z. Xia, Q. Wang, Y. Wang, C. Yu, A CAD/CAE incorporate software framework using a unified representation architecture, Adv. Eng. Softw. 87 (2015) 68–85, https://doi.org/10.1016/J.ADVENGSOFT.2015.05.005.
[26] M.S. Shephard, M.W. Beall, R.M. O'Bara, B.E. Webster, Toward simulation-based design, Finite Elem. Anal. Des. 40 (2004) 1575–1598, https://doi.org/10.1016/j.finel.2003.11.004.
[27] L. Li, C.F. Lange, Y. Ma, Association of design and computational fluid dynamics simulation intent in flow control product optimization, Proc. Inst. Mech. Eng. Part B J. Eng. Manuf. 954405417697352 (2017), https://doi.org/10.1177/0954405417697352.
[28] M. García, J. Duque, P. Boulanger, P. Figueroa, Computational steering of CFD simulations using a grid computing environment, Int. J. Interact. Des. Manuf. 9 (2015) 235–245, https://doi.org/10.1007/s12008-014-0236-1.
[29] A. Rezaeiha, I. Kalkman, B. Blocken, CFD simulation of a vertical axis wind turbine operating at a moderate tip speed ratio: guidelines for minimum domain size and azimuthal increment, Renew. Energy 107 (2017) 373–385, https://doi.org/10.1016/J.RENENE.2017.02.006.
[30] Y. Cao, T. Tamura, Large-eddy simulations of flow past a square cylinder using structured and unstructured grids, Comput. Fluids 137 (2016) 36–54, https://doi.org/10.1016/J.COMPFLUID.2016.07.013.
[31] T.S. Phillips, J.M. Derlaga, C.J. Roy, J. Borggaard, Error transport equation boundary conditions for the Euler and Navier-Stokes equations, J. Comput. Phys. 330 (2017) 46–64, https://doi.org/10.1016/J.JCP.2016.11.002.
[32] A.Q. Ahmed, S. Gao, A.K. Kareem, Energy saving and indoor thermal comfort evaluation using a novel local exhaust ventilation system for office rooms, Appl. Therm. Eng. 110 (2017) 821–834, https://doi.org/10.1016/J.APPLTHERMALENG.2016.08.217.
[33] K.-J. Bathe, H. Zhang, A mesh adaptivity procedure for CFD and fluid-structure interactions, Comput. Struct. 87 (2009) 604–617, https://doi.org/10.1016/j.compstruc.2009.01.017.
[34] H.K. Versteeg, W. Malalasekera, An Introduction to Computational Fluid Dynamics: The Finite Volume Method, Pearson Education, Harlow, England, 2007.
[35] ANSYS CFX-Solver Modeling Guide, 2013.
[36] F. Murena, G. Favale, S. Vardoulakis, E. Solazzo, Modelling dispersion of traffic pollution in a deep street canyon: application of CFD and operational models, Atmos. Environ. 43 (2009) 2303–2311, https://doi.org/10.1016/J.ATMOSENV.2009.01.038.
[37] M. Casey, T. Wintergerste, Best Practice Guidelines, Version 1.0., ERCOFTAC, 2000.
[38] J.H. Ferziger, M. Peric, Computational Methods for Fluid Dynamics, Springer Berlin Heidelberg, Berlin, Heidelberg, 2002 http://doi.org/10.1007/978-3-642-56026-2.
[39] Y. He, J. Su, The almost unconditional convergence of the Euler implicit/explicit scheme for the three dimensional nonstationary Navier-Stokes equations, Discret. Contin. Dyn. Syst. - Ser. B 22 (2017) 3421–3438, https://doi.org/10.3934/dcdsb.2017173.
[40] F.J. Salvador, J. Gimeno, J.M. Pastor, P. Martí-Aldaraví, Effect of turbulence model and inlet boundary condition on the Diesel spray behavior simulated by an Eulerian Spray Atomization (ESA) model, Int. J. Multiph. Flow 65 (2014) 108–116, https://doi.org/10.1016/J.IJMULTIPHASEFLOW.2014.06.003.
[41] W.L. Oberkampf, T.G. Trucano, Verification and validation in computational fluid dynamics, Prog. Aerosp. Sci. 38 (2002) 209–272, https://doi.org/10.1016/S0376-0421(02)00005-2.
[42] F.M. White, Fluid Mechanics, seventh ed., McGraw-Hill, New York, NY, 2011.
[43] L. Li, Y. Ma, CAD/CAE associative features for cyclic fluid control effect modeling, Comput. Aided. Des. Appl. 13 (2016) 208–220, https://doi.org/10.1080/16864360.2015.1084190.
[44] L. Li, C.F. Lange, Y. Ma, Artificial intelligence aided CFD analysis regime validation and selection in feature-based cyclic CAD/CFD interaction process, Comput. Aided. Des. Appl. 15 (2018) 643–652, https://doi.org/10.1080/16864360.2018.1441230.
[45] W.F. Bronsvoort, A. Noort, Multiple-view feature modelling for integral product development, Comput. Des. 36 (2004) 929–946, https://doi.org/10.1016/j.cad.2003.09.008.
[46] V. Venkatakrishnan, M.D. Salas, S.R. Chakravarthy, Barriers and Challenges in Computational Fluid Dynamics, Springer Netherlands, Dordrecht, 1998 http://doi.org/10.1007/978-94-011-5169-6.
[47] P.L. Skousen, Valve Handbook, McGraw-Hill, 2011.
[48] R.D. Blevins, Applied Fluid Dynamics Handbook, Krieger Pub. Co, Malabar, 1992.
[49] M.S. Shah, J.B. Joshi, A.S. Kalsi, C.S.R. Prasad, D.S. Shukla, Analysis of flow through an orifice meter: CFD simulation, Chem. Eng. Sci. 71 (2012) 300–309, https://doi.org/10.1016/J.CES.2011.11.022.
[50] Y.A. Çengel, A.J. Ghajar, Heat and Mass Transfer: Fundamentals and Applications, McGraw-Hill, 2014.