User Guide of BIOTC

Song-Charng Kong
Department of Mechanical Engineering, Iowa State University, United States
Updated on January 1, 2013

BIOTC is an open-source computer code developed at Iowa State University under the support of the NSF Iowa EPSCoR Project (EPS-1101284). The code is available to the research community upon request. Please send a request to Dr. Song-Charng Kong (kong@iastate.edu) so that a user list can be maintained. Once a request is granted, the user will be provided with the access to the code. Users will be notified when new updates are available in the future.

BIOTC is a computational fluid dynamics (CFD) code for simulating BIOmass Thermochemical Conversion in gasifiers, pyrolysis reactors, and combustors. In this document, mathematical formulations of the governing equations and numerical schemes are presented. Much emphasis has been devoted to the coding strategy to give users a clear picture of the computer program. A tutorial case is also provided.

Numerical schemes

Multi-fluid model

A multi-fluid model that can simulate one gas phase and an arbitrary number of solid phases is adopted to characterize in-bed hydrodynamics. In this multi-fluid model, both gas and solid phases are treated as inter-penetrating continua and conservative equations for mass, momentum, energy and species are established. The conservative equations in multi-fluid model are presented as follows.

The continuity equation to describe the gas phase volume fraction is

$$\frac{\partial \alpha_g \rho_g}{\partial t} + \nabla \cdot (\alpha_g \rho_g \boldsymbol{U}_g) = R_g \tag{1}$$

where α_g, ρ_g and $\boldsymbol{U}_g$ are the gas phase volume fraction, material density, and velocity vector, respectively. R_g is the mass transfer source term due to pyrolysis reactions or water evaporation. The ideal gas law is used to evaluate the gas material density.

The momentum conservative equation is

$$\frac{\partial (\alpha_g \rho_g \boldsymbol{U}_g)}{\partial t} + \nabla \cdot (\alpha_g \rho_g \boldsymbol{U}_g \boldsymbol{U}_g) =$$
$$\nabla \cdot (\alpha_g \boldsymbol{\tau}_g) - \alpha_g \nabla p + \sum_{m=1}^{M} \beta_{gsm}(\boldsymbol{U}_{sm} - \boldsymbol{U}_g) + \sum_{m=1}^{M} \boldsymbol{\psi}_{gsm} + \alpha_g \rho_g \boldsymbol{g} \tag{2}$$

where $\boldsymbol{\tau}_g$ is the stress tensor, β_{gsm} is the gas-solid inter-phase momentum exchange coefficient between gas and solid phase m, $\boldsymbol{\psi}_{gsm}$ is the gas-solid momentum exchange due to mass transfer, and $\boldsymbol{g}$ is the gravity vector. $\boldsymbol{\tau}_g$ is expressed as a Newtonian form

$$\tau_g = 2\alpha_g \mu_g \boldsymbol{D}_g + \alpha_g \lambda_g tr(\boldsymbol{D}_g)\boldsymbol{I} \tag{3}$$

where $\boldsymbol{I}$ is the identity tensor and $\boldsymbol{D}_g$ is the gas phase strain tensor. μ_g and λ_g are gas phase dynamic and bulk viscosity, respectively. In this study, only gas-solid drag force is included into β_{gsm} since drag force has been accepted as the dominate contribution over other effects such as virtual mass force, lift force, etc. In the tutorial case, the widely-used Gidaspow type drag force is employed.

$$\beta_{gsm} = \begin{cases} \dfrac{3}{4}C_D \dfrac{\rho_g \alpha_g \alpha_{sm} |\boldsymbol{U}_g - \boldsymbol{U}_{sm}|}{d_{sm}} \alpha_g^{-2.65} & \alpha_g \geq 0.8 \\[4mm] 150\dfrac{\alpha_{sm}(1-\alpha_g)\mu_g}{\alpha_g d_{sm}^2} + \dfrac{7}{4}\dfrac{\rho_g \alpha_{sm} |\boldsymbol{U}_g - \boldsymbol{U}_{sm}|}{d_{sm}} & \alpha_g < 0.8 \end{cases}$$

$$C_D = \begin{cases} \dfrac{24}{\mathrm{Re}}(1+0.15\,\mathrm{Re}^{0.687}) & \mathrm{Re} < 1000 \\[3mm] 0.44 & \mathrm{Re} \geq 1000 \end{cases} \tag{4}$$

$$\mathrm{Re} = \frac{\rho_g d_{sm} |\boldsymbol{U}_g - \boldsymbol{U}_{sm}|}{\mu_g}$$

In the above, α_{sm}, d_{sm} and $\boldsymbol{U}_{sm}$ are the solid volume fraction, particle diameter, and velocity vector in solid phase m, respectively. $\boldsymbol{\psi}_{gsm}$ is described as follows.

$$\boldsymbol{\psi}_{gsm} = R_{gsm}[\zeta \boldsymbol{U}_{sm} + (1-\zeta)\boldsymbol{U}_g]$$
$$\zeta = \begin{cases} 0 & R_{gsm} < 0 \\ 1 & R_{gsm} \geq 0 \end{cases} \tag{5}$$

The energy equation is

$$\frac{\partial(\alpha_g \rho_g C_{pg} T_g)}{\partial t} + \nabla \cdot (\alpha_g \rho_g C_{pg} T_g \boldsymbol{U}_g) = \nabla \cdot (\alpha_g \boldsymbol{q}_g) + \sum_{m=1}^{M} h_{gsm}(T_{sm} - T_g) + \Delta H_g \tag{6}$$

where T_g, C_{pg} and $\boldsymbol{q}_g$ are gas phase temperature, heat capacity, and conductive heat flux, respectively. Fourier's law is applied to the conductive heat flux.

$$\boldsymbol{q}_g = \alpha_g \kappa_g \nabla T_g \tag{7}$$

where κ_g is the gas phase thermal conductivity. h_{gsm} is the gas-solid heat transfer coefficient, similar to gas-solid momentum transfer coefficient, and ΔH_g is the heat resulting from gas phase reactions. In the tutorial case, Ranz-Marshall gas-solid heat transfer is applied to h_{gsm}.

$$h_{gsm} = \frac{6\alpha_{sm}\kappa_g Nu}{d_{sm}^2}$$

$$Nu = 2 + 0.6\,\mathrm{Re}^{1/2}\,\mathrm{Pr}^{1/3} \tag{8}$$

$$\mathrm{Pr} = \frac{C_{pg}\mu_g}{\kappa_g}$$

The mass species equation is

$$\frac{\partial \alpha_g \rho_g Y_{gk}}{\partial t} + \nabla \cdot (\alpha_g \rho_g Y_{gk} U_g) = R_{gk} \tag{9}$$

where Y_{gk} denotes the mass fraction of kth specie in the gas phase.

The continuity and momentum equations of solid phase m are as follows.

$$\frac{\partial \alpha_{sm}\rho_{sm}}{\partial t} + \nabla \cdot (\alpha_{sm}\rho_{sm}U_{sm}) = R_{sm} \tag{10}$$

$$\frac{\partial(\alpha_{sm}\rho_{sm}U_{sm})}{\partial t} + \nabla \cdot (\alpha_{sm}\rho_{sm}U_{sm}U_{sm}) =$$

$$\nabla \cdot (\alpha_{sm}\tau_{sm}) - \alpha_{sm}\nabla p + \beta_{gsm}(U_{sm} - U_g) + \sum_{l=1,l\neq m}^{M} \beta_{slm}(U_{sl} - U_{sm}) + \psi_{gsm} + \alpha_{sm}\rho_{sm}g \tag{11}$$

β_{slm} accounts for the momentum exchange between solid phase l and m. In the present model, τ_s is expressed as

$$\tau_{sm} = -p_{sm}I + 2\alpha_{sm}\mu_{sm}D_{sm} + \alpha_{sm}\lambda_{sm}tr(D_{sm})I \tag{12}$$

where p_{sm}, μ_{sm} and λ_{sm} are the so-called granular pressure, dynamic viscosity, and bulk viscosity, respectively. These three granular properties are determined by the kinetic theory of granular flows (KTGF), a function of the granular energy Θ_{sm}. In this study, Θ_{sm} is calculated based on the algebraic form. With Θ_{sm}, p_{sm}, μ_{sm} and λ_{sm} are calculated according to

$$p_{sm} = 2\rho_{sm}\alpha_{sm}^2 g_{0m}\Theta_{sm}(1+e_{sm})$$

$$\mu_{sm} = \frac{4}{5}\alpha_{sm}^2\rho_{sm}d_{sm}g_{0m}(1+e_{sm})(\Theta_{sm}/\pi)^{\frac{1}{2}} + \frac{10\rho_{sm}d_{sm}\sqrt{\Theta_{sm}\pi}}{96g_{0m}(1+e_{sm})}\left[1+\frac{4}{5}(1+e_{sm})\alpha_{sm}g_{0m}\right]^2 \tag{13}$$

$$\lambda_{sm} = \frac{4}{3}\alpha_{sm}^2\rho_{sm}d_{sm}g_{0m}(1+e_{sm})(\Theta_{sm}/\pi)^{\frac{1}{2}}$$

where g_{0m} and e_{sm} are the radial distribution function and restitution coefficient, respectively.

The energy equation of solid phase m is

$$\frac{\partial(\alpha_{sm}\rho_{sm}C_{sm}T_{sm})}{\partial t} + \nabla \cdot (\alpha_{sm}\rho_{sm}C_{psm}T_{sm}U_{sm}) = \nabla \cdot (\alpha_{sm}q_{sm}) + h_{gsm}(T_{sm} - T_g) + \Delta H_{sm} \qquad (14)$$

The mass species equation of solid phase *m* is

$$\frac{\partial \alpha_{sm}\rho_{sm}Y_{smk}}{\partial t} + \nabla \cdot (\alpha_{sm}\rho_{sm}Y_{smk}U_{sm}) = R_{smk} \qquad (15)$$

Pyrolysis kinetics

In the tutorial case, a global multi-component multi-step reaction kinetics is used to describe biomass fast pyrolysis chemistry. Biomass components include cellulose, hemicellulose and lignin. In the model, virgin biomass is first elevated to an active state, followed by parallel reactions in which tar, syngas and bio-char are formed. At certain temperature, some portion of the tar will further be decomposed into syngas. Mass consumption rate of each species in both phases are determined by first-order Arrhenius kinetics considering the rate constant *A*, activation energy *E*, and temperature *T*.

$$\frac{dm}{dt} = -mk$$
$$k = A\exp(-E/RT) \qquad (16)$$

Code description

The overall structure of BIOTC is shown in Fig. 1. It can be seen that BIOTC is composed of two main sub-solvers, i.e., the multi-fluid fluidized-bed transport sub-solver and pyrolysis reaction sub-solver. In the multi-fluid fluidized-bed transport sub-solver, all conservative equations are discretized and the resulting matrices are solved by their corresponding PDE solvers.
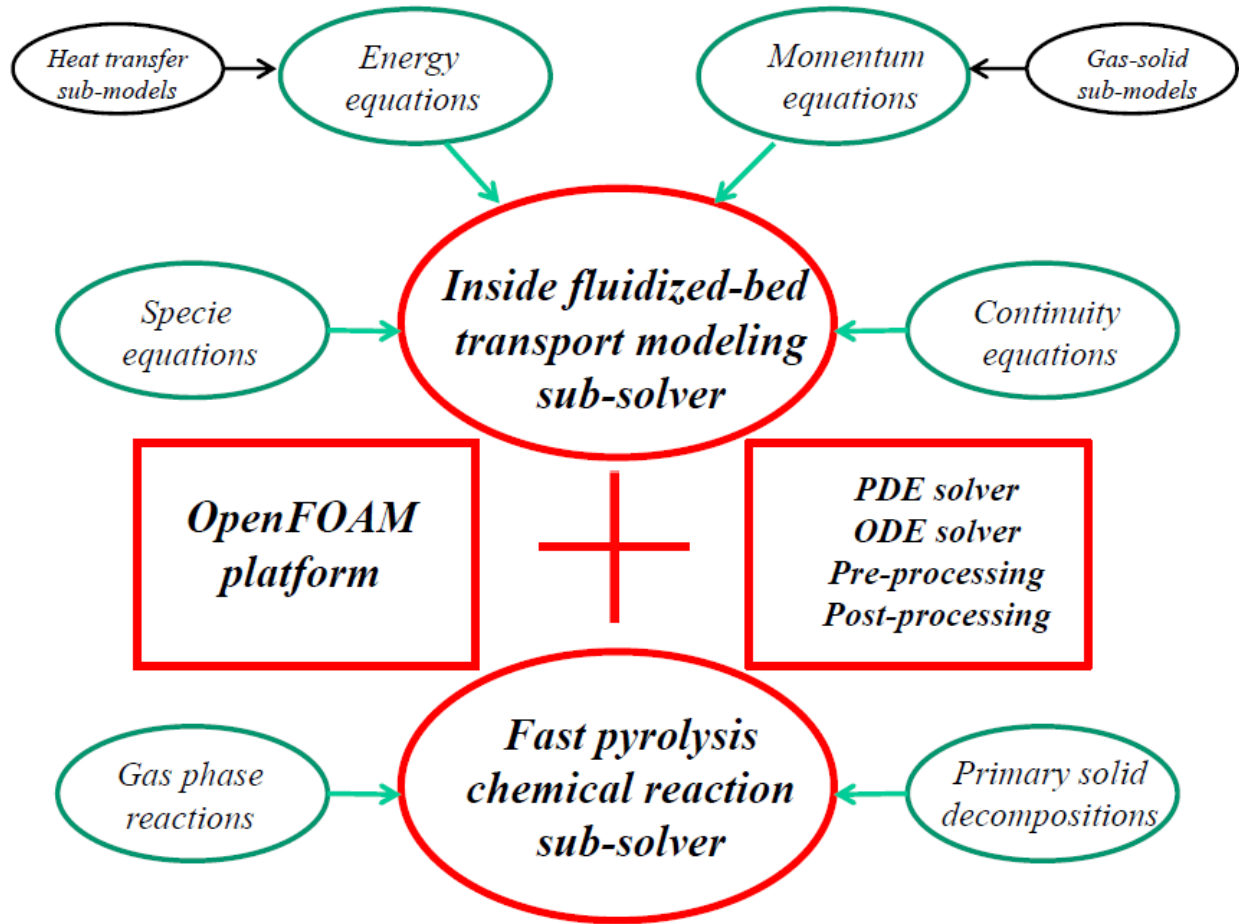
Fig. 1 The overall structure of BIOTC

In the code package, the *alphaEqn* is for the continuity equations of all phases. The principle to construct the discretized matrix has been detailed in [3] where phase-fraction boundedness is ensured by a non-conservative form. The *UEqn* is designed to construct the pre-pressure-solving phase velocity matrices. Then these velocity matrices are used in the *pEqn* for pressure Poisson equation. After the pressure is solved, velocities of all phases are updated. *TEqn* and *YEqn* are designed to solve phase temperature and mass species, respectively. The overall procedure is listed as follows:

a) Solve all solid phase continuity equations to obtain the volume fraction of each solid phase. The model also considers the effects of other solid phases on the volume fraction of the respective solid phase that is being solved. Prior to solving the solid phase continuity equation, the solid granular pressure is obtained, which is then included into the continuity equation to prevent the solid phase volume fraction from exceeding the maximum packing limit. When discretizing the continuity equation, make the volume fraction being solved as implicit as possible to accelerate the matrix iteration convergence. Finally, as the sum of the volume fractions of all phases should be unity in every finite volume, the gas phase volume fraction field is calculated.

b) Calculate the inter-phase interaction coefficients β_{gsm}, h_{gsm} and β_{slm}.

c) Solve the energy equations in all phases to obtain the temperature fields. Same as in step (a), make the temperature being solved as implicit as possible.

d) Solve mass species equations in all phases. It should be mentioned that in order to stabilize the solution, the phase-intensive formulation needs to be employed to treat the instability when α_{sm} of biomass is approaching zero.

e) Construct the velocity equations of all phases. These velocity equations are used to predict intermediate velocity values for establishing fluxes. A conservation scheme is used to improve numerical stability. In this step, gravity is treated explicitly while the drag term is handled semi-implicitly. This is the essential point of the inter-phase coupling.

f) Use the flux constraint in multi-phase flows to construct the pressure Poisson equation. In this step, gravity and the excluded part of the drag term in the last step are included into flux re-establishment.

g) Solve the gas phase pressure Poisson equation and use the resulting pressure to correct phase velocities.

In the pyrolysis sub-solver, the intermediate values of phase properties obtained in the multi-fluid fluidized-bed transport sub-solver are used to calculate reaction source terms in each finite volume by the ordinary differential equation (ODE) solver. After solving the kinetic reactions, physical variables such as densities, mass fractions, and temperatures, etc., are updated. These are realized in the *reactUpdate*.

Tutorial case

The tutorial case is a two-dimensional modeling of biomass fast pyrolysis fluidized-bed reactor. In this simulation, a reactor with 0.34 m in height and 0.038 m in diameter is operated in bubbling fluidization regime, as shown in Fig. 2. Biomass with density of 400 kg/m^3 and diameter approximately from 5×10^{-5} m is fed from a side injector at a rate of 0.1 kg/h while the fluidization nitrogen is added continuously from the bottom at a speed of 0.36 m/s. The fed biomass is purely cellulose. In-bed fluidization agent, which is silica sand with density of 2,649 kg/m^3 and diameter of 5.2×10^{-4} m, is initially packed at a volume fraction of 0.59 to a height of 5.5×10^{-2} m. The reactor is preheated to 773 K and continuous external heating is supplied.
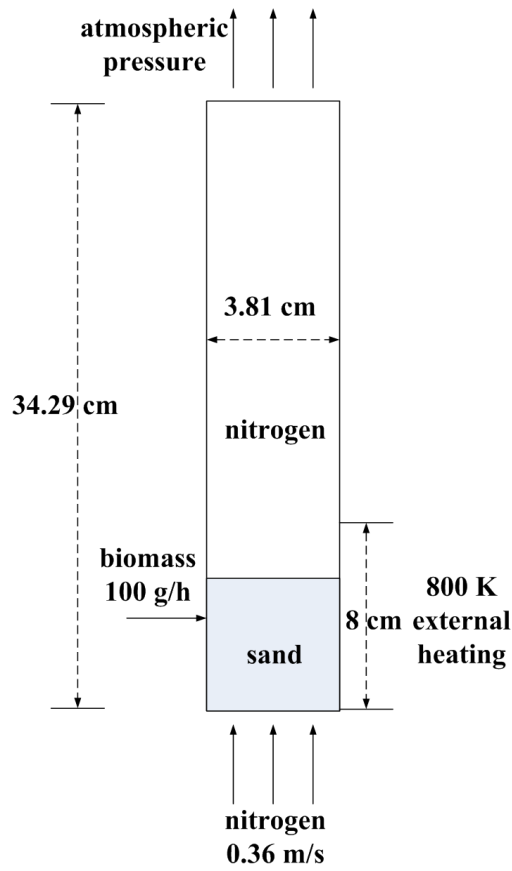
Fig. 2 Schematic of the fluidized-bed reactor.

Before you run the tutorial case, please make sure that an OpenFOAM version higher than 2.1.0 has been installed on your computer. Then copy the code package into your OpenFOAM directory and compile. After compilation, an executable has been created for the coming simulation. Then also copy the tutorial case setup package into your own directory. All the operations to carry out the tutorial case are within this setup directory (which is also called case directory).

Go into the *constant/polyMesh* sub-directory, you will see how the geometry is defined in the *blockMeshDict* file. Vertices, blocks, edges, and boundary types are all listed in this file. Within the case directory, type the command *blockMesh* in the terminal, the mesh is created for the following modeling. Fig. 3 displays the mesh created by the *blockMesh* utility.
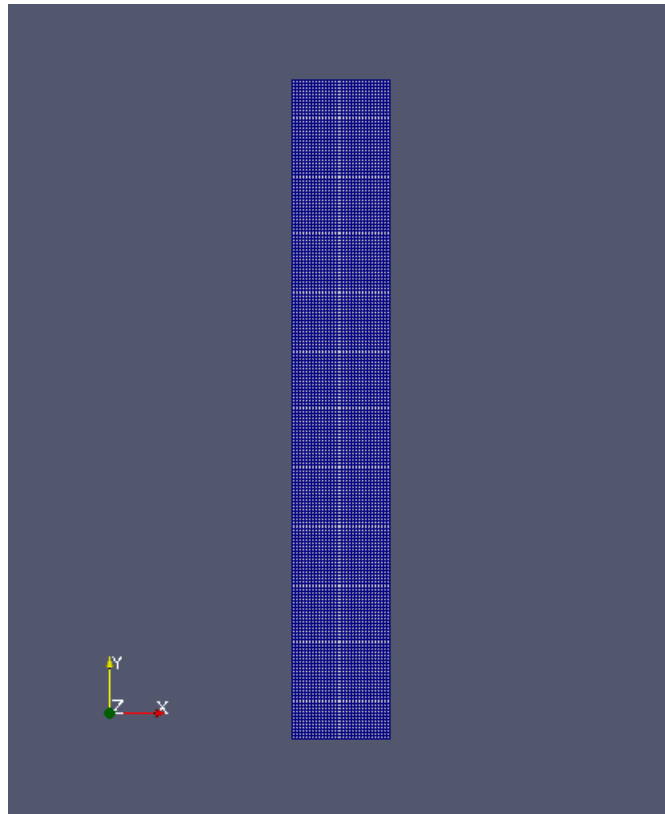
Fig. 2 Mesh created by the blockMesh utility for the tutorial case.

After creation of the mesh, initial and boundary conditions for each variable are specified in the *0* sub-directory. In this sub-directory, files named by variable object name are listed. Take the gas velocity file *gasU* as an example, no-slip wall and inlet/outlet boundary conditions are prescribed as *fixed value* and *zeroGradient*. In addition, variable dimensions should be given in this file.

The physical properties are all given in the files stored within the *constant* sub-directory. *chemistryProperties* defines solid decompositions and the chemistry models used by solid and gas phases. *interPhaseProperties* gives which sub-models selected for gas-solid drag and heat transfer. Solid and gas thermal properties such as heat capacity, density, viscosity, etc., are stored in *solidThermophysicalProperties* and *thermophysicalProperties*.

The running control parameters are stored in the *system* sub-directory. These include *controlDict*, *fvSchemes*, and *fvSolution*. *controlDict* contains the time step, total running time, writing interval for post-processing, etc. *fvSchemes* lists all the numerical schemes that used to discretize finite volume formulations, such as Laplacian operator Δ. *fvSolution* gives the solution strategy for solving each discretized equations.

To run this case, you just need to type *bioPyroFoam* within the case directory. The monitoring information will be displayed on screen. After computation, the results can be post-processed with the adhered software *paraFoam*. How to use *paraFoam* is described in the OpenFOAM user guide.

[1] OpenCFD, OpenFOAM-The Open Source CFD Toolbox-User Guide, 2st ed., OpenCFD Ltd, United Kingdom, 2011.

[2] OpenCFD, OpenFOAM-The Open Source CFD Toolbox-Programmer's Guide, 2st ed., OpenCFD Ltd, United Kingdom, 2011.

[3] H.G. Weller, Derivation, Modelling and Solution of the Conditionally Averaged Two-Phase Flow Equations., OpenCFD Ltd, 2005.