# A REFERENCE GUIDE TO USE MFIX WITH ISAT AND DQMOM

Prepared by:

Rong Fan
Nan Xie
Francine Battaglia
Rodney O Fox

Iowa State University
Ames, Iowa

October 7, 2004

# Contents

# 1 In Situ Adaptive Tabulation (ISAT)

## 1.1 ISAT Theory

In a gas-solids mixture, the thermochemical state can be determined by $N_s$ variables such as void fraction, mass fraction and enthalpy [1]. However, if other dependencies exist between these variables, the degrees of freedom decrease to $D$. The composition vector $\phi$ is defined to include $D$ variables, which is a subset of $N_s$. The transport equation for the composition vector has the form:

$$\frac{\mathrm{d}\phi}{\mathrm{d}t} = S(\phi) + T(\phi) \tag{1}$$

where $\phi$ has components ($\phi_1$, $\phi_2$, $\cdots$, $\phi_D$), and $S(\phi)$ and $T(\phi)$ are the changes in $\phi$ due to chemical reactions and transport, respectively. The composition of reactive flows is affected by reactions and transport which includes convection and diffusion.

When CFD codes use fractional time stepping for reactive flow calculations, splitting techniques can be applied so that the different processes are treated in separate fractional steps. In the first fractional time step, the change due to convection and diffusion is solved for every node using:

$$\frac{\mathrm{d}\phi}{\mathrm{d}t} = T(\phi) \tag{2}$$

with the initial condition $\phi(t)$. The solution of Eq. (2) is denoted as $\phi^*(t+\delta t)$. In the next fractional time step, the change due to chemical reactions is solved using:

$$\frac{\mathrm{d}\phi}{\mathrm{d}t} = S(\phi) \tag{3}$$

with the initial condition $\phi^*(t+\delta t)$. The composition of all nodes are then approximated to $\phi(t+\delta t)$. The overall fractional time stepping is represented as:

$$\phi(t) \overset{\text{transport}}{\longrightarrow} \phi^*(t+\delta t) \overset{\text{reaction}}{\longrightarrow} \phi(t+\delta t). \tag{4}$$

The chemical source terms are decoupled from the transport terms and can be treated with efficient numerical methods such as ISAT. Thus, solving for chemical source terms is based on a given chemical composition at time $t$ to determine the composition at time $t+\delta t$ resulting from chemical reactions. In a time-dependent finite-volume code, Eq. (3) is integrated repetitively for each node at every time step with each new set of initial conditions.

The essential ideas and ingredients of the ISAT approach [1,2] are summarized as follows and the flow chart is shown in Fig. 1:

1. A splitting scheme is employed so that the problem is reduced to determining the reaction mapping $R(\phi)$, which is the solution to Eq. (3) after a time $\delta t$ from the initial condition $\phi^*$.

2. A table is built in situ as the reactive flow calculation is performed so that only the accessed region of the composition space is tabulated.

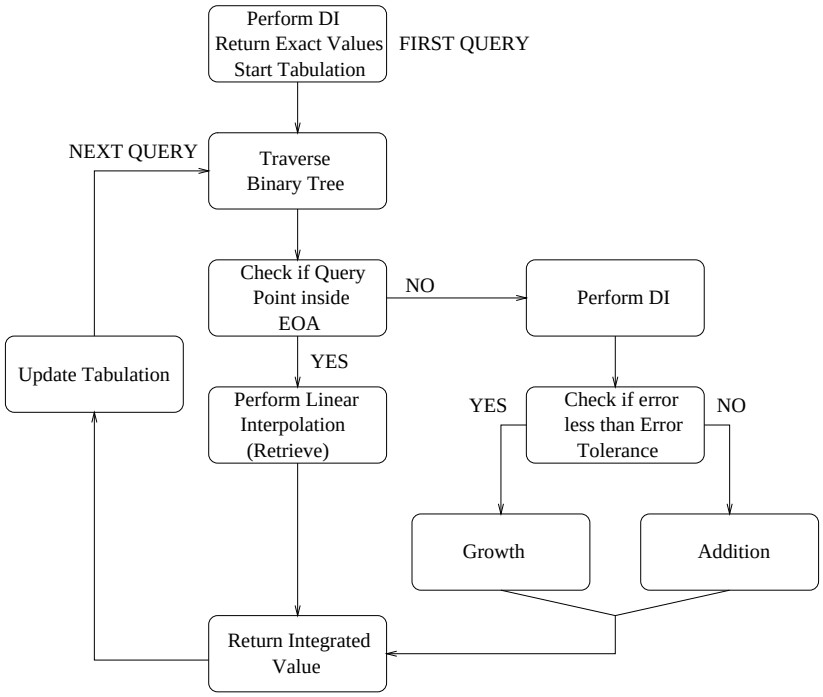


Figure 1: The flow chart of the key steps in ISAT.

3. A table entry (or record) consists of (i) a reference composition vector $\phi^0$, (ii) the reaction mapping evaluated at the reference composition vector $R(\phi^0)$, (iii) the Jacobian matrix for the reaction mapping evaluated at the reference composition vector $A(\phi^0)$, and (iv) information needed to define the ellipsoid of accuracy (EOA). The reaction mapping and its Jacobian can then be used to obtain a linear approximation to the reaction mapping evaluated at a "nearby" query composition $\phi^q$:

$$R(\phi^q) \approx R(\phi^0) + A(\phi^0)(\phi^q - \phi^0) \tag{5}$$

   The EOA is an ellipsoidal region, centered at $\phi^0$, within which Eq. (5) is known to be accurate.

4. The EOA is defined in terms of the singular values of the Jacobian matrix and a user-specified error tolerance. An EOA is initialized and grown to ensure that (with high probability) the error involved in Eq. (5) is within the user-specified error tolerance.

5. The records are stored in a binary tree which, given a query composition $\phi^q$, can be traversed to obtain a table entry $\phi^0$ which in some sense is close to $\phi^q$.

6. Each call to ISAT is referred to as a *query*. If the new query point is within the EOA, then Eq. (5) is employed to estimate the reaction mapping. This step is known as a *retrieve*.

7. If the query is not fulfilled by a *retrieve*, then the reaction mapping is computed by DI. Based on the result, the EOA is examined to determine if the linear approximation is sufficiently accurate. If it is, then the EOA is "grown" to include the query point. This is referred to as a *grow*. If the query is not satisfied by a grow then the point is tabulated as a new data point, referred to as *add*.

8. As the calculation proceeds, with increasing probability, the query composition $\phi^{\mathrm{q}}$ lies within the EOA of a table entry $\phi^0$, so that the mapping is efficiently retrieved using Eq. (5). In a typical calculation, the table is empty at the first time step. Thus a newly created table is dominated by *adds*, followed by an intermediate period dominated by *grows*. Finally, the table reaches maturity where almost every query point leads to a *retrieve*. The overall computational efficiency increases dramatically once the table is mature since the retrieve step is computationally inexpensive as compared to DI.

ISATAB is a Fortran library which implements the ISAT algorithm for a vector-valued general function $\mathbf{f}(\mathbf{x})$ [3]. The user must provide a subroutine to evaluate $\mathbf{f}(\mathbf{x})$ given the vector $\mathbf{x}$ and its derivatives. The function $\mathbf{f}(\mathbf{x})$ and vector $\mathbf{x}$ denote $R(\phi)$ and $\phi$, respectively. In ISATAB the time step $\delta t$ is also tabulated and thus need not be a fixed value for the entire simulation [3].

## 1.2 Implementation of ISAT in MFIX

### 1.2.1 Equations solved in MFIX

Repeated here are the equations solved in MFIX, including continuity, species, and energy for gas and solids phases [4]:

*Gas and solids continuity*

$$\frac{\partial}{\partial t}(\epsilon_{\mathrm{g}}\rho_{\mathrm{g}}) + \bigtriangledown \cdot (\epsilon_{\mathrm{g}}\rho_{\mathrm{g}}\mathbf{V}_{\mathrm{g}}) = \sum_{n=1}^{N_{\mathrm{g}}} R_{\mathrm{g}n}, \tag{6}$$

$$\frac{\partial}{\partial t}(\epsilon_{sm}\rho_{sm}) + \bigtriangledown \cdot (\epsilon_{sm}\rho_{sm}\mathbf{V}_{sm}) = \sum_{n=1}^{N_{sm}} R_{smn}. \tag{7}$$

*Gas and solids species*

$$\frac{\partial}{\partial t}(\epsilon_{\mathrm{g}}\rho_{\mathrm{g}}X_{\mathrm{g}n}) + \bigtriangledown \cdot (\epsilon_{\mathrm{g}}\rho_{\mathrm{g}}X_{\mathrm{g}n}\mathbf{V}_{\mathrm{g}}) = R_{\mathrm{g}n}, \tag{8}$$

$$\frac{\partial}{\partial t}(\epsilon_{sm}\rho_{sm}X_{smn}) + \bigtriangledown \cdot (\epsilon_{sm}\rho_{sm}X_{smn}\mathbf{V}_{sm}) = R_{smn}. \tag{9}$$

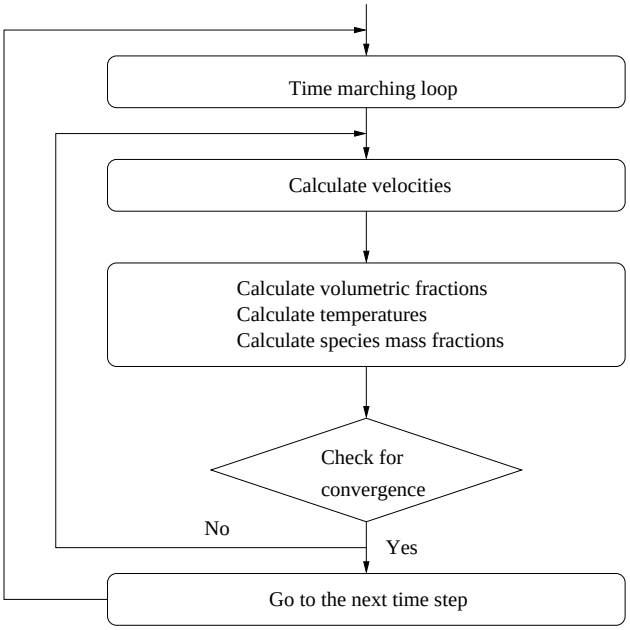


Figure 2: The flow chart of MFIX.

*Gas and solids energy*

$$\epsilon_{\rm g}\rho_{\rm g}C_{\rm pg}\left(\frac{\partial T_{\rm g}}{\partial t}+\mathbf{V}_{\rm g}\cdot\bigtriangledown T_{\rm g}\right) = -\bigtriangledown\cdot\mathbf{q}_{\rm g}+\gamma_{{\rm g}sm}(T_{\rm sm}-T_{\rm g})-\bigtriangleup H_{\rm rg} +H_{\rm wall}(T_{\rm wall}-T_{\rm g}), \quad (10)$$

$$\epsilon_{sm}\rho_{sm}C_{{\rm p}sm}\left(\frac{\partial T_{sm}}{\partial t}+\mathbf{V}_{sm}\cdot\bigtriangledown T_{sm}\right) = -\bigtriangledown\cdot\mathbf{q}_{sm}-\gamma_{{\rm g}sm}(T_{sm}-T_{\rm g})-\bigtriangleup H_{{\rm r}sm}. \quad (11)$$

The subscripts g and $sm$ indicate the gas and solids phases $m$ respectively and $n$ denotes a unique species. Other variables include the phasic volume fraction $\epsilon$, density $\rho$, velocity vector $\mathbf{V}$, mass fraction $X$, rate of formation $R$ and number of chemical species $N$. Figure 2 shows the flow chart of MFIX.

### 1.2.2 Decoupling chemical source terms in MFIX

The equations shown as (6–11) are integrated over a computational cell in the finite-volume method. Usually the chemistry is stiff, for example, due to the large range of time scales. A time-splitting method [1] can be employed to solve the continuity, species and energy equations. The chemical source terms which appear in the continuity, species and energy equations are decoupled so that they can be solved using ISAT.

Applying time-splitting techniques to MFIX, the calculation of species mass fractions, volumetric fractions and temperatures requires two steps:

1. In the first fractional time step, the change due to convection is solved for every node using Eqs. (6)–(11) by setting the mass and heat generation due to reactions equal to zero. The resulting equations have the form:

$$\frac{\partial}{\partial t}(\epsilon_{\rm g}\rho_{\rm g}) + \bigtriangledown \cdot (\epsilon_{\rm g}\rho_{\rm g}\mathbf{V}_{\rm g}) = 0, \tag{12}$$

$$\frac{\partial}{\partial t}(\epsilon_{sm}\rho_{sm}) + \bigtriangledown \cdot (\epsilon_{sm}\rho_{sm}\mathbf{V}_{sm}) = 0, \tag{13}$$

$$\frac{\partial}{\partial t}(\epsilon_{\rm g}\rho_{\rm g}X_{{\rm g}n}) + \bigtriangledown \cdot (\epsilon_{\rm g}\rho_{\rm g}X_{{\rm g}n}\mathbf{V}_{\rm g}) = 0, \tag{14}$$

$$\frac{\partial}{\partial t}(\epsilon_{sm}\rho_{sm}X_{smn}) + \bigtriangledown \cdot (\epsilon_{sm}\rho_{sm}X_{smn}\mathbf{V}_{sm}) = 0, \tag{15}$$

$$\begin{aligned}\epsilon_{\rm g}\rho_{\rm g}C_{\rm pg}\left(\frac{\partial T_{\rm g}}{\partial t} + \mathbf{V}_{\rm g}\cdot\bigtriangledown T_{\rm g}\right) &= -\bigtriangledown\cdot\mathbf{q}_{\rm g} + \gamma_{{\rm g}sm}(T_{sm} - T_{\rm g}) \\ &\quad + H_{\rm wall}(T_{\rm wall} - T_{\rm g}), \end{aligned}\tag{16}$$

$$\epsilon_{sm}\rho_{sm}C_{{\rm p}sm}\left(\frac{\partial T_{sm}}{\partial t} + \mathbf{V}_{sm}\cdot\bigtriangledown T_{sm}\right) = -\bigtriangledown\cdot\mathbf{q}_{sm} - \gamma_{{\rm g}sm}(T_{sm} - T_{\rm g}). \tag{17}$$

with the initial conditions $X_{{\rm g}n}(t)$, $X_{smn}(t)$, $\epsilon_{sm}(t)$, $\rho_{\rm g}(t)$, $T_{\rm g}(t)$ and $T_{sm}(t)$. The solutions of Eqs. (12)–(17) are $X^*_{{\rm g}n}(t+\delta t)$, $X^*_{smn}(t+\delta t)$, $\epsilon^*_{sm}(t+\delta t)$, $\rho^*_{\rm g}(t+\delta t)$, $T^*_{\rm g}(t+\delta t)$ and $T^*_{sm}(t+\delta t)$. For clarity, the spatial dependency of each term is suppressed. However, the reader should keep in mind that the variables must be updated in time at every grid cell.

2. In the next fractional time step, the change due to the chemical source terms for mass fractions are:

$$\frac{\rm d}{{\rm d}t}(\epsilon_{\rm g}\rho_{\rm g}X_{{\rm g}n}) = R_{{\rm g}n}, \tag{18}$$

$$\frac{\rm d}{{\rm d}t}(\epsilon_{sm}\rho_{sm}X_{{\rm sm}n}) = R_{smn}. \tag{19}$$

Differentiating Eqs. (18) and (19) and rearranging terms yields:

$$\frac{\rm d}{{\rm d}t}(X_{{\rm g}n}) = \frac{R_{{\rm g}n}}{\epsilon_{\rm g}\rho_{\rm g}} - \frac{X_{{\rm g}n}}{\epsilon_{\rm g}\rho_{\rm g}}\frac{\rm d}{{\rm d}t}(\epsilon_{\rm g}\rho_{\rm g}), \tag{20}$$

$$\frac{\rm d}{{\rm d}t}(X_{smn}) = \frac{R_{smn}}{\epsilon_{sm}\rho_{sm}} - \frac{X_{smn}}{\epsilon_{sm}\rho_{sm}}\frac{\rm d}{{\rm d}t}(\epsilon_{sm}\rho_{sm}), \tag{21}$$

where the change of volumetric fractions $\epsilon_{\rm g}$ and $\epsilon_{sm}$ are calculated from the continuity equation for the solids phase:

$$\frac{\rm d}{{\rm d}t}(\epsilon_{\rm sm}\rho_{sm}) = \sum_{n=1}^{N_{sm}} R_{smn} \tag{22}$$

with $\epsilon_{\rm g} + \sum_{m=1}^{M} \epsilon_{\rm sm} = 1$. Equation (22) can be rearranged as:

$$\frac{\rm d}{{\rm d}t}(\epsilon_{sm}) = \frac{1}{\rho_{sm}} \sum_{n=1}^{N_{sm}} R_{smn} \tag{23}$$

since $\rho_{sm}$ is constant. Beginning with the continuity equation for the gas phase:

$$\frac{\rm d}{{\rm d}t}(\epsilon_{\rm g}\rho_{\rm g}) = \sum_{n=1}^{N_{\rm g}} R_{gn} \tag{24}$$

the equation can be rewritten:

$$\epsilon_{\rm g}\frac{{\rm d}\rho_{\rm g}}{{\rm d}t} - \rho_{\rm g}\sum_{m=1}^{M}\frac{{\rm d}\epsilon_{sm}}{{\rm d}t} = \sum_{n=1}^{N_{\rm g}} R_{gn}. \tag{25}$$

Then, the change of gas density is:

$$\frac{\rm d}{{\rm d}t}(\rho_{\rm g}) = \frac{1}{\epsilon_{\rm g}}\left(\sum_{n=1}^{N_{\rm g}} R_{gn} + \rho_{\rm g}\sum_{m=1}^{M}\frac{{\rm d}\epsilon_{sm}}{{\rm d}t}\right). \tag{26}$$

From the energy equations for the gas and solids phase, the change due to the chemical source terms are written as:

$$\epsilon_{\rm g}\rho_{\rm g}C_{\rm pg}\frac{\rm d}{{\rm d}t}(T_{\rm g}) = -\triangle H_{\rm rg}, \tag{27}$$

$$\epsilon_{\rm sm}\rho_{\rm sm}C_{\rm psm}\frac{\rm d}{{\rm d}t}(T_{sm}) = -\triangle H_{\rm rsm} \tag{28}$$

and Eqs. (27) and (28) can be rearranged as:

$$\frac{\rm d}{{\rm d}t}(T_{\rm g}) = \frac{-\triangle H_{\rm rg}}{\epsilon_{\rm g}\rho_{\rm g}C_{\rm pg}}, \tag{29}$$

$$\frac{\rm d}{{\rm d}t}(T_{\rm sm}) = \frac{-\triangle H_{\rm rsm}}{\epsilon_{\rm sm}\rho_{\rm sm}C_{\rm psm}}. \tag{30}$$

Thus Eqs. (20) and (21) for mass fractions, Eq. (23) for volumetric fractions, Eq. (26) for gas density and Eqs (29) and (30) for temperatures are integrated with the initial

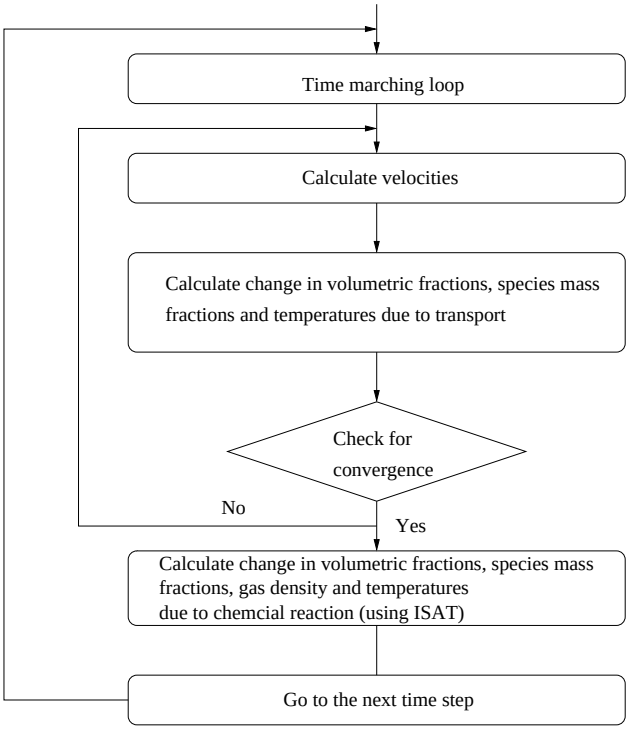


Figure 3: The flow chart of MFIX with ISAT.

conditions $X^*_{gn}(t+\delta t)$, $X^*_{smn}(t+\delta t)$, $\epsilon^*_{sm}(t+\delta t)$, $\rho^*_g(t+\delta t)$, $T^*_g(t+\delta t)$ and $T^*_{sm}(t+\delta t)$. The mass fractions, volumetric fractions, gas density and temperatures at all nodes are updated to $X_{gn}(t+\delta t)$, $X_{sn}(t+\delta t)$ , $\epsilon_s(t+\delta t)$, $\rho_g(t+\delta t)$, $T_g(t+\delta t)$ and $T_{sm}(t+\delta t)$ and pressure is updated using the ideal gas law . In order to use an ODE solver, the user must provide the RHS of Eqs (20), (21), (23), (26), (29) and (30) and their corresponding Jacobian matrix.

With the time-splitting method the chemical reactions for the gas and solids phases are isolated from the transport terms and can be treated with efficient numerical methods such as ISAT. The problem statement for solving the chemical source terms is based on a given chemical composition at time $t$ to determine the composition at time $t+\delta t$ resulting from chemical reactions. During the course of a simulation, Eqs. (20), (21), (23), (26), (29) and (30) are integrated repetitively for each node at every time step with each new set of initial conditions.

As shown in Fig. 3, the chemical source terms are solved using ISAT and the transport terms are calculated using MFIX at each fractional time step. The changes in species mass fractions, volumetric fractions, gas density and temperatures due to transport are calculated with other equations iteratively in the MFIX code. If the equations converge, the change in species mass fractions, volumetric fractions, gas density and temperatures due to chemical reactions are computed using ISAT. Then the calculation moves to the next time step. In order to ensure rapid convergence, MFIX adapts the time step $\delta t_M$ on every iteration.

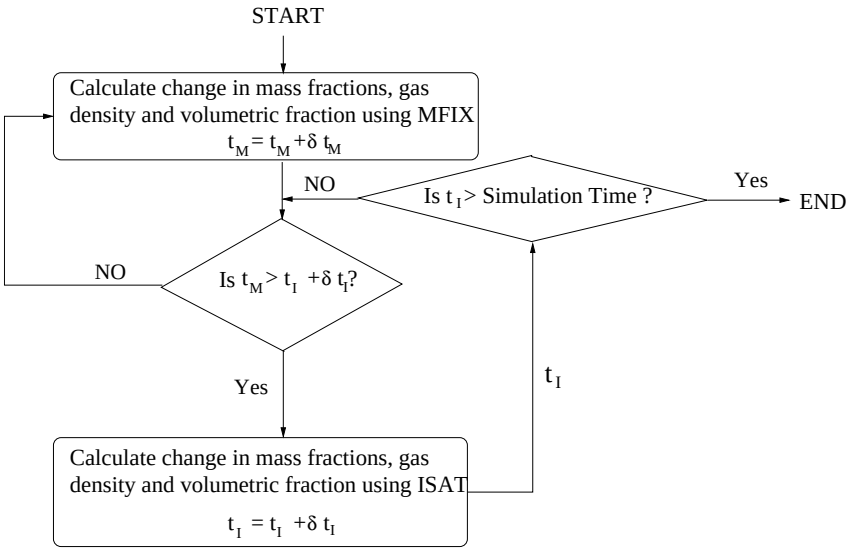


Figure 4: Flow chart of MFIX with ISAT for fixed time step.

From numerical studies [5], the drastically varying time step in MFIX caused poor performance of ISAT. Since ISATAB determines the EOA based both on the distance in the composition space and the difference in the time step, a variable time step in ISAT with similar mass fractions, volumetric fractions and temperatures will produce a new record or growth instead of utilizing the linear interpolation procedure. It appears that a CFD code with a rapidly varying time step needs longer time and more memory to build and store the ISAT table. One possible remedy is to call ISAT from MFIX at a fixed time step $\delta t_{\mathrm{I}}$ during the calculation as shown in Fig. 4. At the beginning, $t_{\mathrm{M}}$ is equal to $t_{\mathrm{I}}$. Time steps in MFIX ($\delta t_{\mathrm{M}}$) vary during the simulation and ISAT is only called when the time in MFIX advances more than $\delta t_{\mathrm{I}}$. Then, ISAT updates the data every $\delta t_{\mathrm{I}}$, which is held constant until $t_{\mathrm{I}}$ is greater than $t_{\mathrm{M}}$.

# 2 Direct Quadrature Method of Moments (DQMOM)

## 2.1 DQMOM Theory

### 2.1.1 Population balance equations

The population balance equations (PBE) for the particle size distribution (PSD) with number density function $n(L;\mathbf{x},t)$ can be written in terms of one internal variable, particle size length $L$, as follows [6]:

$$\frac{\partial n(L;\mathbf{x},t)}{\partial t} + \boldsymbol{\nabla}\cdot[\langle \mathbf{u}_s|L\rangle n(L;\mathbf{x},t)] = S(L;\mathbf{x},t), \tag{31}$$

where $S(L;\mathbf{x},t)$ represents the net rate of introduction of new particles into the system (e.g., due to the chemical reaction, aggregation and breakage) and $\langle \mathbf{u}_s|L\rangle$ is the mean velocity

conditioned on the particle length $L$. By definition, $\langle \mathbf{u}_s | L = L_\alpha \rangle = \mathbf{u}_{s\alpha}$, $\mathbf{x}$ is the spatial coordinate and $t$ is the time.

Using DQMOM, $n(L;\mathbf{x},t)$ can be approximated in terms of a summation of $N$ Dirac delta functions (presumed finite-mode PSD):

$$n(L;\mathbf{x},t) = \sum_{\alpha=1}^{N} \omega_\alpha(\mathbf{x},t)\delta[L - L_\alpha(\mathbf{x},t)], \tag{32}$$

where $\omega_\alpha$ is the *weight* of the delta function centered at the *characteristic length* $L_\alpha$. If Eq. (32) is substituted into Eq. (31), it is possible to derive transport equations for the $N$ weights $\omega_\alpha$ and the $N$ characteristic lengths $L_\alpha$.

The population balance in terms of the presumed finite-mode PSD becomes:

$$\sum_{\alpha=1}^{N} \delta(L-L_\alpha)\left[\frac{\partial \omega_\alpha}{\partial t} + \boldsymbol{\nabla}\cdot(\omega_\alpha \mathbf{u}_{s\alpha})\right] - \sum_{\alpha=1}^{N} \delta'(L-L_\alpha)\left[\omega_\alpha(\frac{\partial L_\alpha}{\partial t} + \mathbf{u}_{s\alpha}\cdot\boldsymbol{\nabla} L_\alpha)\right] = S(L,T) \tag{33}$$

where $\delta'(L-L_\alpha)$ is the first derivative of the Dirac delta function $\delta(L-L_\alpha)$ and after some manipulation, Eq. (33) becomes:

$$\sum_{\alpha=1}^{N} \delta(L-L_\alpha)a_\alpha - \sum_{\alpha=1}^{N} \delta'(L-L_\alpha)[b_\alpha - L_\alpha a_\alpha] = S(L,T), \tag{34}$$

where

$$\frac{\partial \omega_\alpha}{\partial t} + \boldsymbol{\nabla}\cdot(\omega_\alpha \mathbf{u}_{s\alpha}) = a_\alpha, \tag{35}$$

$$\frac{\partial (\omega_\alpha L_\alpha)}{\partial t} + \boldsymbol{\nabla}\cdot(\omega_\alpha L_\alpha \mathbf{u}_{s\alpha}) = b_\alpha. \tag{36}$$

### 2.1.2 Moment transform

Moment transforms can be applied to determine the functional forms of $a_\alpha$ and $b_\alpha$ for solving the PBE. The $k^{\text{th}}$ moment of the PSD is defined as:

$$m_k(\mathbf{x},t) = \int_0^\infty n(L;\mathbf{x},t)L^k dL \approx \sum_{\alpha=1}^{N} \omega_\alpha L_\alpha^k. \tag{37}$$

Given that:

$$\begin{aligned}
&\int_0^\infty \delta(L-L_\alpha)L^k dL = L_\alpha^k, \\
&\int_0^\infty \delta'(L-L_\alpha)L^k dL = -kL_\alpha^{k-1}, \\
&\int_0^\infty \delta''(L-L_\alpha)L^k dL = k(k-1)L_\alpha^{k-2},
\end{aligned} \tag{38}$$

then the moment transform of Eq. (34) yields:

$$\sum_{\alpha=1}^{N} a_\alpha L_\alpha^k (1-k) + k b_\alpha L_\alpha^{k-1} = \overline{S}_k, \tag{39}$$

where

$$\overline{S}_k = \int_0^\infty S(L) L^k dL. \tag{40}$$

The form of Eq. (39) shows the source terms of the transport equations of the $N$ weights $\omega_\alpha$ and characteristic lengths $L_\alpha$ are defined through a linear system involving the first $2N$ moments of the population balance equation (e.g., $k = 0, \cdots, 2N-1$). This linear system can be written in matrix form as:

$$\mathbf{A}\mathbf{x} = \mathbf{d}, \tag{41}$$

where the $2N \times 2N$ coefficient matrix $\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 & \mathbf{A}_2 \end{bmatrix}$ is defined by:

$$\mathbf{A}_1 = \begin{bmatrix} 1 & \cdots & 1 \\ 0 & \cdots & 0 \\ -L_1^2 & \cdots & -L_N^2 \\ \vdots & \ddots & \vdots \\ 2(1-N)L_1^{2N-1} & \cdots & 2(1-N)L_N^{2N-1} \end{bmatrix} \tag{42}$$

and

$$\mathbf{A}_2 = \begin{bmatrix} 0 & \cdots & 0 \\ 1 & \cdots & 1 \\ 2L_1 & \cdots & 2L_N \\ \vdots & \ddots & \vdots \\ (2N-1)L_1^{2N-2} & \cdots & (2N-1)L_N^{2N-2} \end{bmatrix}. \tag{43}$$

$$\mathbf{x} = \begin{bmatrix} a_1 & \cdots a_N & b_1 & \cdots & b_N \end{bmatrix}^{\mathrm{T}} = \begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix}, \tag{44}$$

$$\mathbf{d} = \begin{bmatrix} \overline{S}_0 & \cdots & \overline{S}_{2N-1} \end{bmatrix}^{\mathrm{T}}. \tag{45}$$

If $N = 1$ the PSD is represented by only one delta function and $\mathbf{A}$ is the identity matrix. The source terms are:

$$\begin{bmatrix} a_1 \\ b_1 \end{bmatrix} = \begin{bmatrix} \overline{S}_0 \\ \overline{S}_1 \end{bmatrix}. \tag{46}$$

If $N = 2$ the PSD is described by two delta functions, and

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ -L_1^2 & -L_2^2 & 2L_1 & 2L_2 \\ -2L_1^3 & -2L_2^3 & 3L_1^2 & 3L_2^2 \end{bmatrix}. \tag{47}$$

By inverting A, we can get the source term:

$$\begin{bmatrix} a_1 \\ a_2 \\ b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} (3L_1 - L_2)L_2^2 & -6L_1L_2 & 3(L_1 + L_2) & -2 \\ (L_1 - 3L_2)L_1^2 & 6L_1L_2 & -3(L_1 + L_2) & 2 \\ 2L_2^2L_1^2 & -(4L_1^2 + L_1L_2 + L_2^2)L_2 & 2(L_1^2 + L_1L_2 + L_2^2) & -L_1 - L_2 \\ -2L_2^2L_1^2 & (L_1^2 + L_1L_2 + 4L_2^2)L_2 & -2(L_1^2 + L_1L_2 + L_2^2) & L_1 + L_2 \end{bmatrix} \times \frac{1}{(L_1 - L_2)^3} \begin{bmatrix} \overline{S}_0 \\ \overline{S}_1 \\ \overline{S}_2 \\ \overline{S}_3 \end{bmatrix}. \quad (48)$$

### 2.1.3 Aggregation and breakage equation

The moment transform of the source term only with aggregation and breakage (the molecular growth rate is zero) [7] is:

$$\overline{S}_k(\mathbf{x}, t) = \overline{B}_k^a(\mathbf{x}, t) - \overline{D}_k^a(\mathbf{x}, t) + \overline{B}_k^b(\mathbf{x}, t) - \overline{D}_k^b(\mathbf{x}, t), \quad (49)$$

where

$$\overline{B}_k^a = \frac{1}{2} \int_0^\infty n(\lambda; \mathbf{x}, t) \int_0^\infty \beta(u, \lambda)(u^3 + \lambda^3)^{k/3} n(u; \mathbf{x}, t) du d\lambda, \quad (50)$$

$$\overline{D}_k^a = \int_0^\infty L^k n(L; \mathbf{x}, t) \int_0^\infty \beta(L, \lambda) n(\lambda; \mathbf{x}, t) d\lambda dL, \quad (51)$$

$$\overline{B}_k^b = \int_0^\infty L^k \int_0^\infty a(\lambda) b(L|\lambda) n(\lambda; \mathbf{x}, t) d\lambda dL, \quad (52)$$

$$\overline{D}_k^b = \int_0^\infty L^k a(L) n(L; \mathbf{x}, t) dL, \quad (53)$$

are respectively the moments of the birth and death rates for aggregation and breakage. The variable $\beta(L, \lambda)$ is the aggregation kernel that is the frequency of collision of two particles with length $L$ and $\lambda$, $a(L)$ is the breakage kernel that is the frequency of disruption of a particle of length $L$, and $b(L|\lambda)$ is the fragment distribution function that contains information on the fragments produced by a breakage event.

Applying the quadrature approximation reported in Eq. (37), the source term becomes:

$$\overline{S}_k = \frac{1}{2} \sum_{i=1}^N \omega_i \sum_{j=1}^N \omega_j (L_i^3 + L_j^3)^{k/3} \beta_{ij} - \sum_{i=1}^N L_i^k \omega_i \sum_{j=1}^N \beta_{ij} \omega_j + \sum_{i=1}^N a_i \overline{b_i}^{(k)} \omega_i - \sum_{i=1}^N L_i^k a_i \omega_i, \quad (54)$$

where $\beta_{ij} = \beta(L_i, L_j)$, $a_i = a(L_i)$, and

$$\bar{b}_i^{(k)} = \int_0^{+\infty} L^k b(L|L_i) dL. \quad (55)$$

The aggregation and breakage kernel developed from kinetic theory are used here:

$$\beta_{ij} = \Psi_a g_{ij} \left(\frac{3\theta_s}{\rho_s}\right)^{1/2} (L_i + L_j)^2 \left(\frac{1}{L_i^3} + \frac{1}{L_j^3}\right)^{1/2}, \tag{56}$$

$$a_i = \Psi_b \sum_{j=1}^{N} \omega_j g_{ij} \left(\frac{3\theta_s}{\rho_s}\right)^{1/2} (L_i + L_j)^2 \left(\frac{1}{L_i^3} + \frac{1}{L_j^3}\right)^{1/2}, \tag{57}$$

where $\Psi_a$ and $\Psi_b$ are the success-factors for aggregation and breakage and $\theta_s$ is the average granular temperature. Equations (56) and (57) assume equal density in the derivations.

## 2.2 Implementation of DQMOM in MFIX

### 2.2.1 Equations solved in MFIX

In order to be consistent with the variables used in the multi-fluid model, the weights $\omega_\alpha$ and abscissas $L_\alpha$ are associated with the solid volume fraction $\varepsilon_{s\alpha}$ and the effective length $\varepsilon_{s\alpha} L_\alpha$ for each solid phase [8]. The volume fraction of each solid phase is related to the abscissas $L_\alpha$ and weights $\omega_\alpha$ by:

$$\varepsilon_{s\alpha} = k_v L_\alpha^3 \omega_\alpha = k_v \frac{\mathcal{L}_\alpha^3}{\omega_\alpha^2}, \tag{58}$$

and the effective length of the solid phase is:

$$\varepsilon_{s\alpha} L_\alpha = k_v L_\alpha^4 \omega_\alpha = k_v \frac{\mathcal{L}_\alpha^4}{\omega_\alpha^3}, \tag{59}$$

where $k_v$ is a volumetric shape factor (e.g., for spherical particles $k_v = \pi/6$) and $\mathcal{L}_\alpha = \omega_\alpha L_\alpha$. If Eq. (58) and Eq. (59) are substituted into the transport equation for $\varepsilon_{s\alpha}$ and $\varepsilon_{s\alpha} L_\alpha$, the following equations can be obtained:

$$\frac{\partial(\varepsilon_{s\alpha}\rho_{s\alpha})}{\partial t} + \boldsymbol{\nabla} \cdot (\varepsilon_{s\alpha}\rho_{s\alpha}\mathbf{u}_{s\alpha}) = 3k_v\rho_{s\alpha}L_\alpha^2 b_\alpha - 2k_v\rho_{s\alpha}L_\alpha^3 a_\alpha. \tag{60}$$

$$\frac{\partial(\varepsilon_{s\alpha}L_\alpha\rho_{s\alpha})}{\partial t} + \boldsymbol{\nabla} \cdot (\varepsilon_{s\alpha}L_\alpha\rho_{s\alpha}\mathbf{u}_{s\alpha}) = 4k_v\rho_{s\alpha}L_\alpha^3 b_\alpha - 3k_v\rho_{s\alpha}L_\alpha^4 a_\alpha. \tag{61}$$

The first transport equation (Eq. (60)) represents the continuity equation for the $\alpha^{\text{th}}$ solid phase in the presence of aggregation and breakage, but without mass transfer between gas and solids. Equation (61) is just a new scalar equation for particle length $L_\alpha$ for each solid phase. Now defining $a'_\alpha$ and $b'_\alpha$ to be the source terms of the transport equation for void fraction $\varepsilon_{s\alpha}$ and characteristic length $L_\alpha$, then the source terms can be related with source terms $\mathbf{x}$ by:

$$\mathbf{x}' = \mathbf{C}\mathbf{x} \tag{62}$$

where

$$\mathbf{x}' = \begin{bmatrix} a'_1 & \cdots a'_N & b'_1 & \cdots & b'_N \end{bmatrix}^{\mathrm{T}} = \begin{bmatrix} \mathbf{a}' \\ \mathbf{b}' \end{bmatrix}, \tag{63}$$

$$\mathbf{C} = \begin{bmatrix} -2k_v\rho_{s1}L_1^3 & 0 & 0 & 3k_v\rho_{s1}L_1^2 & 0 & 0 \\ 0 & \ddots & 0 & 0 & \ddots & 0 \\ 0 & 0 & -2k_v\rho_{sN}L_N^3 & 0 & 0 & 3k_v\rho_{sN}L_N^2 \\ -3k_v\rho_{s1}L_1^4 & 0 & 0 & 4k_v\rho_{s1}L_1^3 & 0 & 0 \\ 0 & \ddots & 0 & 0 & \ddots & 0 \\ 0 & 0 & -3k_v\rho_{sN}L_N^4 & 0 & 0 & 4k_v\rho_{sN}L_N^3 \end{bmatrix}. \tag{64}$$

If $N = 1$ then:

$$\begin{bmatrix} a_1' \\ b_1' \end{bmatrix} = \begin{bmatrix} -2k_v\rho_{s1}L_1^3 & 3k_v\rho_{s1}L_1^2 \\ -3k_v\rho_{s1}L_1^4 & 4k_v\rho_{s1}L_1^3 \end{bmatrix} \begin{bmatrix} a_1 \\ b_1 \end{bmatrix} \tag{65}$$

If $N = 2$ then

$$\begin{bmatrix} a_1' \\ a_2' \\ b_1' \\ b_2' \end{bmatrix} = \begin{bmatrix} -2k_v\rho_{s1}L_1^3 & 0 & 3k_v\rho_{s1}L_1^2 & 0 \\ 0 & -2k_v\rho_{s2}L_2^3 & 0 & 3k_v\rho_{s2}L_2^2 \\ -3k_v\rho_{s1}L_1^4 & 0 & 4k_v\rho_{s1}L_1^3 & 0 \\ 0 & -3k_v\rho_{s2}L_2^4 & 0 & 4k_v\rho_{s2}L_2^3 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ b_1 \\ b_2 \end{bmatrix} \tag{66}$$

### 2.2.2 Solution technique

The solution technique that will be coded in MFIX is shown as Fig. 5. If $N = 2$, at time $t = 0$, the initial conditions for the void fractions $\varepsilon_{s1}, \varepsilon_{s2}$ and particle lengths $L_1, L_2$ are given in the mfix.dat file. From these initial values, the weights can be calculated from Eq. (58) and Eq. (59). Source terms are related with weights and lengths by the model for chemical reaction, aggregation and breakage. Using Eq. (48), the source terms for the transport equations for weights and weighted characteristic lengths can be solved. Then using Eq. (66), it is easy to get the source terms for the transportation equations for void fractions and lengths. As long as we get the sources terms for these two transportation equations, we can use MFIX to solve these two transport equations, (transportation equation for void fraction is already in the code, we need solve two more scalar transport equation for each length.). Then using Eq. (58) and Eq. (59), the particle lengths and weights for the next time step can be calculated.

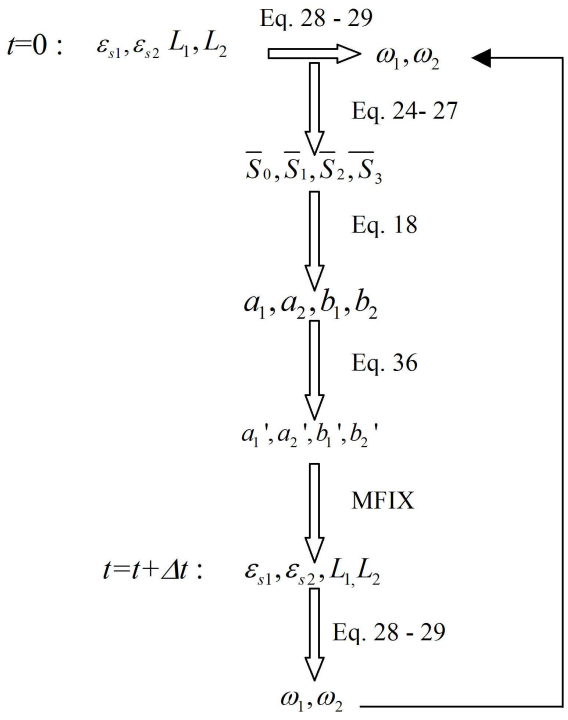


Figure 5: The solution technique used in MFIX.

# References


[1] Pope, S. B. (1997). Computationally efficient implementation of combustion chemistry using in situ adaptive tabulation. *Combustion Theory and Modelling*, *1*, 41–63.

[2] Fox, R. O. (2003). Computational Models for Turbulent Reacting Flows. *Cambridge University Press*.

[3] Pope, S. B. (2000). ISAT User's Guide and References Manual.

[4] Syamlal, M., Rogers, W. and O'Brien, T. (1993). MFIX Documentation: Theory Guide. *Technical Note DOE/METC-95/1013*.

[5] Xie, N., Battaglia, F. and Fox, R. O. (2004). Simulations of multiphase reactive flows in fluidized beds using in situ adaptive tabulation. *Combustion Theory and Modelling*, *8*, 195–209.

[6] Randolph, A. D. and Larson, M. A. (1971). Theory of Particulate Processes. *Academic Press*.

[7] Marchisio, D. L. and Fox, R. O., Solution of population balance equations using direct quadrature method of moments, *Journal of Computational Physics*, submitted.

[8] Fan, R., Marchisio, D. L. and Fox, R. O. (2004), Application of the direct quadrature method of moments to polydisperse gas-solid fluidized beds. *Powder Technology*, *139*, 7–20.

# Appendix A User Reference for ISAT and DQMOM

## A.1 Input for ISAT

### A.1.1 Keywords for input data file

The following are parameters set by the user to invoke ISAT calculations.

| Keyword(dimension) | Type | Description |
|---|---|---|
| CALL_DI [F] | L | Variable to decide if chemical reactions are solved using direct integration (DI) with ODE solver. |
| CALL_ISAT [F] | L | Variable to decide if chemical reactions are solved using ISAT. |
| CALL_GROW [F] | L | Variable to decide if particle growth is calculated. |
| ISATdt | DP | Time step for ISAT simulations (usually the value is less than the average time step in MFIX). |

### A.1.2 Algorithm for ISAT

If "CALL_DI=.TRUE." or "CALL_ISAT=.TRUE.", MFIX uses DI or ISAT to calculate the chemical reactions using the time-splitting method. The ODE solver used here is ODEPACK. The user can download the files from the website (www.netlib.org/odepack) and compile using the same compile option as other files of MFIX. Then ODEPACK should be linked as a library in the make_mfix file.

The algorithm of MFIX using ODEPACK (CALL_DI = .TRUE.) is shown in Fig. 6. By this call, the ODEs will be solved every ISATdt if provided, or every time step of MFIX if ISATdt is not provided by DI.

If "CALL_ISAT=.TRUE.", user must provide the ISATAB library and link it in the make_mfix file. Figure 7 shows the flow of MFIX using ISAT. The user must provide ISATdt to keep the high performance of ISAT.

The following is a list of files for the chemical reaction calculations:

| | |
|---|---|
| check_data_chem.f | Checks user input in mfix.dat |
| mchem_mod.f | Defines the global variables. |
| mchem_init.f | Assigns the initial values. |
| misat_table_init.f | Assigns the values for the controlling parameters for ISATAB |

| | |
|---|---|
| | (not needed if CALL_DI = .TRUE). |
| mchem_odepack_init.f | Assigns the values for controlling parameters for ODEPACK. |
| mchem_time_march.f | Interface between MFIX and ODE solver (assigns variables for integration and transfers back the updated values). |
| react.f | Calculates the interface mass transfer, mass generations of gas and solids phases and calls ODEPACK or ISATAB. |
| usrfg.f | Called by ISATAB to provide the values of direct integration and mapping matrix. |
| exponential.f | Provides the mapping matrix for ISATAB. |
| dgpadm.f | Called by exponential.f. |
| calc_jacobian.f | Provides the Jacobian matrix for jac.f and usrfg.f. |
| fex.f | Provides the source terms for ODEPACK and source terms for chemical reactions rates. |
| jac.f | Provides the Jacobian matrix for ODEPACK. |

The following is a list of files that are usually modified to include the chemical reactions using ODE solver or ISAT:

| | |
|---|---|
| mchem_mod.f | Sets the global variables. |
| misat_table_init.f | Sets the controlling parameters for ISATAB. |
| mchem_odepack_init.f | Sets the controlling parameters for ODEPACK. |
| fex.f | The subroutine provides the source terms for equations in the ODE solver and source terms of reactions rates. The source terms have the following order: $[\rho_g, T_g, X_g], [\varepsilon_{s1}, T_{s1}, X_{s1}, d_1], \cdots, [\varepsilon_{sm}, T_{sm}, X_{sm}, d_m]$. If the variables are constant during the simulations, the source terms should be set to zero. For example, for isothermal chemical reactions, the source terms of $T_g$ and $T_{sm}$ are zero. |
| calc_jacobian.f | Provides the Jacobian matrix for ODE solver. |
| transport_prop.f | Provides transport properties. |
| physical_prop.f | Provides physical properties. |

### A.1.3 Example using ISAT

The non-isothermal silane pyrolysis in a fluidized bed (FB) is used as a benchmark case, where detailed chemical reactions are considered. The FB reactor is fed with a mixture of silane ($SiH_4$) and nitrogen ($N_2$). First, a reversible gas phase reaction occurs:

$$SiH_4 \leftrightarrow SiH_2 + H_2 \ .$$

The highly reactive $SiH_2$ undergoes a gas phase reaction to form $Si_2H_6$:

$$SiH_2 + SiH_4 \leftrightarrow Si_2H_6.$$

Then the heterogeneous decomposition of $SiH_4$ and $SiH_2$ on alumina ($Al_2O_3$) particles is described by two irreversible reactions:

$$SiH_4 \rightarrow Si(s) + 2H_2,$$

$$SiH_2 \rightarrow Si(s) + H_2.$$

Thus there are five gaseous species and two solid species needed to describe this flow.

The following files are modified to simulate this case:

| | |
|---|---|
| mchem_mod.f | Sets the global variables. |
| misat_table_init.f | Controlling parameters for ISATAB are set. |
| mchem_odepack_init.f | Controlling parameters for ODEPACK are set. |
| fex.f | The subroutine provides the source terms to ODE solver and sources terms of reactions. For this case, the order of ODEs is $[\rho_g, T_g, X_{SiH4}, X_{SiH2}, X_{H2}, X_{Si2H6}, X_{N2}]$, $[\varepsilon_{s1}, T_{s1}, X_{Si}, X_{Al2O3}, d_1]$. As written in the code, the user should provide the source terms of reactions, which is RXN_source_g(*n*) for gas phase species *n* and RXN_source_s(*m*,*n*) for solid phase *m* and species *n*. Then the source terms of the ODEs are calculated. Note that $N_2$ and $Al_2O_3$ are inert species; therefore the source terms are set to zero. |
| calc_jacobian.f | The subroutine provides the Jacobian matrix for ODE solver. The example case uses the Jacobian matrix generated by ADIFOR (g_derives.f). The ADIFOR can be downloaded from the website (http://www-unix.mcs.anl.gov/autodiff/ADIFOR). |
| transport_prop.f | Provides the transport properties. |
| physical_prop.f | Provides the physical properties such as averaged molecular weight. |

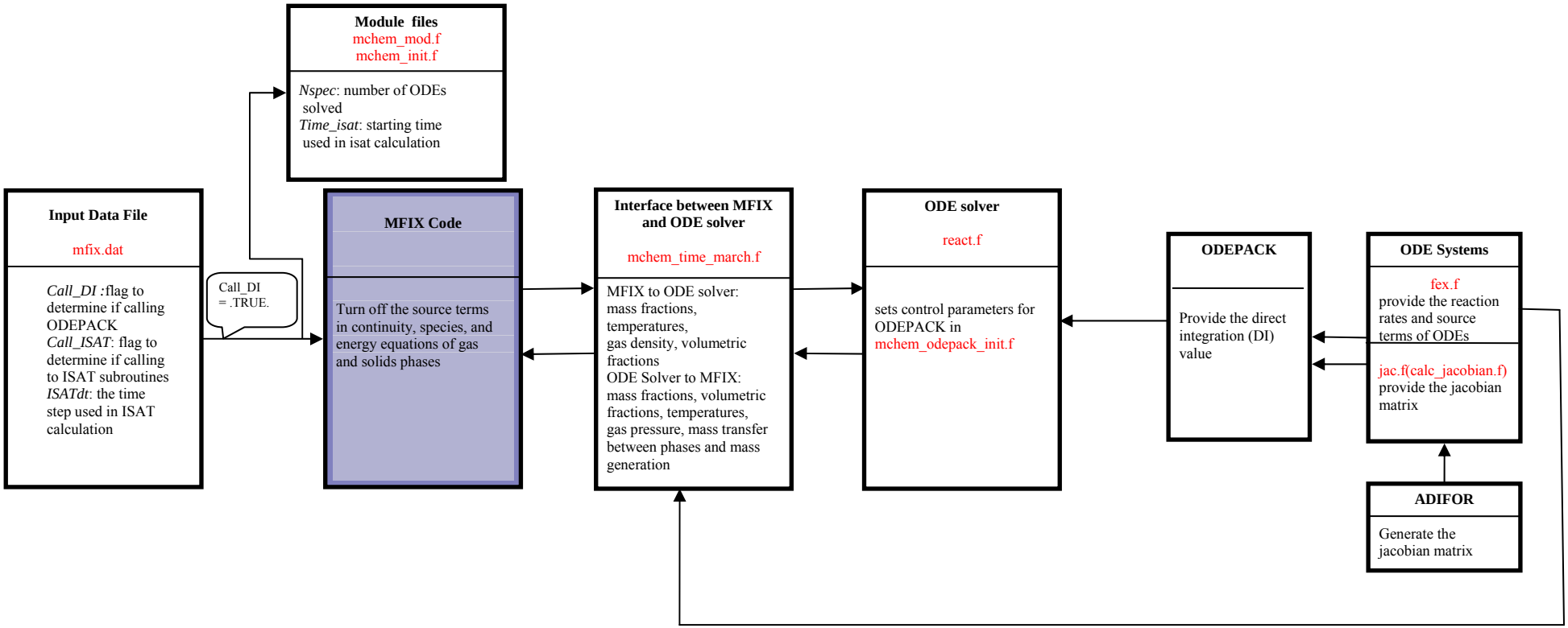


Figure 6: The flow chart of MFIX using ODEPACK (CALL_DI = .TRUE.).

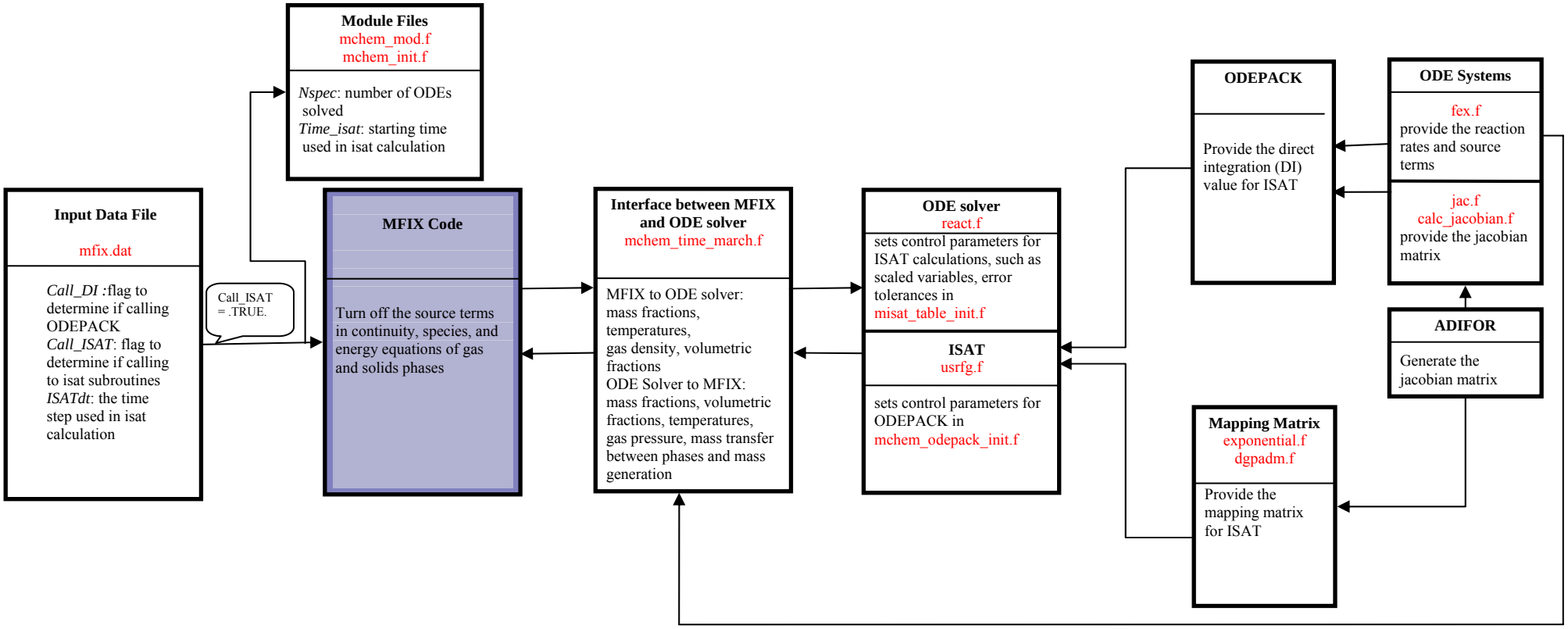


Figure 7: The flow chart of MFIX using ISAT (CALL_ISAT = .TRUE.).

## A.2 Input for DQMOM

### A.2.1 Keywords for input data file

The following are the parameters set by the user to invoke DQMOM calculations.

| Keyword | Type | Description |
|---|---|---|
| CALL_DQMOM [F] | L | Variable to decide if the population balance equations are solved. |
| Nscalar [0] | I | Number of solid phases to solve the population balance equations. |
| D_P0(m) | DP | Initial particle diameters, same as the old one D_P(m). |
| Aggregation_eff [0.0] | DP | Success-factor for aggregation. |
| Breakage_eff [0.0] | DP | Success-factor for breakage. |

### A.2.2 Algorithm for DQMOM

The new files added to MFIX are as follows:

| | |
|---|---|
| odeint.f | ODE solver using adaptive stepsize control for Runge-Kutta. |
| rkck.f | Called by odeint.f. |
| rkqs.f | Called by odeint.f. |
| gaussj.f | Provides matrix inversion. |
| source_population_eq.f | Calculates the source term due to aggregation and breakage for the population balance equations, the main subroutine for the DQMOM method. If you want to use your own aggregation and breakage kernel, you can change in this subroutine. The detail explanation can be seen in the theory guide for the DQMOM method. |
| usr_dqmom.f | Interface between DQMOM and MFIX (update solid void fractions and particle diameter due to aggregation and breakage). |

The algorithm for the interface between MFIX and the DQMOM model flow is shown in Fig. 8.

### A.2.3 Example using DQMOM

The example case is a fluidized bed simulation with two solid phases; each has its own particle size. In the code, the population balance equation is turned on. If aggregation dominates, the average particle size will increase. If breakage dominates, the average particle size will decrease. The user can turn off DQMOM by set “Call_ DQMOM =.FALSE.”. The ODE solver for the scalar in the MFIX code solves the population balance equation, so “Nscalar” has to be set as the number of solid phases. The initial values for the scalar are set as the initial particle diameter. The aggregation and breakage kernel from kinetic theory is used. The success factor of aggregation and breakage can be changed by setting different values for “Aggregation_eff” and “Breakage_eff”. An example mfix.dat file is included.

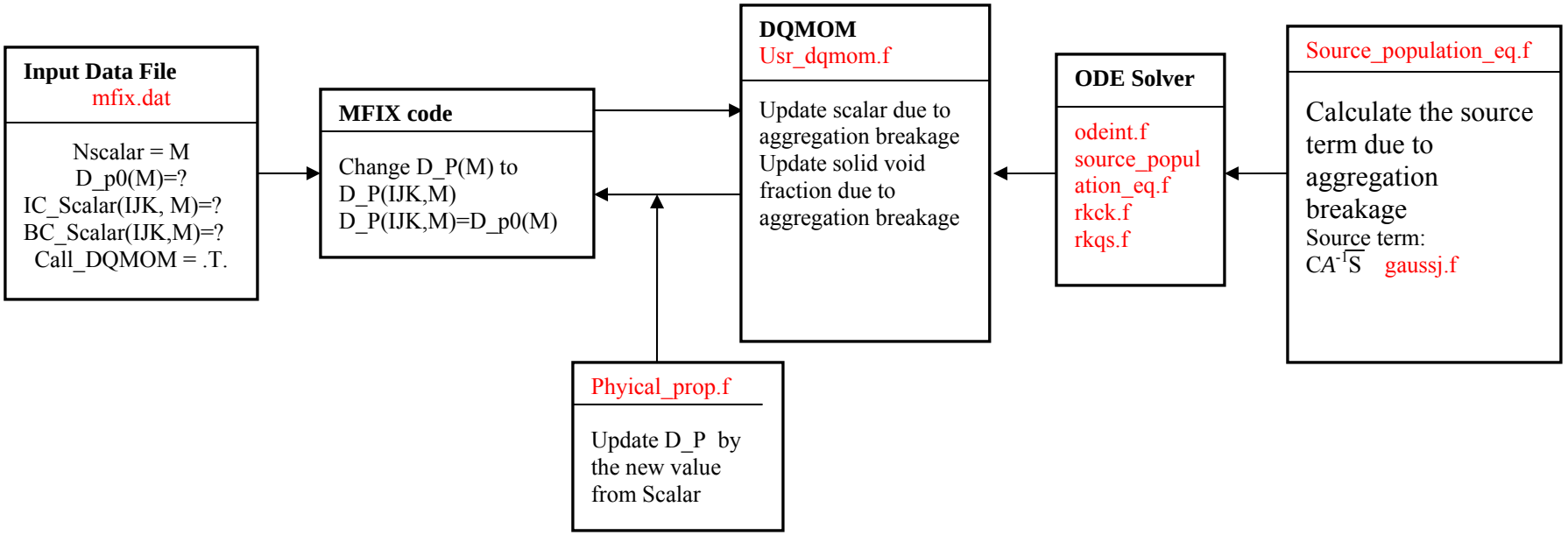


Figure 8: Flow chart for using DQMOM with MFIX.

```
Example mfix.dat
! !

! Run-control section

  RUN_NAME              = 'Binary2'
  DESCRIPTION           = 'DQMOM Test Case'
  RUN_TYPE              = 'new'
  UNITS                 = 'cgs'
  TIME                  = 0.0                      !start time
  TSTOP                 = 10                       ! stop time
  DT                    = 1.0E-5

  CLOSE_PACKED    = .TRUE.  .TRUE.
  ENERGY_EQ    = .FALSE.            !do not solve energy eq
  SPECIES_EQ   = .F. .F. .F.        !do not solve species eq
  MAX_NIT=1000
  DEF_COR =.TRUE.
  CALL_DQMOM=.T.                    ! solve population equation

  DISCRETIZE            =7*2
  DISCRETIZE(9)         =2

! Geometry Section

  COORDINATES           = 'cartesian'
  XLENGTH               = 10.1                     !radius
  IMAX                  = 15                       !cells in i direction
  YLENGTH               = 50.0                     !height
  JMAX                  = 50                       !cells in j direction
  NO_K                  = .TRUE.                   !2D, no k direction


! Gas-phase Section

  MU_g0                 = 1.8E-4               !constant gas viscosity
  MW_avg                = 29.0


! Scalar section
  Nscalar               = 2   ! use scalar solver to solve PBE.
  phase4scalar(1)       = 1
  phase4scalar(2)       = 2

! Solids-phase Section
  MMAX                  = 2
  RO_s(1)               = 2.53d0             !solids density
  RO_s(2)               = 2.53d0
  NMAX(1)               = 0
  NMAX(2)               = 0
  C_f                   = 0.1
  D_p0(1)                = 0.0164536d0         !small particle diameter
  D_P0(2)                = 0.0408151d0         !large particle diameter

  e                     = 0.80d0              !restitution coefficient
  Phi                   = 30.0             !angle of internal friction
  EP_star               = 0.376d0!void fraction at minimum fluidization
```

```
  Segregation_slope_coefficient=2.0e+5
  aggregation_eff       =0.001
  breakage_eff          =0.0001

! Initial Conditions Section

    ! 1. bed
  IC_X_w(1)             =   0.0           !lower half of the domain
  IC_X_e(1)             =   10.1          ! 0 < x < 7, 0 < y < 50
  IC_Y_s(1)             =   0.0
  IC_Y_n(1)             =   15.9         !initial values in the region
  IC_EP_g(1)            =  0.37          !void fraction
  IC_ROP_s(1,1)         =  0.6903
  IC_ROP_s(1,2)         =  0.9036
  IC_Scalar(1,1)        =  0.0164536d0
  IC_Scalar(1,2)        =  0.0408151d0
  IC_U_g(1)             =  0.0            !radial gas velocity
  IC_V_g(1)             =  @(20.0/0.37)   !axial gas velocity
  IC_U_s(1,1)           =  0.0            !radial solids velocity
  IC_V_s(1,1)           =  0.0            !axial solids velocity

  IC_U_s(1,2)           =  0.0            !radial solids velocity
  IC_V_s(1,2)           =  0.0

  IC_P_g(1)             =1010000
  IC_T_g(1)             =298
    !  2. Freeboard
  IC_X_w(2)             =   0.0         !upper half of the domain
  IC_X_e(2)             =   10.1
  IC_Y_s(2)             =  15.9
  IC_Y_n(2)             =  50

  IC_EP_g(2)            =   1.0d0
  IC_Scalar(2,1)        =  0.0164536d0
  IC_Scalar(2,2)        =  0.0408151d0
  IC_U_g(2)             =   0.0
  IC_V_g(2)             =   20.0
  IC_U_s(2,1)           =   0.0
  IC_V_s(2,1)           =   0.0
  IC_U_s(2,2)           =   0.0
  IC_V_s(2,2)           =   0.0

  IC_P_g(2)             =1010000
  IC_T_g(2)             =   298
!  Boundary Conditions Section

      ! 1. Distributor flow
  BC_X_w(1)             =  0.0                !gas distributor plate
  BC_X_e(1)             =  10.1
  BC_Y_s(1)             =  0.0
  BC_Y_n(1)             =  0.0

  BC_TYPE(1)            = 'MI'               !specified mass inflow

  BC_EP_g(1)            =  1.0
  BC_U_g(1)             =  0.0
```

```
  BC_V_g(1)             =  20.0
  BC_P_g(1)             = 1010000
  BC_T_g(1)             =  298
  BC_Scalar(1,1)        = 0
  BC_Scalar(1,2)        = 0

! 2. Exit
  BC_X_w(2)             =   0.0            !top exit
  BC_X_e(2)             =   10.1
  BC_Y_s(2)             =  50
  BC_Y_n(2)             =  50

  BC_TYPE(2)            =  'PO'          !specified pressure outflow
  BC_P_g(2)             =  1010000
  BC_T_g(2)             =  298
  BC_Scalar(2,1)        =  0
  BC_scalar(2,2)        =  0
!
!  Output Control
!
  OUT_DT                = 10.   !write text file BUB02.OUT every 10 s
  RES_DT                = 0.01  !write binary restart file
                                !BUB02.RES every 0.01 s
  NLOG                  = 25    !write logfile BUB02.LOG
                                !every 25 time steps
  FULL_LOG              = .TRUE.  !display residuals on screen


!SPX_DT values determine how often SPx files are written.  Here
!BUB02.SP1, which contains void fraction (EP_g), is written every
!0.01s, BUB02.SP2, which contains gas and solids pressure (P_g,
!P_star), is written every 0.1 s, and so forth.

SPX_DT = 0.1 0.1      0.1  0.1  0.1      100. 100.     0.1     0.1


!  Sweep Direction

  LEQ_SWEEP(1) = 'ISIS'
  LEQ_SWEEP(2) = 'ISIS'
  LEQ_SWEEP(3) = 'ISIS'
  LEQ_SWEEP(4) = 'ISIS'
  LEQ_SWEEP(5) = 'ISIS'
  LEQ_SWEEP(6) = 'ISIS'
  LEQ_SWEEP(7) = 'ISIS'
  LEQ_SWEEP(8) = 'ISIS'
  LEQ_SWEEP(9) = 'ISIS'

 NODESI=1 NODESJ=1 NODESK=1
```