DOE / METC-95 / 1013
(DE95000031)

MFIX Documentation
User's Manual

November 1994

U.S. Department of Energy
Office of Fossil Energy
Morgantown Energy Technology Center
Morgantown, West Virginia

DOE/METC-95/1013
(DE95000031)
Distribution Category UC-103

MFIX Documentation
User's Manual

By
Madhava Syamlal

EG&G Technical Services of West Virginia, Inc.
990 Elmer Prince Drive
Morgantown, West Virginia 26505-3276
Under Contract No. DE-AC21-90MC26328

U.S. Department of Energy
Office of Fossil Energy
Morgantown Energy Technology Center
P.O. Box 880
Morgantown, West Virginia 26507-0880

November 1994

Contents

Page

Contents

(Continued)

Page

List of Figures

List of Tables

Executive Summary

MFIX (**M**ultiphase **F**low with **I**nterphase e**X**changes) is a general-purpose hydrodynamic model for describing chemical reactions and heat transfer in dense or dilute fluid-solids flows, which typically occur in energy conversion and chemical processing reactors. MFIX calculations give time-dependent information on pressure, temperature, composition, and velocity distributions in the reactors. The theoretical basis of the calculations is described in the MFIX Theory Guide (Syamlal, Rogers, and O'Brien 1993).

This report, which is the MFIX User's Manual, gives an overview of the numerical technique, and describes how to install the MFIX code and post-processing codes, set up data files and run MFIX, graphically analyze MFIX results, and retrieve data from the output files. Two tutorial problems that highlight various features of MFIX are also discussed.

1 Introduction

MFIX is a general-purpose hydrodynamic model for describing chemical reactions and heat transfer in dense or dilute fluid-solids flows, which typically occur in energy conversion and chemical processing reactors. MFIX is written in FORTRAN and has the following modeling capabilities: multiple particle types, three-dimensional Cartesian or cylindrical coordinate systems, uniform or nonuniform grids, energy balances, and gas and solids species balances. MFIX calculations give time-dependent information on pressure, temperature, composition, and velocity distributions in the reactors. With such information, the engineer can visualize the conditions in the reactor, conduct parametric studies and what-if experiments, and, thereby, assist in the design process.

The theoretical basis of MFIX is described in a companion report (Syamlal, Rogers, and O'Brien 1993). The governing equations are summarized in Appendix A.

Section 2 gives an overview of the numerical technique used in MFIX and the organization of the code. Appendix B gives diagrams of the coordinate system and the numerical grid used in MFIX and details about the code such as the definition of the major FORTRAN symbols, the purpose of the major subroutines, and a list of statement functions used in MFIX.

Section 3 describes the process of setting up and running MFIX and the post processing codes ANIMATE_MFIX and POST_MFIX, starting and restarting MFIX runs, monitoring error messages, obtaining convergence, and CPU and memory requirements.

Section 4 describes the input and output files of MFIX. The data required for making an MFIX run is supplied through an input data file called *mfix.dat*. The definitions of the keywords used in *mfix.dat* are given in Appendix D. Additional information (chemical reaction kinetics, special body-force specifications, etc.) may be supplied by modifying appropriate subroutines. The output data is stored in two text files and eight binary files.

Section 5 describes the post-processing program ANIMATE_MFIX. This program allows the user to create animations of MFIX results.

Section 6 describes the post-processing program POST_MFIX. This program enables the user to examine and print out user-specified MFIX variables. POST_MFIX can also create MFIX files for making special restart runs. User-defined subroutines may be incorporated into POST_MFIX to manipulate MFIX results, for example, to calculate the average axial pressure drop from the pressure distribution calculated by MFIX.

Section 7 is a tutorial for setting up a simulation of a bubbling fluidized bed. This example involves the solution of mass and momentum balance equations only. The details of writing the data file are shown in this tutorial.

Section 8 is a tutorial for setting up the simulation of a partial combustor. In this example, the mass, momentum, energy, and species balance equations are solved. The tutorial contains discussions on writing the reaction rates and other user-defined subroutines, using a nonuniform grid, using obstacles and internal surfaces for defining the reactor geometry, and writing the data file. The files discussed in this tutorial are given in Appendix C.

2 Overview of MFIX Program

The equations used in MFIX are described in detail by Syamlal, Rogers, and O'Brien (1993) and are summarized in Appendix A. A brief overview of the solution technique is provided in this section. Some of the details of the method may be found in O'Brien and Syamlal (1990). The essence of the method was developed by Harlow and Amsden (1975) and was implemented in the KFIX code (Rivard and Torrey 1977). The method was later adapted for describing gas solids flows at the Illinois Institute of Technology (Gidaspow and Ettehadieh 1983; Syamlal 1985).

2.1 Discretization

MFIX uses a staggered grid arrangement as shown in Figure 1. Scalars are stored at the cell centers and the components of velocity vector are stored at the cell faces. The equations for scalar quantities, pressure, void fraction, temperature, and mass fractions, are solved on the main grid. The equations for the velocity vector components are solved on staggered grids. Thus there are four grids used for the solution.

The convective terms in mass and momentum equations are upwind differenced. The energy and species balance equations are differenced in a non-conservative manner by subtracting the differenced mass balance equations from (upwind) differenced conservative forms of the equations.

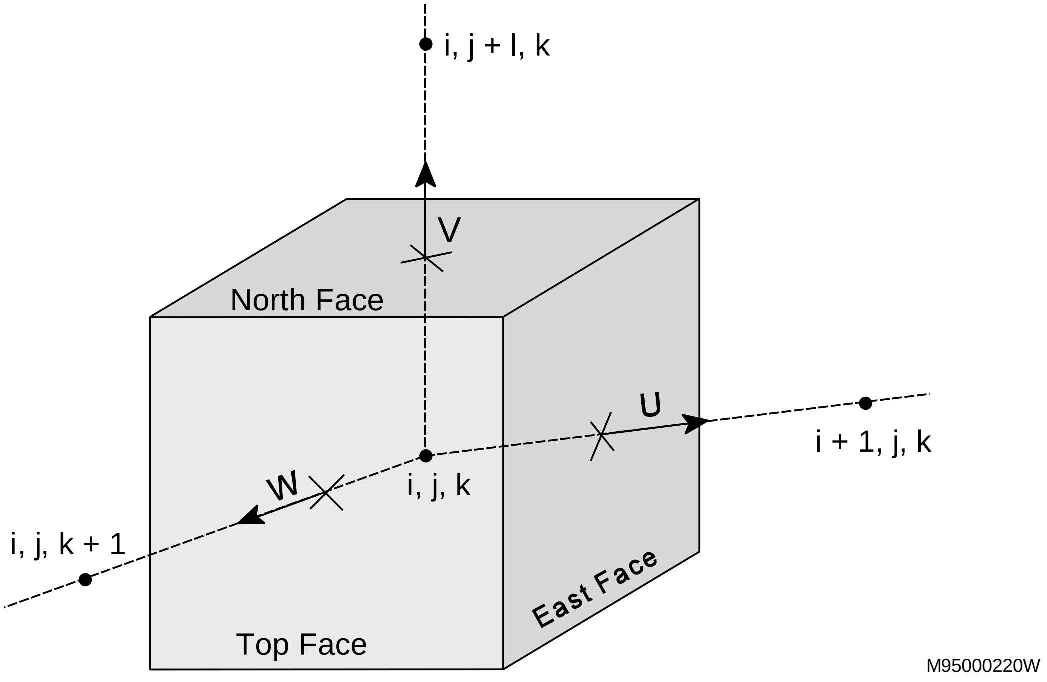


Figure 1. A Computational Cell

The mass balance equations are differenced fully implicitly. The convective and diffusive terms in the momentum equations are differenced explicitly. When the frictional stress formulation is turned on, however, the viscous stress terms for the solids are differenced implicitly. The pressure term and the drag term are treated implicitly. All the terms in the species balance and the energy equations are differenced implicitly.

2.2 Solution Technique

At the beginning of each time step all the explicit quantities are calculated. Then the difference equations are solved sequentially within the iteration loops shown in Figure 2. The cells are visited one at a time starting from the cell with the lowest index and going to the cell with the largest index. Convergence is said to occur when all the discretized equations are satisfied simultaneously in all the cells without having to make adjustments in any of the cells.

The solution starts with a guessed pressure field. First it is checked whether the gas continuity equation is satisfied. If not, the pressure is adjusted using a Newton's method or a secant method. If the solids are in the packed-bed range, a solids pressure is adjusted so that the void-fraction versus solids pressure function is satisfied.

All the reaction rates are calculated. If the energy or the species balance equations need to be solved, they are solved for temperatures and species mass fractions. The fluid density is updated. The coupled fluid and solids momentum equations are solved for the velocity components. The solids continuity equations are then solved for the solids volume fractions. The void fraction or the fluid volume fraction is then calculated by subtracting the sum of solids volume fractions from one.

The above steps complete one pass of the inner iteration for a cell. The inner iterations are continued until the fluid continuity equations and solids pressure equation (only for packed regions) are simultaneously satisfied or the number of inner iterations exceed a specified limit, which is typically five.

The inner iterations are continued for all the numerical cells to complete one pass of the outer iteration. The outer iterations are continued until the solution reaches the point where no inner iterations are needed, which implies that the existing solution satisfy all the discretized equations simultaneously in all the cells. When such a convergence is obtained or when the number of outer iterations exceeds a specified limit, which is typically 500, the calculations proceed to the next time step.

After the iterations for the fluid pressure and the solids pressure have converged, it is checked whether the solution satisfies the energy and the species balance equations. If the solution does not satisfy the energy or the species balance equations, no inner iterations are done but an outer iteration is forced.

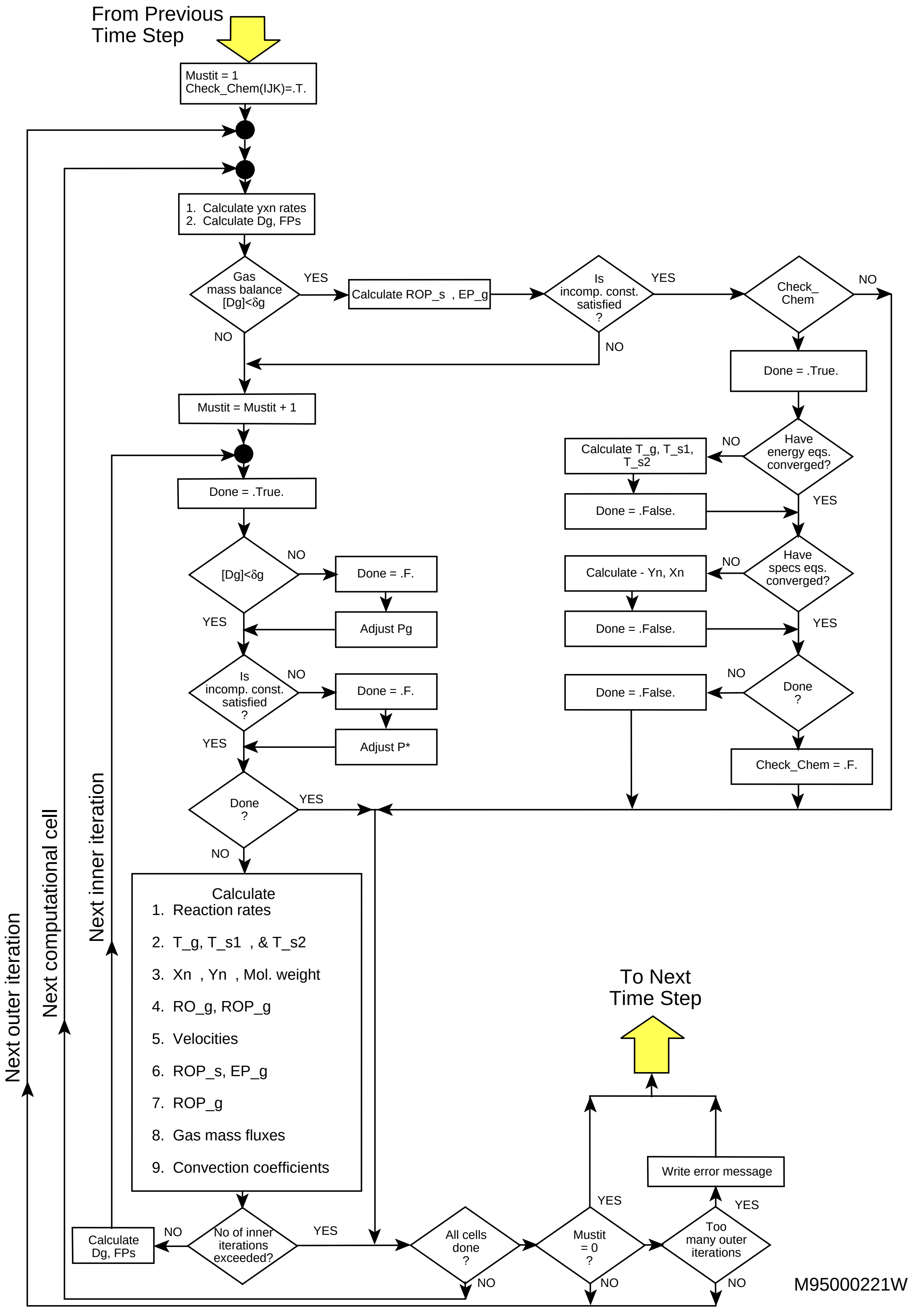


Figure 2. Iteration Scheme Used in MFIX

2.3 The Computer Program

MFIX is a modular computer program consisting of subroutines, each having only a single purpose. A list of the major subroutines and their purpose is given in Appendix B.

The names of the major variables correspond to the symbols used in the equations (Appendix A) and are listed in Appendix B. A subscript or superscript in the symbol is identified by an underscore, e.g., þ$_g$ = EP_g. An "o" represents division and "x" represents multiplication, e.g., $\frac{t}{x}$ = DToDX; $_g v_g$ = ROP_gxV_g.

If the user wants to modify MFIX routines, particular attention must be paid to the indices used in MFIX. The three-dimensional arrays (I,J,K) are stored as one-dimensional arrays with the index IJK, which is calculated using the function FUNIJK. The coefficients of the neighboring cells are obtained by calling the subroutine SET_INDEX1. The definitions of the frequently used indices are given in Appendix B.

MFIX uses several statement functions for programming convenience, and for minimizing programming errors. A list of frequently used statement functions is given in Appendix B. There are a number of averaging functions listed in the file *functions_avg1.inc* that the user may find useful for (spatially) averaging variables.

3 Setting Up and Running MFIX

The procedure for installing and running MFIX is given in this section. The commands shown here are for the UNIX operating system. Command procedures are available for UNIX only, but can be easily developed for other operating systems.

3.1 Reading the MFIX Tape

1. To read MFIX from a DAT tape, go to the home directory and type

   ```
   tar xvo
   ```

2. To read MFIX from a 3½" PC diskette, set the FTP to the binary mode, then download the file on the diskette to the workstation disk and give it the file name *mfix.tar.Z*. Uncompress the file by typing

   ```
   uncompress mfix.tar
   ```

   Read MFIX files by typing

   ```
   tar xvf mfix.tar
   ```

3. The above procedure will create a directory *mfix* that contains the four subdirectories *model*, *post_mfix*, *animate_mfix*, and *tutorials*. The subdirectory *model* contains the MFIX program files. The subdirectories *post_mfix* and *animate_mfix* contain the files for the respective post-processing programs. The subdirectory *tutorials* contains the files discussed in the tutorials in sections 7 and 8.

3.2 Installing the MFIX Code

1. ANIMATE_MFIX works only on Silicon Graphics workstations. An executable file is given in the *animate_mfix* directory. To make the file executable, type

   ```
   chmod u+x animate_mfix
   ```

   For convenience, create an alias by adding the following line to the *.login* file (assuming MFIX was installed in the home directory):

   ```
   alias animate_mfix ~/mfix/animate_mfix/animate_mfix
   ```

   Then the post-processor can be activated from any directory by typing

   ```
   animate_mfix
   ```

2. An executable file for *post_mfix* needs to be created. If user-defined post-processing is required, modify *usr_post.f* file in the directory *post_mfix* and make the executable file by typing

   ```
   make
   ```

   For convenience, create an alias by adding the following line to the *.login* file (assuming MFIX was installed in the home directory):

   ```
   alias post_mfix ~/mfix/post_mfix/post_mfix
   ```

   Then the post_processor can be activated from any directory by typing

   ```
   post_mfix
   ```

3. An MFIX executable file needs to be created for each of the runs. For convenience, create an alias by adding the following line to the *.login* file (assuming MFIX was installed in the home directory):

   ```
   alias make_mfix ~/mfix/model/make_mfix
   ```

   For the first time, the user should logout and login or type

   ```
   source .login
   ```

   so that the aliases take effect.

3.3 Starting a New Run

1. A separate subdirectory should be created for each run. Write or modify user-defined files in the run directory. (See section 4.2.)

2. Copy *param.inc* from the mfix directory into the run directory and change the array dimensions in the file.

3. Write an MFIX data file and name it *mfix.dat*.

4. Create an MFIX executable file by typing

   ```
   make_mfix
   ```

   Answer **y** to the question "Do you need any MFIX file from the run-directory?" so that *param.inc* and other user-defined files are copied into the *model* directory. The MFIX executable file *mfix.exe* will be created and copied into the run-directory.

5. Run MFIX in the background by typing

mfix.exe &

6. Keep track of the progress of the run by looking at the end of the *.LOG* file, e.g., type

   tail *.LOG

7. MFIX results can be retrieved, even while the run is in progress. To activate visualization, run *animate_mfix* by typing

   animate_mfix

   The program will create the two files *animate_mfix.control* and *animate_mfix.minmax* for storing some parameters.

8. For retrieving and manipulating data and for creating special restart files, run the post_processor *post_mfix* by typing

   post_mfix

3.4 MFIX Messages

MFIX reports errors while reading the data file, while processing input data, and during the run time. Errors in reading the data file and in opening files are reported to the terminal. All other errors are reported in the .LOG file.

While reporting errors in reading the data file, MFIX displays the offending line of input, so that the error can be easily detected. The possible causes of error are incorrect format for the name-list input, unknown (misspelt) variable name, or the dimension of the name list item is too small. For example, if the dimension of DX is 10, and if the input data file contains an entry DX(11), MFIX will report an input processing error.

While processing the input data MFIX will report errors if the data specified is insufficient or physically unrealistic. MFIX will supply default values only when the code is certain that the default value is correct.

An occasional input-processing error is the inability to determine the flow plane for a boundary condition. The boundary planes defined in the input data file must have a wall-cell on one side and a fluid-cell on the other side. If the initial condition is not specified for the fluid-cell, MFIX will not recognize the cell as a fluid-cell and, hence, MFIX will be unable to determine the flow plane.

During the run time, MFIX will report any instance of non-convergence encountered. The message will show a "% converged" value based on the ratio of the number of cells converged to the total number of fluid-cells.

Every NLOG (see Appendix D) number of time steps, MFIX monitors whether the mass fractions add up to 1.0, the overall reaction rates add up to zero, the viscosities, conductivities, and specific heats are greater than zero, and the temperatures are within the specified limits. A message will be printed out if any errors are encountered. The run may be aborted depending upon the severity of the error. Every NLOG time steps, MFIX will print out the number of iterations during the previous time step and the total solids inventory in the reactor.

A message is written to the *.LOG* file whenever the *.RES* and *.SPx* files are written.

For the specified mass-outflow condition, after the elapse of time BC_DT_0, MFIX prints out time-averaged mass flow rates. For cyclic boundary conditions, MFIX will print out the volume averaged mass fluxes every NLOG time steps.

3.5 Restarting a Run

There are four types of restarts that can be selected with the RUN_TYPE specification (see Appendix D). The most common type is 'restart_1', which allows the user to continue a run that was stopped. A run is restarted by modifying *mfix.dat* so that RUN_TYPE is 'restart_1' and TIME is the last time at which *.RES* file was updated (see the end of *.LOG* file). Delete the old *.OUT* file. Delete the *.LOG* file or rename it, for example, *log1*, *log2*, etc. Restart the run by typing

```
mfix.exe &
```

3.6 Convergence and CPU Time

To get initial convergence, it is necessary to specify a reasonable initial condition. For example, using a hydrostatic rather than a uniform pressure distribution as an initial condition will aid convergence in fluidized-bed simulations. If a uniform pressure distribution is used as the initial condition instead, setting the reactor pressure slightly larger than the pressure at the exit may aid convergence.

The convergence of the iterations is limited by the convective time step. As a rule of thumb, the time step should be less than about a tenth of the convective time scale; i.e.,

$$t \quad 0.1\,\frac{smallest\ cell\ size}{largest\ velocity} \quad . \tag{1}$$

Time steps usually encountered are in the range of 10^{-3} to 10^{-4} s. Sometimes the time step may be controlled by the time scale of the fastest reaction. With combustion reactions, the time step may become as low as 10^{-5} s. Some of the initial non-convergence or occasional non-convergence may be ignored. If non-convergence is persistent, the time step should be reduced till convergence can be obtained. If the time step is considerably smaller than the ranges indicated above, the problem specification may be erroneous, in which case reducing the time step will not help.

As a rule of thumb, the run time can be minimized by choosing a time step such that the number of iterations per time step is between 10 and 50.

The use of uniform grids with an aspect ratio of 1 gives the best results. However, it is often necessary to use nonuniform grids to capture fine details of the flow, yet minimize the number of numerical cells. For nonuniform grids, the sizes of adjacent cells should not differ by more than 20 percent. Also, the aspect ratio of the sides of the cell should be less than 5:1. For flows aligned with one of the grid directions, this restriction on the aspect ratio may be relaxed.

When nonuniform grids are used, the following formulas are useful for doing mesh size calculations. For a starting location X_0 with a mesh size dX_0 and an expansion (contraction) ratio *f* such that X_1 = X_0 + dX_0 þ f, X_2 = X_1 + dX_0 þ f þ f, ... we get

$$X_n \quad X_0 \quad dX_0 \, \frac{f^{\,n\;1} \quad f}{f \;\; 1} \quad . \tag{2}$$

Often we would like to know the expansion (contraction) ratio *f* and the number of cells required to go from X_0 to X_n when the grid size changes from dX_0 to dX_n. Those quantities are given by

$$f \quad \frac{1}{1 \quad \dfrac{dX_n \;\; dX_0}{X_n \;\; X_0}} \quad , \tag{3}$$

and

$$n \quad \frac{\log\,(dX_n\,/\,dX_0)}{\log f} \quad . \tag{4}$$

3.7 Disk Space Requirements

The amount of disk storage required for SPx files only are shown below, since they account for most of the storage. The disk space required per variable in Megabytes, Mbv, is given by

$$Mbv \quad \frac{4\,(IMAX \;\; 2)\,(JMAX \;\; 2)\,(KMAX \;\; 2)}{2^{20}} \quad . \tag{5}$$

If any of the directions has been turned off (e.g., No_K = .T.), set the dimension for that direction to 1 in the above formula.

Then the disk space required for the various SPx files (in Megabytes) will be slightly greater than the following:

SP1	Mbv x (1 + TSTOP/SPX_DT(1));
SP2	2 x Mbv x (1 + TSTOP/SPX_DT(2));
SP3	3 x Mbv x (1 + TSTOP/SPX_DT(3));
SP4	3 x Mbv x MMAX x (1 + TSTOP/SPX_DT(4));
SP5	Mbv x MMAX x (1 + TSTOP/SPX_DT(5));
SP6	3 x Mbv x (1 + TSTOP/SPX_DT(6)); and
SP7	$\sum_{m=0}^{MMAX} NMAX(m)$ x Mbv x (1 + TSTOP/SPX_DT(7)).

4 Input and Output Files of MFIX

An MFIX run is controlled with the help of an input file called *mfix.dat* and several user-defined FORTRAN files. MFIX produces two text files that the user may access directly and eight binary files that the user accesses through the two post-processing codes. The user interface of MFIX is schematically shown in Figure 3.

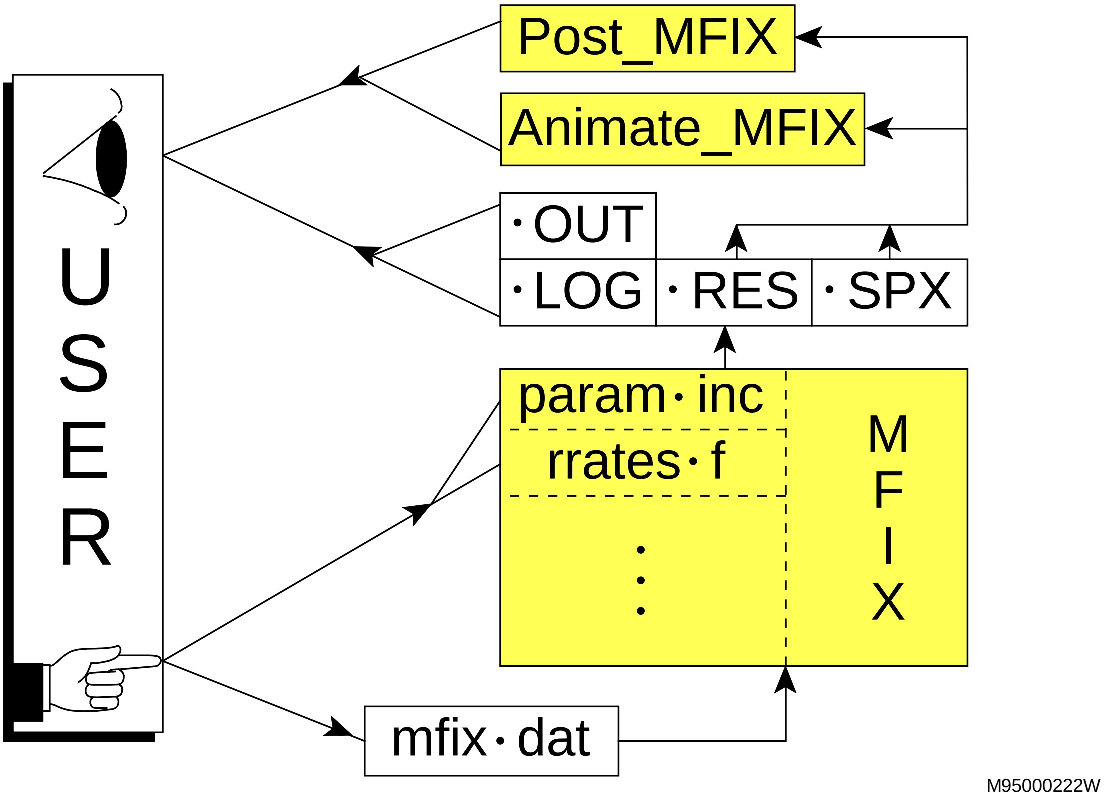


Figure 3. User Interfaces of MFIX

4.1 The Input Data File

The input data file *mfix.dat* uses a NAMELIST format. A NAMELIST input has the form "Variable name = Value." The variable names and descriptions are given in Appendix D.

Each data input line is limited to 80 columns. If any data appears in columns 81 to 132, MFIX will print an error message and stop execution.

Any number of comment or blank lines are allowed in the data file. Comments start with a "#" or a "!" and may appear anywhere in the data file.

When dimensioned variables are written on multiple lines, the variable name should be repeated at the beginning of each line with the correct index. An index of 1 is the default. Thus, for example, DX(10) may be written in two lines as

```
DX    = 1.  1.  1. 0.5  0.5  0.5  1.
DX(8) = 1.  1.  1.
```

The starting index in the first line is assumed to be 1, whereas in the subsequent lines the starting index needs to be explicitly stated, as, for example, 8 in the second line above. Also, the above data may be entered in a condensed form as

```
DX    = 3*1.  3*0.5  4*1.
```

For multiply dimensioned key words, the indices need to be explicitly stated. For example,

```
IC_ROP_s (1, 1)      =        0.4      0.5
```

will be read as

IC_ROP_s(1,1) = 0.4 and IC_ROP_s(2,1) = 0.5.

Some simple formulas may be entered in the data file to improve the readability of the data file. The formulas are enclosed by the strings "@(" and ")". The only symbols recognized are "PI", "*", "/", and unary minus. Addition (+) and subtraction (-) are not recognized. Scientific notation for numbers is recognized. For example, a quantity such as 2þ/3 can be entered as @(2*PI/3). Not more than four or five formulas should be entered on the same line.

The variable names may be entered in any order. For the ease of writing and reading the data file, however, a particular grouping of data items is suggested in Appendix D. For details of the various data items, refer to Appendix D.

4.2 User-Defined Files

In addition to writing the data file *mfix.dat*, the user will occasionally need to modify some of the MFIX files. The files are listed below, somewhat in the order of decreasing frequency with which the user may change the files.

1. *param.inc*:
   The array dimensions in this file need to be modified to match the problem specification. MFIX will run more efficiently this way, although the array dimensions could be larger than needed for the problem. If the array dimensions are too small, MFIX will print an error message and abort.

2. *calc_physprop.f*:
   The fluid viscosity, fluid and solids specific heat and thermal conductivity, and wall heat transfer coefficient specified in this file may be modified.

3. *rrates.f*:
   All the reaction kinetic expressions and heats of reaction are given in this file. Instructions for writing the file are given as comments in the file. See section 8 for an example of writing this file.

4. *usr0.f*:
   If CALL_USR is true, the subroutine USR0 in this file is called once by subroutine TIME_MARCH, before the time loop. USR0 should be used for initializing user-defined variables other than namelist variables.

   This file should also be used for including other user-defined FORTRAN files. For MFIX to recognize user-defined include files, their file-name modifier should be .inc. The user-defined include files should be named differently from the MFIX include files.

5. *usr1.f*:
   If CALL_USR is true, the subroutine USR1 in this file is called once every time step by subroutine TIME_MARCH, at the beginning of the time loop. USR1 should be used for setting time-dependent quantities or for writing out information for debugging purposes.

6. *usr2.f*:
   If CALL_USR is true, the subroutine USR2 in this file is called by subroutine ITERATE at the beginning of every outer iteration. USR2 should be used for setting quantities that need to be updated every iteration or for writing out information for debugging purposes.

7. *usr_init_namelist.f*:
   The subroutine USR_INIT_NAMELIST in this file is used for initializing user-defined namelist variables, if needed.

8. *usr_namelist.inc*:
   This file is for defining user-defined namelist variables and for storing user-defined common blocks.

9. *usr_write_out0.f*:
   If CALL_USR is true, the subroutine USR_WRITE_OUT0 in this file is called once for writing quantities to the *.OUT* file.

   Statements for opening file units for user-defined outputs should be included in this subroutine. FORTRAN file units should be opened in the range 7 to 49 or greater than 70. File units 50 to 70 are used by MFIX.

10. *usr_write_out1.f*:
If CALL_USR is true, the subroutine USR_WRITE_OUT1 in this file is called at intervals OUT_DT specified in the input data file. If OUT_DT is undefined, this subroutine will not be called.

11. *write_usr0.f*:
This subroutine is called once from the main program.

12. *write_usr1.f*:
This program is called every USR_DT intervals, which is supplied from the input data file. The number of such user-defined outputs are specified by the parameter DIMENSION_USR in the file *param.inc*.

13. *body_force.inc*:
This file is used for defining body forces. The default specification in MFIX is that of gravity acting in the negative y direction.

14. *eosg.f*:
The function EOSG is used for specifying the equation of state for a gas. The default is the ideal gas law.

15. *tolerance.inc*:
This file contains the tolerances used for judging convergence, the upper and lower bounds for temperatures, and the limits for inner and outer iterations.

    TOL_DG may be increased to obtain convergence in fewer iterations and, thus, to make the run faster. The gas mass balance, however, will be adversely affected by this.

4.3 Output Files

1. *.LOG*:
The *.LOG* text file contains error messages, information on convergence and number of iterations, and messages about the output files being written.

2. *.OUT*:
The first part of an *.OUT* text file echoes the input specified by *mfix.dat* and default values specified in MFIX. If OUT_DT is defined, the file also contains data on field variables written at intervals OUT_DT. For problems using a large number of nodes, it is recommended that OUT_DT not be defined, to prevent the *.OUT* file from becoming voluminous. A similar text file of field-variable data may be written by using Option 5 of POST_MFIX.

    When a new problem is being set up, run MFIX with TSTOP=0 and OUT_DT specified. MFIX will then read the data file, write an *.OUT* file, and stop. Examine the *.OUT* file to verify that the input data has been entered correctly. Then start the MFIX run again by setting TSTOP to the appropriate value and leaving OUT_DT unspecified.

Pay particular attention to the map of cell flags in the *.OUT* file to ensure that initial condition regions, boundary conditions, and internal surfaces are at the correct locations. The initial and boundary conditions specified are shown in the following map. Each computational cell is represented by a string of three characters. The first character represents the type of cell, and the last two characters give a number that identifies a boundary or initial condition. For example, .02 indicates a cell where Initial Condition No. 2 will be specified. Only the last two digits are written, although the number of initial and boundary conditions are only limited by the DIMENSION_IC and DIMENSION_BC statements in *param.inc.* Hence, for example, Condition No. 12, 112, 212, etc., will be represented only as 12.

First Character	Description
.	Initial condition
W	No slip wall
S	Free-slip wall
c	Cyclic boundary
C	Cyclic boundary with pressure drop
I	Specified mass-flux inflow cell
O	Outflow cell
p	Specified pressure inflow cell
P	Specified pressure outflow cell

Internal surfaces at East, North, or Top of a cell is represented by the following letters to the right of the three-character string:

Side	Impermeable	Semipermeable
East	E	e
North	N	n
Top	T	t

For cells with internal surfaces on more than one side, the characters will be overwritten in the above order.

3. *.RES*:
The *.RES* file is a double precision binary file containing all the data required for restarting a run. The file is updated at intervals RES_DT. Only the most recent data set is retained in this file.

4. *.SPx*:
The *.SPx* files are single precision binary files containing data for various field variables as shown in Table 1. The files are augmented at intervals SPx_DT. The files retain all the data sets written to them.

Table 1. Variables Stored in SPx Files

.*SPx*	Variable (FORTRAN symbol)
.*SP1*	þ_g (EP_g)
.*SP2*	P_g, P^* (P_g, P_star)
.*SP3*	u_g, v_g, w_g (U_g, V_g, W_g)
.*SP4*	u_{sm}, v_{sm}, w_{sm} (U_s, V_s, W_s)
.*SP5*	þ_{sm} þ_{sm} (ROP_s)
.*SP6*	T_g, T_{s1}, T_{s2} (T_g, T_s1, T_s2)
.*SP7*	X_{gn}, X_{smn} (X_g, X_s)

5 ANIMATE_MFIX: Graphical Representation of Data

Graphical representation of MFIX results can be obtained by using the ANIMATE_MFIX program. This program should be accessed from the run directory, which contains the *.RES* and *.SPx* files. First, the program will request a run name. After receiving a valid run name, the program will check whether the two files *animate_mfix.control* and *animate_mfix.minmax* exist in the run directory. If these files are not found in the directory, certain default values are used for configuring the ANIMATE_MFIX display and the files are written with these default values.

The entries in *animate_mfix.control* are as follows:

1. Run name
2. Time-start, Time-end
3. Full screen (1) or small screen for VCR (0)
4. Title

5-7. Specify the three lines as F.

The entries in *animate_mfix.minmax* give the ranges of the variables plotted, as labelled in the file. These ranges may be changed while running the program. The most recent value of the ranges will be stored in the file when ANIMATE_MFIX is exited.

The screen display given by ANIMATE_MFIX is schematically shown in Figure 4. At the top of the screen is a control panel that is activated by pointing (moving the arrow) to the control buttons and clicking the left mouse button. The panel below it shows the title on the left side and the time on the right side. Below that are the three main display panels, which allow simultaneous two-dimensional views of up to three variables. The panels display the values plotted using a color scale on X-Y planes at specified Z values. At the bottom of the screen are panels showing the variable name, color scale, maximum and minimum values on the color scale, the i,j,k indices of a location in the reactor, the value of the variable at that location, and a plot showing time versus variable at that location.

Click the left mouse button to select the options from the control panel at the top of the screen. The options are listed below:

Top row:

grid — Turn on or off the grid display. Usually the plot needs to be magnified to see the grid clearly.

symmetry — Turn on or off display with reflection across the line of symmetry on the left. This option is useful for plotting values from calculation in cylindrical coordinates.

freeze/go — Stop or start animation. When the animation is stopped, the displays can be advanced frame-by-frame by clicking the left mouse button anywhere other than in the control panel.

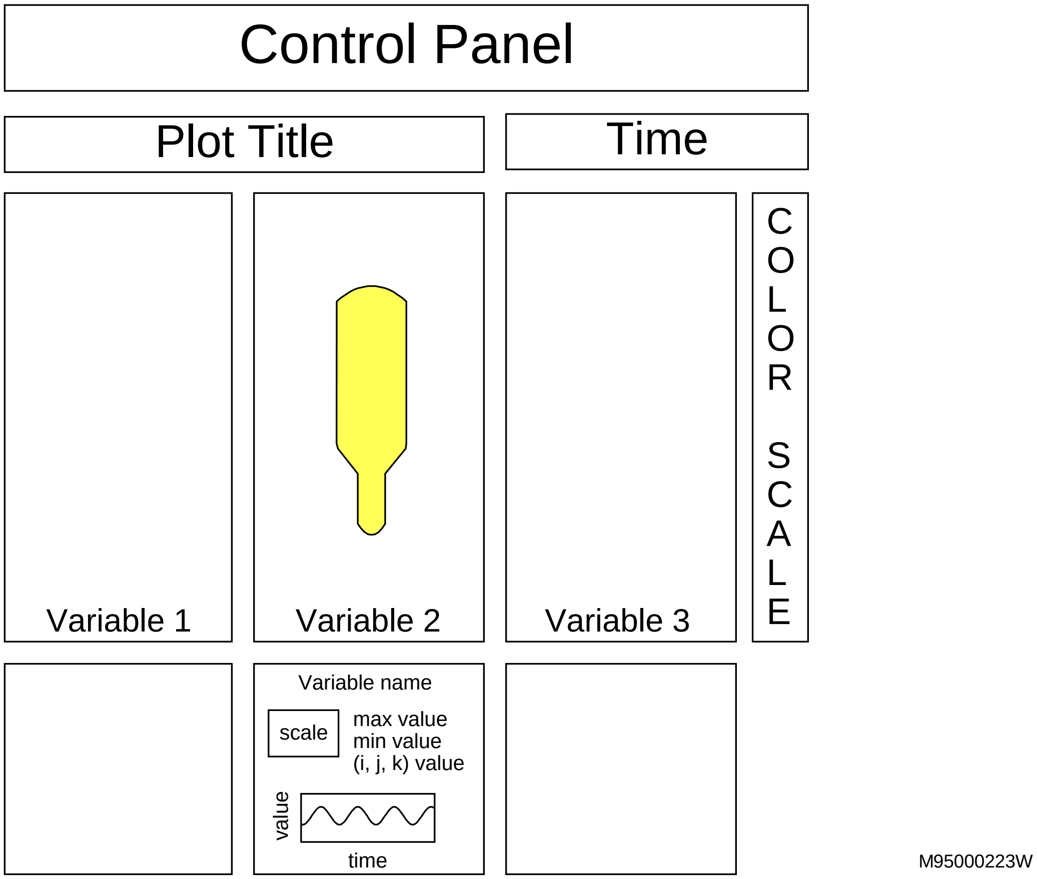


Figure 4. ANIMATE_MFIX Display

y-scale + Magnify only the vertical direction. The x:y aspect ratio of the plot will not be correct.

x-scale + Magnify only the horizontal direction. The x:y aspect ratio of the plot will not be correct.

xy-scale + Magnify the plot in both the directions. Use the arrow keys to change the displayed region.

reset Reset the plot to original size.

Middle row:

set time Select a specified time by entering a floating point number from the keyboard.

mod var 1 Select the variable to be displayed on the left panel. Highlight the variable to be selected and click the right mouse button.

mod var 2 Select the variable to be displayed on the middle panel.

mod var 3 Select the variable to be displayed on the right panel.

i,j,k Enter the i, j, and k values for the time versus variable plot shown in the bottom panel.

mod(i,j,k) Alternatively, select the location in the reactor for the time versus variable plot by clicking the left mouse button.

scales Specify the range of variables used in the color plots. The latest values will be stored in the *animate_mfix.minmax* file.

Bottom row:

MISC menu Select parameters for plotting time-averaged values.

xy-scale - Reduce the plot in both directions.

INDEX menu Change values of the k(or Z)-slice, solids phase number, and gas and solids species number.

vectors Toggle vector plots on or off. The gas velocity vectors will appear on the left panel and solids velocity vectors will appear on the right panel. The "Mod NVEC_X" and "Mod NVEC_Y" in the menu allows the skipping of data points; for example, if Mod_NVEC_X=3, then only every third vector in the X direction will be plotted. The "Mod FACTOR..." in the menu will allow the user to increase or decrease the length of the vectors. With the "Toggle Vector Scaling" option, vector scaling may be turned on or off. If the vector scaling is turned off, all the vectors will have the same length and the plots will represent only the direction of the velocity, not the magnitude.

grid only Turn on or off the plotting of the variable on the grid, when the grid is being plotted.

snapshot Create RGB files of the active panels. The RGB files are sequentially named as *animate_mfix_xx0000gb*, etc.

exit Exit ANIMATE_MFIX program.

6 POST_MFIX: Data Retrieval and Manipulation

The post-processing program POST_MFIX is used for retrieving data from the *.RES* and *.SPx* files. POST_MFIX is compiled with fairly large array sizes, and hence it will work for most MFIX runs. If the array sizes are too small, POST_MFIX will give an error message, and the code will need to be recompiled with adequate array dimensions.

To start post processing, go to the run directory and run POST_MFIX by typing

post_mfix

```
0   - Exit POST_MFIX
1   - Examine/print data
2   - Write .RES from data in .SPx files
3   - Interpolate and write .RES
4   - Calculate miscellaneous quantities
5   - Print out variables
6   - Call user defined subroutine
      USR_POST
```

POST_MFIX Main Menu

The code will first ask for the RUN_NAME. If any output files are not found, the code will report that error. Then a main menu as shown above will appear, from which the options are selected by typing the appropriate number. The various options in POST_MFIX are discussed below.

6.1 Examining Data

Option 1 from the main menu starts an interactive program for examining MFIX data and for writing data into text files. A sample interactive session is shown on the right. In this example, time-averaged data from time 0.5s to 0.6s is requested. The variable selected is the axial gas velocity (V_g). When the range of I indices is given as 2 to 10, POST_MFIX reminds the user that the maximum I in this run is 9. No averaging over I (or radial direction) is desired. The velocity profile is desired at an axial location corresponding to J=10. Since this is a two-dimensional run, there is no need to specify the azimuthal location. By specifying the file name as *, the data is displayed at the terminal. (If the file name is specified as !, POST_MFIX goes to the beginning of Option 1 menu without retrieving data.)

```
Enter menu selection > 1

Interactive data retrieval program.
Type ? any time for help, or press
RETURN to select default values
shown in parenthesis.

 Time: (   0.000,   0.000) > .5 .6
 Time average ? (N) > y
 Variable: (EP_g     ) >V_g
 I range: (  1,  1) >2 10
 I1 and I2 should be in the range 1
 to   9
 I range: (  2, 10) >2 8
 Average or sum over I? (N) >
 J range: (  1,  1) >10 10
 K range: (  1,  1) >
 File: (*                      ) >
```

POST_MFIX Option 1 Menu

The data retrieved by POST_MFIX is shown on the right. POST_MFIX displays the physical coordinates that are constant, the time, variable name, and the retrieved data as a table. Since this is a two-dimensional run, the Z location printed out should be ignored.

After printing out the data, POST_MFIX returns to the interactive menu. Now the default values are the values chosen during the previous data retrieval.

The valid variable names for POST_MFIX are shown here and on the following page. The minimum and maximum values of these variables in a range may be determined by typing the variable name preceded by a 0 or a 1.

6.2 Writing Special Restart Files

Sometimes it is necessary to restart a run from an earlier time. For example, consider a run that aborts at 5.3s and the data in *.RES* file is so corrupted that the run cannot be restarted even with a smaller time step. Then the user might wish to restart the run from an earlier time, say 5.0s, with a smaller time step. It is not possible to do this directly, since the *.RES* file retains only the most recent data, which happens to be for 5.3s. Option 2 becomes useful in such cases, since it enables the user to write a *.RES* file based on data for 5.0s from the *.SPx* files. Option 2 will modify *.RES* and *.SPx* files such that the run is set back to the specified time.

```
   Y =   9.0000
   Z =  -3.1416
   Time average of V_g      from
 0.50000     to  0.60000
   Time =  0.60000
        X             V_g
    0.50000        65.937
     1.5000        63.214
     2.5000        59.819
     3.5000        56.492
     4.5000        53.443
     5.5000        53.207
     6.5000        52.894
```

Data Retrieved by POST_MFIX

```
    EP_g      -  Void fraction
    P_g       -  Gas pressure
    P_star    -  Solids pressure
                 (frictional regime)
    U_g       -  X component of gas
                 velocity
    V_g       -  Y component of gas
                 velocity
    W_g       -  Z component of gas
                 velocity
    U_s       -  X component of solids
                 velocity
    V_s       -  Y component of solids
                 velocity
    W_s       -  Z component of solids
                 velocity
    ROP_s     -  Solids density x volume
                 fraction
    T_g       -  Gas temperature
    T_s1      -  Solids-1 temperature
    T_s2      -  Solids-2 temperature
    X_g       -  Gas species mass
                 fraction
    X_s       -  Solids species mass
                 fraction
```

Field Variable Names in POST_MFIX

Option 3 in the POST_MFIX menu allows the user to create a new *.RES* file from an existing *.RES* file. When the new *.RES* file is written, the user is able to refine or coarsen the grids and add or delete a coordinate dimension. The user is also able to increase or decrease the number of gas or solids species. For example, this option will be used to obtain the initial conditions for a three-dimensional simulation from the results of a two-dimensional simulation or to obtain the initial conditions for a simulation considering chemical reactions from the results of a simulation without chemical reactions. To use this option, first write a new *mfix.dat* file containing all the desired modifications in the geometry and the number of species. POST_MFIX will read this *mfix.dat* file to determine the conditions for the new run and write the *.RES* file accordingly.

6.3 Other POST_MFIX Options

Option 4 allows the user to print a text file of granular viscosity and temperature calculated from the data in *.RES* file, to write a series of data sets of granular viscosity and temperature in the format of the *.SP1* file, to calculate time-averaged gas and solids fluxes, or to calculate time-averaged void fraction and axial gas velocity and the standard deviation of void fraction and axial gas velocity fluctuations.

Option 5 allows the user to get a text file containing the field variables just as in the *.OUT* file. The print out can be obtained for data read from the *.RES* file or for selected times from the *.SPx* files.

If a user-defined subroutine exists, it can be accessed by using Option 6.

XFLOW_gx	-	Gas species flow rate in X direction
XFLOW_sx	-	Solids species flow rate in X direction
XFLOW_gy	-	Gas species flow rate in Y direction
XFLOW_sy	-	Solids species flow rate in Y direction
XFLOW_gz	-	Gas species flow rate in Z direction
XFLOW_sz	-	Solids species flow rate in Z direction
MFLOW_gx	-	Mass flow rate of gas in X direction
MFLOW_sx	-	Mass flow rate of solids in X direction
MFLOW_gy	-	Mass flow rate of gas in Y direction
MFLOW_sy	-	Mass flow rate of solids in Y direction
MFLOW_gz	-	Mass flow rate of gas in Z direction
MFLOW_sz	-	Mass flow rate of solids in Z direction
VFLOW_gx	-	Volumetric flow rate of gas in X direction
VFLOW_sx	-	Volumetric flow rate of solids in X direction
VFLOW_gy	-	Volumetric flow rate of gas in Y direction
VFLOW_sy	-	Volumetric flow rate of solids in Y direction
VFLOW_gz	-	Volumetric flow rate of gas in Z direction
VFLOW_sz	-	Volumetric flow rate of solids in Z direction
MASS_g	-	Total mass of gas in the volume specified
MASS_s	-	Total mass of solids in the volume specified

Flow Rates and Inventory

7 Tutorial 1: Bubbling Fluidized Bed

This tutorial demonstrates the modeling of a bubbling fluidized bed. The fluidized bed is a cylindrical vessel 14 cm in diameter and 100 cm tall (Figure 5). The vessel contains a 50 cm high minimally fluidized bed of sand (density = 2000 kg/m^3 and diameter = 4 x 10^{-4} m). Fluidizing air flows in through the bottom distributor at a velocity of 0.259 m/s. Additional air flows into the bed through a central jet of diameter 2 cm at a jet velocity of 1 m/s. The velocity of the central jet drops to 0.259 m/s after 0.2 s of operation.

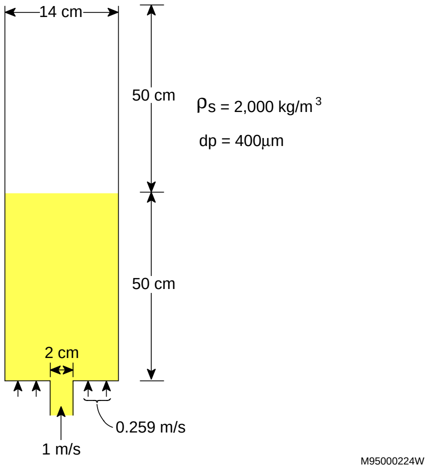


Figure 5. Geometry of the Bubbling Fluidized Bed

7.1 Writing the Data File

The problem described above is coded into an MFIX data file called *mfix.dat*, which will be located in the directory in which the calculation is initiated. It is desirable to include a few (optional) comment statements to identify the problem, the author, creation date, etc., in the data file as shown below.

```
#
#   Bubbling Fluidized Bed Simulation
#
#   F.B. Modeler                    9-6-94
#
```

The first section of the data file lists run-control information. The run name BUB01 specified here consists of the mnemonic BUB for bubbling beds and the number 01. In subsequent parametric studies, usually only this number is changed; e.g., BUB02, BUB03, etc. The description specified will appear in the output files and in the graphical display. This run will use CGS units. This is a new run starting from time zero and ending at 1 s. The time step size is 10^{-4} s. Since this is an isothermal fluidized bed, the energy equations are not solved. Also the gas and solids species equations are not solved.

There are no inputs needed in the physical and numerical parameters section. We choose to solve the problem in axisymmetric cylindrical coordinates with 7 cells in the radial

```
#
# Run-control section
#
  RUN_NAME = 'BUB01'
  DESCRIPTION = 'Bubbling Fluidized Bed Simulation'
  RUN_TYPE = 'new'
  UNITS = 'cgs'
  TIME  = 0.0    TSTOP = 1.0    DT = 1.0E-4
  ENERGY_EQ = .FALSE.
```

direction and 100 cells in the axial direction. The domain length in the radial direction is 7 cm and 100 cm in the axial direction. Since a uniform mesh size is used, there is no need to specify DX and DY values. Since this is an axisymmetric run, calculations in the azimuthal (K) direction are turned off.

```
#
# Geometry Section
#
  COORDINATES = 'cylindrical'

  XLENGTH  =   7.0    IMAX =   7
  YLENGTH  = 100.0    JMAX = 100
```

The gas-phase section specifies the viscosity and the average molecular weight of the gas. By specifying the gas viscosity, it will be kept constant during the run. The equation of state specified in the *eosg.f* file will be used to calculate the gas density.

```
#
# Gas-phase Section
#
  MU_g0  = 1.8E-4
```

The solids-phase section specifies the solids density, particle diameter, the coefficient of restitution, the angle of internal friction, and the void fraction at maximum packing. By setting the angle of internal friction to zero we are turning off the frictional stress calculations in MFIX.

```
#
# Solids-phase Section
#
  RO_s    = 2.0
  D_p     = 0.04

  e       = 0.8
  Phi     = 0.0
```

For setting the initial conditions, the reactor is divided into two regions: the bed and the freeboard. These regions are identified by the coordinates of their edges. Each column corresponds to an initial-condition region. The entries are read from left to right as, for example, IC_X_w(1) and IC_X_w(2).

```
#
# Initial Conditions Section
#
      !              Bed       Freeboard
  IC_X_w          =  0.0             0.0
  IC_X_e          =  7.0             7.0
  IC_Y_s          =  0.0            50.0
  IC_Y_n          = 50.0           100.0
```

For each of the regions specified above, various initial conditions are specified. In the bed region, we specify a bed void fraction and a gas velocity in the axial direction. The gas velocity is calculated by dividing the bottom distributor velocity by the void fraction. In the freeboard region, the void fraction is set to 1.0 and the gas velocity is the same as that at the bottom distributor. We have left the pressure unspecified. This forces MFIX to initialize a hydrostatic pressure distribution, assuming minimum fluidization. A reasonable initialization of the pressure is desirable for ensuring initial convergence. Since it is a minimally fluidized bed, the solids pressure and velocity are initialized to zero. In the specification of the solids velocity IC_U_s, for example, two indices are specified: the first index identifies the initial-condition number and the second index identifies the solids phase; in this example there is only one solids phase. Again, the different initial conditions are read from left to right as, for example, IC_U_s(1,1) and IC_U_s(2,1).

```
  IC_EP_g         =  0.42            1.0

  IC_U_g          =  0.0             0.0
  IC_V_g          =@(25.9/0.42)     25.9

  IC_U_s(1,1)     =  0.0             0.0
  IC_V_s(1,1)     =  0.0             0.0
```

```
#
#  Boundary Conditions Section
#
      !              Jet   Grid     Exit
  BC_X_w          =  0.0    1.0      0.0
  BC_X_e          =  1.0    7.0      7.0
  BC_Y_s          =  0.0    0.0    100.0
```

Three flow boundary conditions are specified. No wall boundary conditions are specified, since by default MFIX will set the wall to be a no-slip-wall and the center-line to be a symmetry plane or free-slip-wall. The boundary planes are defined by specifying the coordinates of the edges. Since the flow planes are normal to the y-axis in this example, the y-south and the y-north are equal.

```
BC_TYPE           = 'MI'   'MI'    'PO'

BC_EP_g           =  1.0    1.0

BC_U_g            =  0.0    0.0
BC_V_g            = 25.9   25.9

BC_P_g(3)         = 1.013E6 1.013E6 1.013E6
```

For each of the flow planes, first a boundary condition type is specified. The first two, at the bottom of the bed, are specified mass-inflow (MI) boundaries, and the last one, at the top of the bed, is a specified pressure outflow (PO) boundary. The void fraction and the velocities are specified for each of the mass-inflow boundaries. Only the pressure needs to be specified for the pressure outflow boundary. At the mass-inflow boundaries the specified pressure is used only for computing the gas-mass flow into the reactor.

```
BC_Jet_g0        = 25.9
BC_DT_0          =  0.0
BC_Jet_gh        =100.0
BC_DT_h          =  0.2
BC_Jet_gl        = 25.9
```

Additional conditions must be specified for the central jet since it is intermittent. The jet starts out at a velocity of 100 cm/s. The velocity decreases to 25.9 cm/s after 0.2 s and remains at that velocity for 0.9 s. An awkward part of the data file is when the run is restarted, one needs to change BC_Jet_g0 and BC_DT_0 to get the correct condition for the intermittent jet. For example, if this run were stopped at 0.1 s and then restarted the specifications would be BC_Jet_g0 = 100 and BC_DT_0 = 0.1. If the run were stopped at 0.4 s and restarted, then BC_Jet_g0 = 25.9 and BC_DT_0 = 0.7.

The last section of the data file is the output control section. In this section we have specified that the restart file will be updated (overwritten) every 0.01 s, the void fraction data will be written every 0.01 s, and the gas and solids velocities and pressures will be written every 0.1 s. The other three SPX data files need not be written. Hence a large time interval is specified for writing those files. The NLOG specification tells how often the *.LOG* file is updated with information on iterations and other run-time diagnostics. We have not specified an interval for writing the *.OUT* file so as not to write field-variable data into the *.OUT* file.

7.2 Running MFIX

Copy the file *param.inc* into the run directory and change the dimensions to DIMENSION_I = 9 and DIMENSION_J = 102, which are determined by adding 2 to IMAX and JMAX to account for the fictitious cells. The product of the above two dimensions give

```
#
#  Output Control
#
  RES_DT = 0.01
        !
        ! EP_g      P_g       U_g  U_s  ROP_s     T_g  X_g
        !      P_star    V_g  V_s            T_s1      X_s
        !                W_g  W_s            T_s2
  SPX_DT = 0.01 0.1           0.1  0.1  100.           100.
          100.
```

DIMENSION_3 = 918. All other user-defined dimensions are set at a value of 1. To make an MFIX executable file, type

make_mfix

```
        BC/IC condition flags at K =    1

          J   I=      1   2   3   4   5   6   7   8   9
        102        W-- P03 P03 P03 P03 P03 P03 P03 W--
        101        S-- .02 .02 .02 .02 .02 .02 .02 W--
        100        S-- .02 .02 .02 .02 .02 .02 .02 W--
         99        S-- .02 .02 .02 .02 .02 .02 .02 W--
                                  ...

          3        S-- .01 .01 .01 .01 .01 .01 .01 W--
          2        S-- .01 .01 .01 .01 .01 .01 .01 W--
```

Answer **y** to the question "Do you need any MFIX files from the run-directory? (y/n)," so that MFIX is recompiled with this new *param.inc* file. The executable file *mfix.exe* will be copied into the run-directory automatically.

As a first step, we will set TSTOP=0 and make a run to create the *.OUT* file. As a minimum, the cell flags, as shown below, printed at the end of the first part of *.OUT* file should be verified. The interior cells at the bottom are marked .01, which stands for initial condition number 01. The interior cells at the top are marked .02, for initial condition number 02. In restart runs, since no initial conditions from *mfix.dat* are used, the cell flag .-- will be used for the interior cells.

MFIX has set the axis (left boundary) as a free-slip-wall (S) because cylindrical coordinates are being used, and has set the right boundary as a no-slip-wall (W). The bottom boundary consist of one I01, which stands for specified mass-inflow condition 1, and six I02, which stands for specified mass-inflow condition number 2. The top boundary is uniformly marked P03, which stands for constant pressure outflow condition number 3.

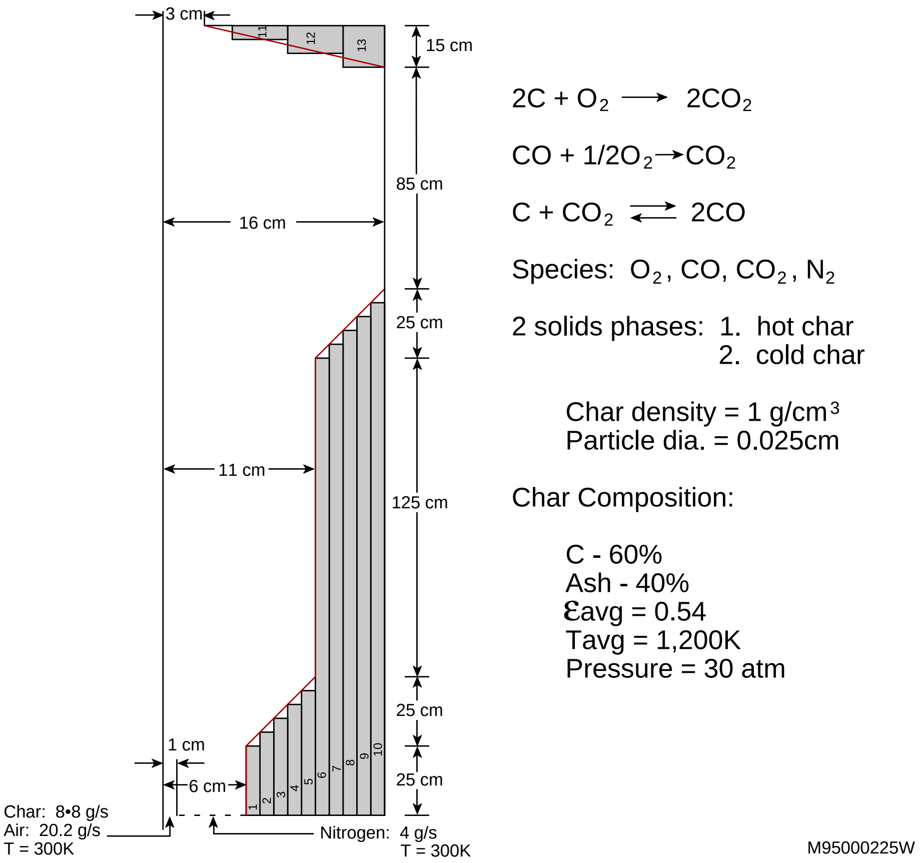
As an MFIX run progresses, the user should monitor the *.LOG* file to determine whether a converged solution is being obtained. Excerpts from the *.LOG* file for the present run are shown below. The file shows that the run converges quite readily. The run was completed in about 7.4 h on an SGI Crimson computer.

```
MFIX Simulation (V01.90)

                           ...

TIME(SEC)  OUT_IT   SMASS      % CONV              CPU TIME
0.00000     56     8928.4     100.00                   5.
 Data in .RES file updated at time =   9.999999999999950E-03
 Data written to .SP1 file at time =   9.999999999999950E-03

                           ...

TIME(SEC)  OUT_IT   SMASS      % CONV              CPU TIME
    0.50000     31     8928.4      100.00            11658.
 Data in .RES file updated at time =   0.5099999999999602
 Data written to .SP1 file at time =   0.5099999999999602

                           ...

TIME(SEC)  OUT_IT   SMASS      % CONV              CPU TIME
    0.99000     24     8928.4      100.00            26403.
```

Excerpts from the *.LOG* File

8 Tutorial 2: Partial Combustor

In this tutorial we will model a partial combustor with a geometry as shown in Figure 6. Char (8.8 g/s) and air (20.2 g/s) are fed into the combustor through a central tube. A second stream of fluidizing nitrogen (4 g/s) enters the annular region surrounding the tube. Partial combustion of the char occurs in the combustor. The product gases and unconverted solids exit from the top of the combustor.



$2C + O_2 \longrightarrow 2CO_2$

$CO + 1/2O_2 \rightarrow CO_2$

$C + CO_2 \rightleftarrows 2CO$

Species: O_2, CO, CO_2, N_2

2 solids phases: 1. hot char
2. cold char

Char density = 1 g/cm^3
Particle dia. = 0.025cm

Char Composition:

C - 60%
Ash - 40%
εavg = 0.54
Tavg = 1,200K
Pressure = 30 atm

Figure 6. Geometry of the Partial Combustor

An MFIX simulation will be used to determine the stability of the flame, the location and the magnitude of maximum temperature, and the gas and the solids mixing patterns in the combustor.

The char density is 1 g/cm^3 and the particle diameter is 250 µm. The average void fraction in the reactor is estimated to be 0.54. The gas residence time is approximately 36 s and the solids residence time is over 2 hours. It is not practical to do an MFIX simulation of the filling of the bed and to establish steady-state temperature conditions in the bed. Therefore, the estimated values of void fraction and bed temperature are specified as initial conditions.

Since the incoming char is at a low temperature and it is critical to consider char heat-up, the incoming char will be treated as a second solids phase called "cold-char." The char already in the combustor will be called "hot-char." When the temperature of the cold-char exceeds a certain specified value, say 800 K, it is assumed to convert into hot-char. A fast pseudo-reaction is specified to convert the cold-char at temperatures above that value to hot-char.

Four gas species are considered: 1. O_2, 2. CO, 3. CO_2, and 4. N_2. Two pseudo-species are considered in the solids phases: 1. Fixed carbon and 2. Ash. The numbering scheme shown above will be used for identifying the species in MFIX.

The chemical reactions modeled are the following:

Carbon combustion: $2C + O_2$ þ $2CO$

CO combustion: $CO + ½O_2$ þ CO_2

Gasification: $C + CO_2$ þ $2CO$

Pseudo-reaction: Cold-char þ Hot-char.

8.1 Specifying Reaction Kinetics

The reaction kinetics are specified in the file *rrates.f*, a copy of which appears in Appendix C. The file starts out with two PARAMETER statements. The first one specifies the value of the gas constant in certain desired units. The second one specifies a maximum temperature to limit the temperatures used in the kinetics calculations. It is desirable to bound the temperatures because the temperatures may momentarily become large during iterations and, if not bounded, the Arrhenius terms may become too large and cause no-convergence. By bounding the temperatures, no error is introduced, if the converged values of the temperatures are less than the value used for bounding the temperature. Every NLOG number of time-steps, MFIX checks whether the converged temperatures are within the bounding values.

The *rrates.f* subroutine is divided into five sections. User input is required in the first four sections. In the first section, the reaction kinetics for the four reactions are coded. The forward and backward reactions are written separately. For specifying the rate of the pseudo reaction, we have used the user-defined constants C(1) and C(2), so that they may be entered from the data file. In the second section, the reaction rates computed in the first section are used to compute the formation rates for individual species. In the third section, the reaction rates are used to compute the mass transfer between the phases. In the fourth section, the heats of reaction are used to compute the heat generation or consumption in each of the phases. In the fifth section, some bookkeeping and error checking computations are performed, which require no user input.

8.2 Writing User-Defined Subroutines

In addition to *rrates.f*, we also need to write the user-defined routines *calc_physprop.f*, *usr0.f*, *usr_init_namelist.f*, and *usr_namelist.inc*, which are given in Appendix C. In *calc_physprop.f*, the gas viscosity, gas specific heat, solids specific heat, gas conductivity, solids conductivity, wall heat loss, and Sherwood number are computed.

In *usr0.f*, the user-defined input are checked for error and constants are calculated based on the user-defined input. In *usr_init_namelist.f*, the user-defined namelist variables are initialized as UNDEFINED, which is a number defined in MFIX to indicate undefined double precision numbers. By initializing values this way, the user is able to check whether the values are specified through the input file. *usr_namelist.inc* file contains the definitions of user-defined variables, user-defined common blocks, and a namelist section.

The *param.inc* file is updated with all the necessary array dimensions. The initial part of the file is shown in Appendix C.

8.3 Writing the Data File

The input data file *mfix.dat* is given in Appendix C. In the run-control section, the energy balance calculations and species balance calculations are turned on. Since the subroutine *usr0.f* is to be used, the CALL_USR switch is turned on.

The two user-defined constants C(1) and C(2), used in the subroutine RRATES, are specified. They have been given (optional) names, with which to label them in the *.OUT* file.

In the radial direction, a nonuniform grid is used to get a better resolution near the jet. Up to a radial distance of 2 cm a mesh size of 0.5 cm is used. From 6 cm to 16 cm a mesh size of 1 cm is used. The transition is made smooth by expanding the mesh size from 0.5 cm to 1.0 cm while going from 2 cm to 6 cm. Equation (3) gives a uniform expansion ratio of 1.14 and Equation (4) gives the number of grids as 5.19. Based on this information, the grid sizes over five cells were adjusted with a spreadsheet program to get the desired grid. The expansion ratios are less than 1.2 as desired. The axial cell size was chosen as 5 cm.

The number of gas species is specified as 4 and their molecular weights are given. The number of solids phase species is specified as two. The molecular weight of ash is specified as 56. (This value, however, is not used in the *rates.f* routine we have written).

For simplicity, a uniform initial condition is specified for the entire computational domain. Although this is not appropriate for the many wall-cells in the range, such an initial condition specification will not cause any problem. Also, for the central jet region, the specified void fraction is too low. The solids will clear out of that region, however, as the MFIX calculations progress.

There are two flow inlets and one outlet. Note that mass flows are specified instead of the axial velocity component. Also the composition of gases and solids at the inlet are specified. At the outlet only the pressure is specified.

The reactor geometry is specified with 13 obstacles. We have started the obstacles' specification as boundary condition number 20, so that any future additions or deletions of inlets or outlets can be accommodated without having to change the boundary condition indices for the obstacles.

An internal surface has been specified to simulate the central tube at the bottom of the reactor.

The *.RES* file is written every 0.01 s and all the *.SPx* files are written every 0.1 s. Diagnostics from the code are printed out every 100th time step.

8.4 Running MFIX

The FORTRAN files discussed in sections 8.1 and 8.2 and the input data file discussed in section 8.3 are stored in a run directory. To create an MFIX executable, type

make_mfix

The script file will then ask whether any user-defined files need to be copied. Type **y** to use user-defined files and specify the path name to the run directory. All the user-defined files discussed in sections 8.1 and 8.2 will be copied into the MFIX directory and an MFIX executable will be created and copied into the run directory.

For the file *param.inc,* the script asks whether that file needs to be copied. Type **y** so that the file is copied. For subsequent compilations, when *param.inc* is not modified, do not recopy the file since it will force the recompilation of almost all MFIX files.

As always, set TSTOP=0 and run MFIX. Correct any input errors until MFIX successfully reads the input data file and exits. MFIX will write all the output files. Read the *.OUT* file to verify that the inputs were read correctly. Verify that the cell flags are assigned correctly. The cell flags for this run are shown below:

```
 J   I=     1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20  21
62        W-- P03 P03 P03 P03 P03 P03 W-- W-- W-- W-- W-- W-- W-- W-- W-- W-- W-- W-- W-- W--
61        S-- .01 .01 .01 .01 .01 .01 .01 .01 W30 W30 W30 W30 W31 W31 W31 W31 W32 W32 W32 W--
60        S-- .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 W31 W31 W31 W31 W32 W32 W32 W--
59        S-- .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 W32 W32 W32 W--
58        S-- .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 W--
57        S-- .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 W--
56        S-- .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 W--
55        S-- .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 W--
54        S-- .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 W--
53        S-- .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 W--
52        S-- .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 W--
51        S-- .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 W--
50        S-- .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 W--
49        S-- .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 W--
48        S-- .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 W--
47        S-- .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 W--
46        S-- .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 W--
45        S-- .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 W--
44        S-- .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 W--
43        S-- .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 W--
42        S-- .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 W--
41        S-- .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 W--
40        S-- .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 W29 W--
39        S-- .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 W28 W29 W--
38        S-- .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 W27 W28 W29 W--
37        S-- .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 W26 W27 W28 W29 W--
36        S-- .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 W25 W26 W27 W28 W29 W--
35        S-- .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 W25 W26 W27 W28 W29 W--
34        S-- .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 W25 W26 W27 W28 W29 W--
33        S-- .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 W25 W26 W27 W28 W29 W--
32        S-- .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 W25 W26 W27 W28 W29 W--
31        S-- .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 W25 W26 W27 W28 W29 W--
30        S-- .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 W25 W26 W27 W28 W29 W--
29        S-- .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 W25 W26 W27 W28 W29 W--
28        S-- .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 W25 W26 W27 W28 W29 W--
27        S-- .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 W25 W26 W27 W28 W29 W--
26        S-- .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 W25 W26 W27 W28 W29 W--
25        S-- .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 W25 W26 W27 W28 W29 W--
24        S-- .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 W25 W26 W27 W28 W29 W--
23        S-- .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 W25 W26 W27 W28 W29 W--
22        S-- .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 W25 W26 W27 W28 W29 W--
21        S-- .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 W25 W26 W27 W28 W29 W--
20        S-- .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 W25 W26 W27 W28 W29 W--
19        S-- .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 W25 W26 W27 W28 W29 W--
18        S-- .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 W25 W26 W27 W28 W29 W--
17        S-- .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 W25 W26 W27 W28 W29 W--
16        S-- .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 W25 W26 W27 W28 W29 W--
15        S-- .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 W25 W26 W27 W28 W29 W--
14        S-- .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 W25 W26 W27 W28 W29 W--
13        S-- .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 W25 W26 W27 W28 W29 W--
12        S-- .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 W25 W26 W27 W28 W29 W--
11        S-- .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 W25 W26 W27 W28 W29 W--
```

```
10        S-- .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 W24 W25 W26 W27 W28 W29 W--
 9        S-- .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 W23 W24 W25 W26 W27 W28 W29 W--
 8        S-- .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 W22 W23 W24 W25 W26 W27 W28 W29 W--
 7        S-- .01 .01 .01 .01 .01 .01 .01 .01 .01 .01 W21 W22 W23 W24 W25 W26 W27 W28 W29 W--
 6        S-- .01 .01E.01 .01 .01 .01 .01 .01 .01 W20 W21 W22 W23 W24 W25 W26 W27 W28 W29 W--
 5        S-- .01 .01E.01 .01 .01 .01 .01 .01 .01 W20 W21 W22 W23 W24 W25 W26 W27 W28 W29 W--
 4        S-- .01 .01E.01 .01 .01 .01 .01 .01 .01 W20 W21 W22 W23 W24 W25 W26 W27 W28 W29 W--
 3        S-- .01 .01E.01 .01 .01 .01 .01 .01 .01 W20 W21 W22 W23 W24 W25 W26 W27 W28 W29 W--
 2        S-- .01 .01E.01 .01 .01 .01 .01 .01 .01 W20 W21 W22 W23 W24 W25 W26 W27 W28 W29 W--
 1        W-- I01 I01 I02 I02 I02 I02 I02 I02 I02 W-- W-- W-- W-- W-- W-- W-- W-- W-- W-- W--
```

Note how the wall cells have been assigned to approximate the geometry of the reactor. Also note how the impermeable internal surface has been specified at the east face of the cells i=3 and j=2-6.

9 References

Gidaspow, D., and B. Ettehadieh, 1983, "Fluidization in Two-Dimensional Beds with a Jet; 2. Hydrodynamic Modeling," *I&EC Fundamentals*, **22**, 193-201.

Harlow, F.H., and A.A. Amsden, 1975, "Numerical Calculation of Multiphase Fluid Flow," *J. of Comp. Physics*, **17**, 19-52.

O'Brien, T.J., and M. Syamlal, 1990, "NIMPF: A Nonisothermal Multiparticle Fluidized-Bed Hydrodynamic Model," in Numerical Method for Multiphase Flows, Eds. I. Celik, D. Hughes, C.T. Crowe, and D. Lankford, **FED-Vol. 91**, ASME, New York, 65-72.

Rivard, W.C., and M.D. Torrey, 1977, "K-FIX: A Computer Program for Transient, Two-Dimensional, Two-Fluid Flow," LA-NUREG-6623, Los Alamos National Laboratory, Los Alamos.

Syamlal, M., 1985, Multiphase Hydrodynamics of Gas-Solids Flow, Ph.D. Dissertation, Illinois Institute of Technology.

Syamlal, M., W. Rogers, and T.J. O'Brien, 1993, "MFIX Documentation: Theory Guide," Technical Note, DOE/METC-94/1004, NTIS/DE94000087, National Technical Information Service, Springfield, VA.

Appendix A. Summary of Equations

A.1 Equations

The equations that are solved in version 1.91 of MFIX are summarized in this section.

Gas continuity:

$$\frac{}{t}(\ {}_{g}\ {}_{g}) \qquad (\ {}_{g}\ {}_{g} v_{g}) \qquad {}^{N_g}_{n\ 1} R_{gn} \tag{6}$$

Solids continuity:

$$\frac{}{t}(\ {}_{sm}\ {}_{sm}) \qquad (\ {}_{sm}\ {}_{sm} v_{sm}) \qquad {}^{N_{sm}}_{n\ 1} R_{smn} \tag{7}$$

Gas momentum balance:

$$\begin{aligned} v_{g}) \qquad (\ {}_{g}\ {}_{g} v_{g} v_{g}) \qquad {}_{g}\ P_{g} \qquad \overset{=}{}{}_{g} \qquad {}^{M}_{m\ 1} F_{gm}(v_{sm} \\ {}_{g}\ {}_{g} g \qquad {}^{M}_{m\ 1} R_{0m} \quad {}_{0m} v_{sm} \quad \overline{}\ {}_{0m} v \end{aligned} \tag{8}$$

Solids momentum balance:

$$\begin{aligned} {}_{m} v_{sm}) \qquad (\ {}_{sm}\ {}_{sm} v_{sm} v_{sm}) \qquad {}_{sm}\ P_{g} \qquad \overline{\overline{S}}_{sm} \\ F_{gm}(v_{sm} \quad v_{g}) \qquad {}^{M}_{l\ 1} F_{slm} \\ {}_{sm}\ {}_{sm} g \qquad {}^{M}_{l\ 0} R_{ml} \quad {}_{ml} v_{sl} \end{aligned} \tag{9}$$

Gas energy balance:

$$\frac{T_g}{t} \quad v_g \quad T_g\Big) \qquad q_g \quad {}_{g1}(T_{s1}\ \ T_g) \quad {}_{g2}(T_{s2}\ \ T$$
$$H_{wall}(T_{wall}\ \ T_g) \tag{10}$$

Solids - 1 energy balance:

$${}_{s1}C_{ps1}\left(\frac{T_{s1}}{t} \quad v_{s1} \quad T_{s1}\right) \qquad q_{s1} \quad {}_{g1}(T_{s1}\ \ T_g) \tag{11}$$

Solids - 2 energy balance:

$${}_{sm}C_{psm}\left(\frac{T_{s2}}{t} \quad v_{sm} \quad T_{s2}\right) \qquad q_{s2} \quad {}_{g2}(T_{s2}\ \ T_g) \tag{12}$$

Gas species balance:

$$\frac{}{t}(\ {}_g\ {}_g X_{gn}) \qquad (\ {}_g\ {}_g X_{gn} v_g) \quad R_{gn} \tag{13}$$

Solids species balance:

$$\frac{}{t}(\ {}_{sm}\ {}_{sm} X_{smn}) \qquad (\ {}_{sm}\ {}_{sm} X_{smn} v_{sm}) \quad R_{smn} \tag{14}$$

Gas-solids drag:

$$F_{gm} \quad \frac{3\ \ {}_{sm}\ {}_g\ {}_g}{4V_{rm}^2 d_{pm}}\ 0.63\ \ 4.8\sqrt{V_{rm}/Re_m}\ ^2 \quad v_{sm}\ \ v_g \tag{15}$$

$$0.5\ A\ \ 0.06Re_m\ \ \sqrt{(0.06Re_m)^2\ \ 0.12Re_m(2B\ A)\ \ } \tag{16}$$

$$A \qquad {}_{g}^{4.14} \tag{17}$$

$$B \quad \begin{cases} 0.8\ {}_{g}^{1.28} & if \quad {}_{g}\ 0.85 \\ \ {}_{g}^{2.65} & if \quad {}_{g}>0.85 \end{cases} \tag{18}$$

$$Re_m \quad \frac{d_{pm}\ v_{sm}\ \ v_g\ \ {}_g}{\mu_g} \tag{19}$$

Solids-solids drag:

$$\frac{3(1\ \ e_{lm})(\frac{}{2}\ \ \frac{C_{flm}}{8}{}^{2})\ {}_{sl}\ {}_{sl}\ {}_{sm}\ {}_{sm}(d_{pl}\ \ d_{pm})^2 g_{0_{lm}}\ \ v_{s}}{2\ (\ {}_{sl}d_{pl}^3\ \ \ {}_{sm}d_{pm}^3)} \tag{20}$$

$$g_{0_{lm}} \quad \frac{1}{\ {}_g} \quad \frac{3\left({}^{M}_{1}\ \ {}_{s}/d_p\right) d_{pl}d_{pm}}{{}_{g}^{2}(d_{pl}\ \ d_{pm})} \tag{21}$$

Gas-phase stress:

$$\overline{\overline{\ }}_g \quad 2\ {}_g\mu_{gt}\overline{\overline{D}}_g\ \ \frac{2}{3}\ {}_g\mu_{gt}tr(\overline{\overline{D}}_g)\overline{\overline{I}} \tag{22}$$

$$\mu_{gt} \quad Min(\mu_{gmax},\ \mu_g\ \ \ \mu_e) \tag{23}$$

$$\mu_e \quad 2\ l_s^2\ \ {}_g\ \ {}_g\sqrt{I_{2Dg}} \tag{24}$$

$$I_{2Dg} \quad \frac{1}{6}\,(D_{g11} \quad D_{g22})^{2} \quad (D_{g22} \quad D_{g33})^{2} \quad (D_{g33} \quad D_{g11})^{2}$$
$$D_{g12}^{2} \quad D_{g23}^{2} \quad D_{g31}^{2} \qquad (25)$$

Porous media model:

$$f_{g} \quad \frac{\mu_{g}}{c_{1}} v_{g} \quad \frac{c_{2}}{2} \; {}_{g} \; v_{g} \; v_{g} \qquad (26)$$

Granular stress:

$$\overline{\overline{S}}_{sm} \quad \begin{cases} P^{p}_{sm}\overline{\overline{I}} \quad \overline{\overline{\ }}^{p}_{sm} \quad if \quad {}_{g} \quad {}_{g} \\ P^{v}_{sm}\overline{\overline{I}} \quad \overline{\overline{\ }}^{v}_{sm} \quad if \quad {}_{g} > \; {}_{g} \end{cases} \qquad (27)$$

Plastic Regime:

$$P^{p}_{sm} \quad {}_{sm} P \qquad (28)$$

$$P \quad 10^{25}(\;{}_{g} \quad {}_{g})^{10} \qquad (29)$$

$$\overline{\overline{\ }}^{p}_{s1} \quad 2\,\mu^{p}_{s1}\overline{\overline{D}} \qquad (30)$$

$$\mu^{p}_{s1} \quad \frac{P \;\; \sin}{2\sqrt{I_{2Ds}}} \qquad (31)$$

$$I_{2Ds} \quad \frac{1}{6}\,(D_{s11} \quad D_{s22})^{2} \quad (D_{s22} \quad D_{s33})^{2} \quad (D_{s33} \quad D_{s11})^{2}$$
$$D_{s12}^{2} \quad D_{s23}^{2} \quad D_{s31}^{2} \qquad (32)$$

Viscous Regime:

$$P_{sm}^{v} \quad K_{1m} \; {}_{sm}^{2} \; {}_{m} \tag{33}$$

$$\overline{\overline{}}_{sm}^{v} \quad {}_{sm}^{v} tr(\overline{\overline{D}}_{sm})\overline{\overline{I}} \quad 2\mu_{sm}^{v}\overline{\overline{D}}_{sm} \tag{34}$$

$${}_{sm}^{v} \quad K_{2m} \; {}_{sm}\sqrt{\quad {}_{m}} \tag{35}$$

$$\mu_{sm}^{v} \quad K_{3m} \; {}_{sm}\sqrt{\quad {}_{m}} \tag{36}$$

$$K_{1m} \quad 2(1 \; e_{mm}) \; {}_{sm} g_{0_{mm}} \tag{37}$$

$$K_{2m} \quad 4 d_{pm} \; {}_{sm}(1 \; e_{mm}) \; {}_{sm} g_{0_{mm}} / (3\sqrt{\;}) \quad \frac{2}{3} K_{3m} \tag{38}$$

$$\begin{aligned} {}_{3m} \quad & \frac{d_{pm} \; {}_{sm}}{2} \left\{ \frac{\sqrt{\;}}{3(3 \; e_{mm})} [1 \; 0.4(1 \; e_{mm})(3e_{mm} \; 1) \; {}_{sm} g_{0_{m}} \right. \\ & \left. \frac{8 \; {}_{sm} g_{0_{mm}}(1 \; e_{mm})}{5\sqrt{\;}} \right\} \end{aligned} \tag{39}$$

$$g_{0_{mm}} \quad \frac{1}{{}_{g}} \quad \frac{3\left({}_{1}^{M} \frac{{}_{s}}{d_{p}} \right) d_{pm}}{2 \; {}_{g}^{2}} \tag{40}$$

Gas-solids heat transfer:

$${}_{gm} \quad \frac{C_{pg} R_{0m}}{\exp\left(\frac{C_{pg} R_{0m}}{{}_{gm}^{0}} \right) \; 1} \tag{41}$$

$$ {}^{0}_{gm} \qquad \frac{6\ k_{g}\quad {}_{sm}\ Nu_{m}}{d_{pm}{}^{2}} \qquad , \tag{42} $$

$$ \begin{aligned} Nu_{m} &\quad (7\ \ 10\ {}_{g}\ \ 5\ {}^{2}_{g})(1\ \ 0.7 Re_{m}^{0.2} Pr^{1/3}) \\ &\quad (1.33\ \ 2.4\ {}_{g}\ \ 1.2\ {}^{2}_{g}) Re_{m}^{0.7} Pr^{1/3} \quad . \end{aligned} \tag{43} $$

Granular energy equation:

$$ \frac{{}_{m} tr(\bar{\bar{D}}_{sm})\ \ \sqrt{K_{1m}^{2} tr^{2}(\bar{\bar{D}}_{sm})\ {}^{2}_{sm}\ \ 4K_{4m}\ {}_{sm}[K_{2m} tr^{2}(\bar{\bar{D}}_{sm})\ \ 2K_{3}}}{2\ {}_{sm}K_{4m}} $$

$$ K_{4m} \qquad \frac{12(1\ \ e_{mm}^{2})\ {}_{sm}\, g_{0mm}}{d_{pm}\sqrt{\ }} \tag{45} $$

A.2 Nomenclature

c_1	-	Permeability of porous media; m^2
c_2	-	Inertial resistance factor of porous media; m^{-1}
C_{Ds}	-	Single particle drag function
C_{pg}	-	Specific heat of the fluid phase; J/kg þ K
C_{flm}	-	Coefficient of friction for solids phases l and m
C_{psm}	-	Specific heat of the mth solids phase; J/kg þ K
d_{pm}	-	Diameter of the particles constituting the mth solids phase; m
$\overline{\overline{D}}_g$	-	Rate of strain tensor, fluid phase; s^{-1}
$\overline{\overline{D}}_{sm}$	-	Rate of strain tensor, solids phase-m; s^{-1}
e_{lm}	-	Coefficient of restitution for the collisions of mth and lth solids phases
f_g	-	Fluid flow resistance due to porous media; N/m^3
F_{gm}	-	Coefficient for the interphase force between the fluid phase and the mth solids phase; kg/m^3 þ s
F_{slm}	-	Coefficient for the interphase force between the lth solids phase and the mth solids phase; kg/m^3 þ s
g	-	Acceleration due to gravity; m/s^2
$g_{0_{lm}}$	-	Radial distribution function at contact
þH_{rg}	-	Heat of reaction in the fluid phase; J/m^3 þ s
þH_{rsm}	-	Heat of reaction in the mth solids phase; J/m^3 þ s
H_{wall}	-	Wall heat transfer coefficient; J/m^3 þ K þ s
I_{2Dg}	-	Second invariant of the deviator of the strain rate tensor for gas phase; s^{-2}
I_{2Ds}	-	Second invariant of the deviator of the strain rate tensor for solids phase-1; s^{-2}

I_{gm} - Momentum transfer from fluid phase to mth solids phase; N/m^3

I_{ml} - Momentum transfer from mth to lth solids phases; N/m^3

k_g - Fluid-phase conductivity; J/m þ K þ s

k_{pm} - Conductivity of material that constitutes solids phase-m; J/m þ K þ s

k_{sm} - Solids phase-m conductivity; J/m þ K þ s

k_m - Granular energy conductivity; J þ s/m^3

l - Index of the lth solids phase; also used as a miscellaneous index

l_s - A turbulence length-scale parameter; m

m - Index of the mth solids phase. "m=0" indicates fluid phase

M - Total number of solids phases

Mw - Average molecular weight of gas

n - Index of the nth chemical species

N_g - Total number of fluid-phase chemical species

N_{sm} - Total number of solids phase-m chemical species

Nu_m - Nusselt number

P_g - Pressure in the fluid phase; Pa

P_{sm}^{p} - Pressure in Solids phase-m, plastic regime; Pa

P_{sm}^{v} - Pressure in Solids phase-m, viscous regime; Pa

P^* - Total solids pressure in plastic regime; Pa

Pr - Prandtl number

q_g - Fluid-phase conductive heat flux; J/m^2 þ s

q_{s1} - Solids-phase-1 conductive heat flux; J/m^2 þ s

q_{s2} - Solids-phase-2 to M conductive heat flux; J/m^2 þ s

R - Universal gas constant; Pa þ m^3/kmol þ K

Re_m - mth solids phase particle Reynolds number

R_{km} - Ratio of solids to fluid conductivity

R_{ml} - Rate of transfer of mass from mth phase to lth phase. l or m = 0 indicates fluid phase; kg/m^3 þ s

R_{gn} - Rate of production of the nth chemical species in the fluid phase; kg/m^3 þ s

R_{smn} - Rate of production of the nth chemical species in the mth solids phase; kg/m^3 þ s

$\overline{\overline{S}}_g$ - Fluid-phase stress tensor; Pa

$\overline{\overline{S}}_{sm}$ - Solids phase-m stress tensor; Pa

t - Time; s

T_g - Thermodynamic temperature of the fluid phase; K

T_{s1} - Thermodynamic temperature of the solids phase-1; K

T_{s2} - Average thermodynamic temperature of the solids phases, m = 2,...,M; K

T_{wall} - Wall temperature; K

v_g - Fluid-phase velocity vector; m/s

v_{sm} - mth solids-phase velocity vector; m/s

V_{rm} - The ratio of the terminal velocity of a group of particles to that of an isolated particle

X_{gn} - Mass fraction of the nth chemical species in the fluid phase

X_{smn} - Mass fraction of the nth chemical species in the mth solids phase

GREEK LETTERS

$þ_{gm}$ - Fluid-solids heat transfer coefficient corrected for interphase mass transfer; J/m^3 þ K þ s

$^{0}_{gm}$ - Fluid-solids heat transfer coefficient not corrected for interphase mass transfer; J/m^3 þ K þ s

$_{m}$ - Granular energy dissipation due to inelastic collisions; J/m^3 þ s

þ$_{g}$ - Volume fraction of the fluid phase (void fraction)

$_{g}$ - Packed-bed (minimum) void fraction

þ$_{sm}$ - Volume fraction of the mth solids phase

þ - Function of restitution coefficient

þ$_{m}$ - Granular temperature of phase-m; m^2/s^2

þ$_{rm}$ - Solids conductivity function

$^{v}_{sm}$ - Second coefficient of solids viscosity, viscous regime; kg/m þ s

μ_{e} - Eddy viscosity of the fluid phase; kg/m þ s

μ_{g} - Molecular viscosity of the fluid phase; kg/m þ s

μ_{gmax} - Maximum value of the turbulent viscosity of the fluid phase; kg/m þ s

μ_{gt} - Turbulent viscosity of the fluid phase; kg/m þ s

μ^{p}_{s1} - Solids viscosity, plastic regime; kg/m þ s

μ^{v}_{s1} - Solids viscosity, viscous regime; kg/m þ s

þ$_{ml}$ - þ$_{ml}$ = 1 if $R_{ml} < 0$; else þ$_{ml}$ = 0.

þ$_{g}$ - Microscopic (material) density of the fluid phase; kg/m^3

$_{g}$ - Macroscopic (effective) density of the fluid phase; kg/m^3

þ$_{sm}$ - Microscopic (material) density of the mth solids phase; kg/m^3

$_{sm}$ - Macroscopic (bulk) density of the mth solids phase; kg/m^3

$\underset{g}{=}$ - Fluid phase deviatoric stress tensor; Pa

$=^{p}_{sm}$ - Solids phase-m deviatoric stress tensor, plastic regime; Pa

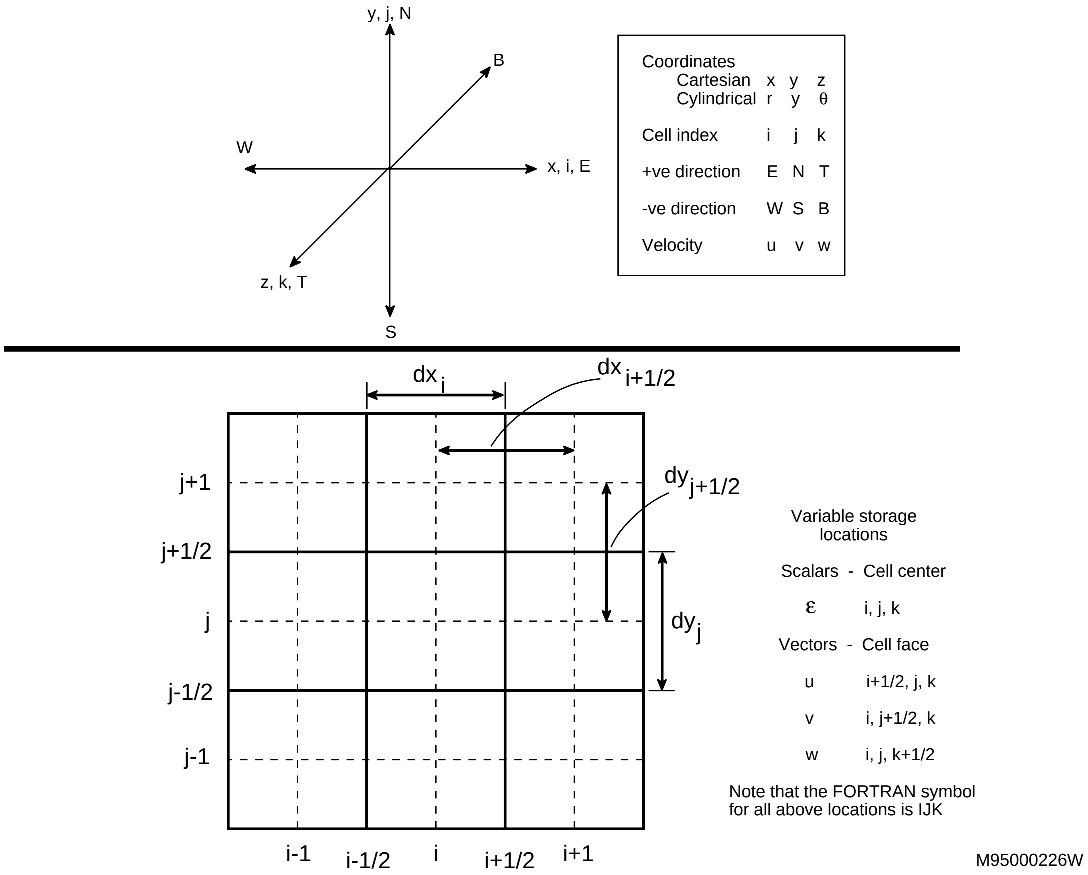
$=^{v}_{sm}$ - Solids phase-m deviatoric stress tensor, viscous regime; Pa

þ - Angle of internal friction

$þ_{k}$ - Contact area fraction in solids conductivity model

Appendix B. Details of MFIX Code

B.1 Coordinate System and Grid



Coordinates			
Cartesian	x	y	z
Cylindrical	r	y	θ
Cell index	i	j	k
+ve direction	E	N	T
-ve direction	W	S	B
Velocity	u	v	w

Variable storage locations

Scalars -	Cell center
ε	i, j, k
Vectors -	Cell face
u	i+1/2, j, k
v	i, j+1/2, k
w	i, j, k+1/2

Note that the FORTRAN symbol for all above locations is IJK

Figure 7. Numerical Grid and Naming Conventions

B.2 List of FORTRAN Symbols

C_pg	Fluid-phase specific heat -- C_{pg}
C_ps	Solids-phase specific heat -- C_{psm}
DTxF_gs	Time step x Fluid-solids drag coefficient -- þt x F_{gm}
DTxF_ss	Time step x Fluid-solids drag coefficient -- þt x F_{slm}
EP_g	Void fraction -- þ$_{g}$
GAMMA_g1	Fluid-solids heat transfer coefficient -- þ$_{g1}$
GAMMA_g2	Fluid-solids heat transfer coefficient -- þ$_{g2}$
HOR_g	Heat of reaction for fluid phase -- þH_{rg}
HOR_s	Heats of reaction for solids phases -- þH_{rs1} and þH_{rs2}
I	X-direction index
IJK	Composite index for i, j, k
IMJK	Composite index for i-1, j, k
IPJK	Composite index for i+1, j, k
IJMK	Composite index for i, j-1, k
IJPK	Composite index for i, j+1, k
IJKM	Composite index for i, j, k-1
IJKP	Composite index for i, j, k+1
IJKW	= IMJK, if IMJK is a fluid cell; else IJK
IJKE	= IPJK, if IPJK is a fluid cell; else IJK
IJKS	= IJMK, if IJMK is a fluid cell; else IJK
IJKN	= IJPK, if IJPK is a fluid cell; else IJK
IJKB	= IJKM, if IJKM is a fluid cell; else IJK

IJKT	= IJKP, if IJKP is a fluid cell; else IJK
J	Y-direction index
K	Z-direction index
K_g	Fluid-phase thermal conductivity -- k_g
K_s	Solids-phase thermal conductivity -- k_{sm}
LAMBDA_s	Solids-phase second coefficient of viscosity -- $þ_{sm}$
M	Index for solids phase
MU_g	Fluid viscosity -- μ_g
MU_s	Solids-phase viscosity -- μ_{sm}
N	Index of gas or solids species
P_g	Fluid pressure -- P_g
P_s	Solids pressure -- P_s
P_star	Solids pressure in plastic regime -- P^*
R_gp	Rate of production of gas species n (part of R_{gn})
RoX_gc	Rate of consumption of gas species n divided by X_{gn} (part of R_{gn}/X_{gn})
R_sp	Rate of production of solids species n (part of R_{smn})
RoX_sc	Rate of consumption of solids species n divided by X_{smn} (part of R_{smn}/X_{smn})
RO_g	Fluid density -- $þ_g$
ROP_g	Macroscopic (effective) density of the fluid: $_g \quad _g$
SUM_R_g	$_{n\ 1}^{N_g} R_{gn}$
SUM_R_s	$_{n\ 1}^{N_{sm}} R_{smn}$

T_g	Fluid temperature -- T_g
T_s1	Solids phase-1 temperature -- T_{s1}
T_s2	Average temperature of solids phases 2 to M -- T_{s2}
U_g	X-component of fluid velocity -- u_g
U_s	X-component of solids velocity -- u_{sm}
V_g	Y-component of fluid velocity -- v_g
V_s	Y-component of solids velocity -- v_{sm}
W_g	Z-component of fluid velocity -- w_g
W_s	Z-component of solids velocity -- w_{sm}
X_g	Fluid species mass fraction -- X_{gn}
X_s	Solids species mass fraction -- X_{sm}

B.3 List of Functions

BOTTOM_OF (IJK)	Calculate IJKB.
EAST_OF (IJK)	Calculate IJKE.
EP_s (IJK, M)	Calculate the volume fraction of the M^{th} solids phase.
FLUID_AT (IJK)	Logical function to determine whether the cell IJK is a fluid cell.
FUNIJK (I, J, K,)	Calculate the composite index IJK from I, J, and K values.
IM_OF (IJK)	Calculate IMJK.
IP_OF (IJK)	Calculate IPJK.
IS_AT_E (IJK)	Logical function to determine whether there is an internal surface at the east-face of the IJK cell.
IS_AT_N (IJK)	Logical function to determine whether there is an internal surface at the north-face of the IJK cell.
IS_AT_T (IJK)	Logical function to determine whether there is an internal surface at the top-face of the IJK cell.
JM_OF (IJK)	Calculate IJMK.
JP_OF (IJK)	Calculate IJPK.
KM_OF (IJK)	Calculate IJKM.
KP_OF (IJK)	Calculate IJKP.
NORTH_OF (IJK)	Calculate IJKN.
SOUTH_OF (IJK)	Calculate IJKS.
TOP_OF (IJK)	Calculate IJKT.
WALL_AT (IJK)	Logical function to determine whether the cell IJK is a wall cell.
WEST_OF (IJK)	Calculate IJKW.

In addition to the above, a number of averaging functions has been defined in the files *functions_avg1.inc* and *functions_avg2.inc*.

B.4 List of Include Files

bc.inc	Common blocks for boundary condition variables
body_force.inc	Statement function for body force
constants.inc	Common blocks for constants
field_variables.inc	Common blocks for field variables
file_units.inc	Assignment of file unit numbers
functions.inc	Miscellaneous statement functions
functions_avg1.inc	Type definitions for averaging functions
functions_avg2.inc	Statement functions for doing spatial averaging of variables
geometry.inc	Common blocks for geometry variables
ic.inc	Common blocks for initial condition variables
indices.inc	Common blocks for cell indices
is.inc	Common blocks for internal surface variables
machine.inc	Machine-dependent constants
namelist.inc	Namelist definitions
other_variables.inc	Common blocks for arrays used in MFIX
output_control.inc	Common blocks for output control variables
param.inc	Parameter definitions for array dimensions and constants
physical_prop.inc	Common blocks for physical property variables
run_control.inc	Common blocks for run control variables
solids_pr.inc	Statement function for the solids pressure function
time_cpu.inc	Common block for CPU time variables
tolerance.inc	Various convergence criteria and iteration limits
usr_namelist.inc	Common blocks for user-defined variables and definition of user-defined namelist variables

B.5 List of Subroutines

calc_dil_ue.f Use dilute flow algorithm to calculate velocity in the east momentum cell.

calc_dil_uw.f Use dilute flow algorithm to calculate velocity in the west momentum cell.

calc_dil_vn.f Use dilute flow algorithm to calculate velocity in the north momentum cell.

calc_dil_vs.f Use dilute flow algorithm to calculate velocity in the south momentum cell.

calc_dil_wb.f Use dilute flow algorithm to calculate velocity in the bottom momentum cell.

calc_dil_wt.f Use dilute flow algorithm to calculate velocity in the top momentum cell.

calc_drag.f Calculate fluid-solids and solids-solids drag.

calc_gamma.f Calculate fluid-solids heat transfer coefficient.

calc_mu_gt.f Calculate turbulent viscosity for fluid.

calc_physprop.f Calculate physical properties.

calc_q.f Calculate heat transfer by conduction.

calc_rop_s.f Solve solids continuity equations to calculate solids volume fractions.

calc_t.f Solve energy equations.

calc_theta.f Calculate granular temperature and granular viscosity coefficients and pressure.

calc_tild.f Calculate the explicitly differenced quantities in the momentum equations.

calc_velocity.f Solve momentum equations.

calc_x.f Solve species balance equations.

drag_gs.f Calculate fluid-solids drag coefficient.

drag_ss.f	Calculate solids-solids drag coefficient.
eosg.f	Calculate fluid density.
g_0.f	Calculate radial distribution function at contact.
get_data.f	Read and process input data.
iterate.f	Adjust fluid and solids pressures and control iterative solution.
mfix.f	Main program.
part_tau_s.f	Calculate solids viscous stress terms implicitly for plastic stress computations.
read_namelist.f	Read namelist variables.
rrates.f	Calculate reaction rates, species formation rates, and heats of reactions.
set_bc0.f	Set boundary conditions initially.
set_bc1.f	Set boundary conditions after every time step.
set_flags.f	Set the cell flags.
set_geometry.f	Set geometry variables.
set_ic.f	Set initial conditions.
set_index1.f	Set the indices for neighboring cells.
solved_t.f	Test the convergence of the energy balance equations.
solved_x.f	Test the convergence of the species balance equations.
time_march.f	Control marching in time.

Appendix C. Tutorial 2 Files

C.1 rrates.f

```
CvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvC
C                                                                      C
C  Module name: RRATES                                                 C
C  Purpose: Calculate gasification and combustion reaction rates       C
C                                                                      C
C  Author: M. Syamlal                                 Date: 26-SEP-94  C
C  Reviewer:                                          Date: dd-mmm-yy  C
C                                                                      C
C                                                                      C
C  Literature/Document References:                                     C
C                                                                      C
C     Desai, P.R. and C.Y. Wen, "Computer Modeling of the MERC Fixed   C
C        Bed Gasifier," MERC/CR-78/3, March, 1978.                     C
C                                                                      C
C     Syamlal, M., Rogers, W., and T.J. O'Brien, "MFIX Documentation   C
C        Theory Guide, DOE/METC-94/1004, NTIS/DE94000087, National     C
C        Technical Information Service, Springfield, VA, 1993.         C
C                                                                      C
C     Wen, C.Y., H. Chen, and M. Onozaki, "User's Manual for Computer  C
C       Simulation and Design of the Moving Bed Coal Gasifier,"        C
C       DOE/MC/16474-1390 (DE83009533), January, 1982                  C
C                                                                      C
C     Westbrook, C.K., and F.L. Dryer, "Simplified mechanisms for the  C
C        oxidation of hydrocarbon fuels in flames," Combustion Science C
C        and technology, Vol. 27, pp. 31-43, (1981).                   C
C                                                                      C
C  Variables referenced: MMAX, IJK, T_g, T_s1, D_p, X_g, X_s, EP_g,    C
C            P_g, HOR_g, HOR_s                                         C
C                                                                      C
C                                                                      C
C  Variables modified: M, N, R_gp, R_sp, RoX_gc, RoX_sc, SUM_R_g,      C
C                      SUM_R_s                                         C
C                                                                      C
C  Local variables:                                                    C
C^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^C
C
C
      SUBROUTINE RRATES
      IMPLICIT NONE
C
C     Gas constant for O2 in cm^3.atm/g.K
      DOUBLE PRECISION R_O2
      PARAMETER (R_O2 = 82.06/32.)
C
C     Limit of maximum temperature (K)
      DOUBLE PRECISION MAX_TEMP
      PARAMETER (MAX_TEMP = 2500.0)
C
      INCLUDE 'param.inc'
      INCLUDE 'field_variables.inc'
      INCLUDE 'other_variables.inc'
      INCLUDE 'geometry.inc'
      INCLUDE 'run_control.inc'
      INCLUDE 'indices.inc'
      INCLUDE 'physical_prop.inc'
      INCLUDE 'constants.inc'
      INCLUDE 'file_units.inc'
      INCLUDE 'usr_namelist.inc'
      INCLUDE 'devol_mgas.inc'
C
C  Function subroutines
C
      LOGICAL COMPARE
C
C  Local Variables
C
C                      Local phase index
      INTEGER          L, LM
C
C                      Temperatures, Pressures, Proximate Analysis,
C                      Rate constants, activation energies,
C                      reaction rates
      DOUBLE PRECISION TGS1X, TGS2X, PO2, PCO, DIFF,
     &                 PCO2, PATM, PATM_MW,
     &                 K_f, R_D1, R_D2, RXNA, K_a, RXNA1F, RXNA2F,
     &                 RXNA1B, RXNA2B, CAR1, CAR2, EP_s1, EP_s2,
     &                 TGX, TS1X, TS2X, EQ5, RXNC, RXNC1F, RXNC2F, RXNC1B,
     &                 RXNC2B, RXNF2F, RXNF2B, RXNJ1F, RXNJ2F,  K_r
C
      INCLUDE 'functions.inc'
C
C  User input is required in sections 1 through 4.
C
C1111111111111111111111111111111111111111111111111111111111111111111111111111111
C
C
C 1. Write the rates of various reactions:
C    Write the reaction rates for each of the reactions as RXNxF and RXNxB (both
C    quantities >= 0), where x identifies the reaction, F stands for forward
C    rate, and B stands for the backward rate.  The rates can be in
C    g-mole/(cm^3.s) or g/(cm^3.s).  For the sake of clarity, give the reaction
C    scheme and the units in a comment statement above the rate expression.
C    The volume (cm^3) is that of the computational cell.  Therefore, for
C    example, the rate term of a gas phase reaction will have a multiplicative
C    factor of epsilon. Note that X_g and X_s are mass fractions
C
C     Hot Char  - solids-1
C     Cold Char - solids-2
C
C     PAA  ash fraction -- Proximate analysis
C     PAFC Fixed carbon fraction -- Proximate analysis
C     EP_A  Ash layer void fraction
C
```

```
C              No   1   2    3   4
C     GAS Species  O2, CO, CO2, N2
C
      TGX   = MIN(MAX_TEMP, T_g(IJK))
      TS1X  = MIN(MAX_TEMP, T_s1(IJK))
      TS2X  = MIN(MAX_TEMP, T_s2(IJK))
      TGS1X = HALF * (TGX + TS1X)
      TGS2X = HALF * (TGX + TS2X)
C
C     COMPUTE PARTIAL PRESSURE of various gases IN ATM.  P_g in dynes/cm^2
C
      PATM    = P_g(IJK) / 1013000.
      PATM_MW = PATM * MW_MIX_g(IJK)
      PO2     = PATM_MW * X_g(IJK, 1) / MW_g(1)
      PCO     = PATM_MW * X_g(IJK, 2) / MW_g(2)
      PCO2    = PATM_MW * X_g(IJK, 3) / MW_g(3)
C
      EP_s1  = EP_s(IJK,1)
      EP_s2  = EP_s(IJK,2)
C
C     concentration of carbon, gmole/cc
C
      CAR1   = ROP_s(IJK,1) * X_s(IJK, 1, 1) / MW_s(1,1)
      CAR2   = ROP_s(IJK,2) * X_s(IJK, 2, 1) / MW_s(2,1)
C
C  a)   COMBUSTION: 2C + O2 --> 2CO; g-mole/(cm^3.s)
C       Wen at al. (1982), Syamlal et al. (1993), Desai and Wen (1978)
C
      IF ( PO2 .GT. ZERO .AND. .NOT.COMPARE(EP_g(IJK), ONE) ) THEN
        IF(PAFC .NE. ZERO) THEN
          IF(X_s(IJK,1,1) .GT. ZERO) THEN
            R_D1 = (X_s(IJK,1,1) * PAA / (X_s(IJK,1,2) * PAFC))**(1./3.)
            R_D1 = MIN(ONE, R_D1)
          ELSE
            R_D1 = ZERO
          ENDIF
          IF(X_s(IJK,2,1) .GT. ZERO) THEN
            R_D2 = (X_s(IJK,2,1) * PAA / (X_s(IJK,2,2) * PAFC))**(1./3.)
            R_D2 = MIN(ONE, R_D2)
          ELSE
            R_D2 = ZERO
          ENDIF
        ELSE
          R_D1 = ZERO
          R_D2 = ZERO
        ENDIF
C
        DIFF = 4.26 * ((TGX/1800.)**1.75) / PATM
C
        IF(R_D1 .EQ. ZERO .OR. EP_s1 .EQ. ZERO) THEN
          RXNA1F = ZERO
        ELSE
          K_f = DIFF * N_sh(IJK, 1) / (D_p(1) * R_O2 * TGX)
          K_r = 8710. * EXP( -27000./1.987/TS1X) * R_D1*R_D1
```

```
C
          IF(R_D1 .GE. ONE) THEN
            RXNA = ONE / (ONE / K_f + ONE / K_r)
          ELSE
            K_a = 2. * DIFF * f_EP_A * R_D1
     &          / ( D_p(1) * ( ONE - R_D1 ) * R_O2 * TS1X )
            RXNA = ONE / ( ONE /K_f + ONE/K_a + ONE/ K_r)
          ENDIF
          RXNA1F = RXNA * PO2 * 6.0 * EP_s1 / ( D_p(1) * 32.0 )
        ENDIF
C
        IF(R_D2 .EQ. ZERO .OR. EP_s2 .EQ. ZERO) THEN
          RXNA2F = ZERO
        ELSE
C
          K_f = DIFF * N_sh(IJK, 2) / (D_p(2) * R_O2 * TGX)
          K_r = 8710. * EXP( -27000./1.987/TS2X) * R_D2*R_D2
C
          IF(R_D2 .GE. ONE) THEN
            RXNA = ONE / (ONE / K_f + ONE / K_r)
          ELSE
            K_a = 2. * DIFF * f_EP_A * R_D2
     &          / ( D_p(2) * ( ONE - R_D2 ) * R_O2 * TS2X )
            RXNA = ONE / ( ONE /K_f + ONE/K_a + ONE/ K_r)
          ENDIF
          RXNA2F = RXNA * PO2 * 6.0 * EP_s2 / ( D_p(2) * 32.0 )
        ENDIF
      ELSE
        RXNA1F = ZERO
        RXNA2F = ZERO
      ENDIF
      RXNA1B = ZERO
      RXNA2B = ZERO
C
C b)  CHAR-CO2 REACTION: C + CO2 --> 2CO;   g-mole/(cm^3.s)
C     Wen et al. (1982)
      IF ( EP_s1 .GT. ZERO ) THEN
        EQ5 = EXP ( 20.9238 - 20281.8 / TGS1X )
        RXNC = AK5*EXP(-AE5/(1.987*TGS1X))*CAR1
        RXNC1F = RXNC * PCO2
        RXNC1B = RXNC * PCO*PCO/EQ5
      ELSE
        RXNC1F = ZERO
        RXNC1B = ZERO
      ENDIF
C
      IF ( EP_s2 .GT. ZERO ) THEN
        EQ5 = EXP ( 20.9238 - 20281.8 / TGS2X )
        RXNC = AK5*EXP(-AE5/(1.987*TGS2X))*CAR2
        RXNC2F = RXNC * PCO2
        RXNC2B = RXNC * PCO*PCO/EQ5
      ELSE
        RXNC2F = ZERO
        RXNC2B = ZERO
      ENDIF
C
C c)  CO COMBUSTION: CO + 1/2O2 --> CO2; g-mol/(cm^3.s)
C      Westbrook and Dryer (1981)
      IF(PCO .GT. ZERO .AND. PO2 .GT. ZERO) THEN
        RXNF2F = 3.98E14 * EXP(-40000.0/(1.987*TGX)) * EP_g(IJK)
     &          * (RO_g(IJK)*X_g(IJK,1)/MW_g(1)) ** 0.25
     &          * (RO_g(IJK)*X_g(IJK,2)/MW_g(2))
     &          * (RO_g(IJK)*X_g(IJK,6)/MW_g(6)) ** 0.5
      ELSE
        RXNF2F = ZERO
      ENDIF
      RXNF2B = ZERO
C
C  Psuedo-reactions that convert cold char to hot char;  g/(cm^3.s)
C
       IF(EP_s2 .GT. ZERO .AND. T_g2(IJK) .GT. C(1) )THEN
C
C        Carbon (cold char) --> Carbon (hot char)
         RXNJ1F = C(2) * ROP_s(IJK,2) * X_s(IJK,2,1)
C
C        Ash (cold char) --> Ash (hot char)
         RXNJ2F = C(2) * ROP_s(IJK,2) * X_s(IJK,2,2)
       ELSE
         RXNJ1F = ZERO
         RXNJ2F = ZERO
       ENDIF
C
C222222222222222222222222222222222222222222222222222222222222222222222222222222
C
C 2. Write the formation and consumption rates of various species:
C    Obtain the rates of formation and consumption of various species
C    in g/(cm^3.s) from the rate expressions RXNxF and RXNxB obtained in the
C    previous section.  Pay attention to the units of RXNxF and RXNxB.  All
C    the formation rates for gas species n are added to get R_gp (IJK, n).
C    All the consumption rates are added and then divided by X_g(IJK, n) to
C    get RoX_gc(IJK, n).  If X_g(IJK, n) is zero and species n is likely
C    to be consumed in a reaction then it is recommended that RoX_gc (IJK, n)
C    be initialized to the derivative of the consumption rate with respect to
C    X_g at X_g=0.
C    If the slope is not known analytically a small value such as 1.0e-9 may
C    instead be used.  A similar procedure is used for all the species in the
C    solids phases also.
C
C  GAS SPECIES
C  the gaseous species expressions have the forms:
C                                      R_gp(cell#, species#)
C                                      RoX_gc(cell#, species#)
C
C  (1) O2
      R_gp(IJK, 1) = ZERO
      IF(X_g(IJK, 1) .GT. ZERO) THEN
        RoX_gc(IJK, 1) = (RXNA1F + RXNA2F + HALF * RXNF2F ) * MW_g(1)
     &                    / X_g(IJK, 1)
```

```
      ELSE
        RoX_gc(IJK, 1) = 1.0e-9
      ENDIF
C
C  (2) CO
      R_gp(IJK, 2) = (2. * (RXNA1F + RXNA2F) + 2. * (RXNC1F + RXNC2F) )
     &                * MW_g(2)
      IF(X_g(IJK, 2) .GT. ZERO) THEN
        RoX_gc(IJK, 2) = ( 2. * (RXNC1B + RXNC2B) + RXNF2F) * MW_g(2)
     &                    / X_g(IJK, 2)
      ELSE
        RoX_gc(IJK, 2) = 1.0e-9
      ENDIF
C
C  (3) CO2
      R_gp(IJK, 3) = (RXNC1B + RXNC2B + RXNF2F ) * MW_g(3)
      IF(X_g(IJK, 3) .GT. ZERO) THEN
        RoX_gc(IJK, 3) = (RXNC1F + RXNC2F) * MW_g(3) / X_g(IJK, 3)
      ELSE
        RoX_gc(IJK, 3) = 1.0e-9
      ENDIF
C
C   (7) N2
C     inert here
      R_gp(IJK, 4) = ZERO
      RoX_gc(IJK, 4) = ZERO
C
C  SOLIDS SPECIES
C  COAL PSEUDO-SPECIES
C  the solid species' expressions have the form:
C                                   R_sp(cell#,solid phase#,solid species#)
C                                   RoX_sc(cell#,solid phase#,solid species#)
C
C (1)  CARBON
      R_sp(IJK, 1, 1) = (RXNC1B) * MW_s(1, 1) + RXNJ1F
      IF(X_s(IJK, 1, 1) .GT. ZERO) THEN
        RoX_sc(IJK, 1, 1) = (2. * RXNA1F + RXNC1F) * MW_s(1, 1)
     &                       / X_s(IJK, 1, 1)
      ELSE
        RoX_sc(IJK, 1, 1) = 1.0e-7
      ENDIF
C
      R_sp(IJK, 2, 1) = (RXNC2B ) * MW_s(2, 1)
      IF(X_s(IJK, 2, 1) .GT. ZERO) THEN
        RoX_sc(IJK, 2, 1) = ((2. * RXNA2F + RXNC2F) * MW_s(2, 1) + RXNJ1F)
     &                       / X_s(IJK, 2, 1)
      ELSE
        RoX_sc(IJK, 2, 1) = 1.0e-7
      ENDIF
C
C (4) ASH
      R_sp(IJK, 1, 2) = RXNJ2F
      RoX_sc(IJK,1,2) = ZERO
C
```

```
      R_sp(IJK, 2, 2) = ZERO
      IF(X_s(IJK,2,2) .GT. ZERO) THEN
        RoX_sc(IJK,2,2) = RXNJ2F/X_s(IJK,2,2)
      ELSE
        RoX_sc(IJK,2,2) = ZERO
      ENDIF
C
C
C3333333333333333333333333333333333333333333333333333333333333333333333333333333
C
C 3.  Determine the g/(cm^3.s) transferred from one phase to the other.
C     e.g. R_temp(0,1) -  mass generation of gas phase from solids-1,
C          R_temp(0,2) -  mass generation of gas phase from solids-2,
C          R_temp(1,0) -  mass generation of solid-1 from gas = -R_temp(0,1)
C          R_temp(1,2) -  mass generation of solid-1 from solids-2.
C     Note, for example, that if gas is generated from solids-1 then
C     R_temp(0,1) > 0.
C     The R-phase matrix is skew-symmetric and diagonal elements are not needed.
C     Only one of the two skew-symmetric elements needs to be specified.
C
    R_temp(0,1) = RXNA1F * (2. * MW_s(1,1) )+ (RXNC1F - RXNC1B) * MW_s(1,1)
C
    R_temp(0,2) = RXNA2F * (2. * MW_s(2,1) )+ (RXNC2F - RXNC2B) * MW_s(2,1)
C
    R_temp(1,2) = RXNJ1F + RXNJ2F
C
C
C4444444444444444444444444444444444444444444444444444444444444444444444444444444
C
C 4.  Determine the heat of reactions in cal/(cm^3.s) at the
C     temperature T_g or T_s1.  Note that for exothermic reactions
C     HOR_g or HOR_s should be negative. Heat of Reaction for the C + O2
C     reaction is split into parts; CO formation is assigned to the solid
C     phase and CO2 formation from CO to the gas phase.
C
      HOR_g(IJK)    = (-67636.0) * (RXNF2F - RXNF2B)
      HOR_s(IJK, 1) =   (-52832.0) * (RXNA1F - RXNA1B)
     &               +   (41220.0) * (RXNC1F - RXNC1B)
      HOR_s(IJK, 2) =   (-52832.0) * (RXNA2F - RXNA2B)
     &               +   (41220.0) * (RXNC2F - RXNC2B)
C
C==============================================================================
C
C         NO USER INPUT IS REQUIRED IN THIS SECTION
C
C 5. Determine g/(cm^3.s) of mass generation for each of the phases by adding
C    the reaction rates of all the individual species.
C
      IF(SPECIES_EQ(0)) THEN
        SUM_R_g(IJK) = ZERO
        DO 5000 N = 1, NMAX(0)
          SUM_R_g(IJK) = SUM_R_g(IJK) + R_gp(IJK, N)
     &                             - RoX_gc(IJK, N) * X_g(IJK, N)
5000    CONTINUE
```

```
      ENDIF
C
      DO 5200 M = 1, MMAX
        IF(SPECIES_EQ(M)) THEN
          SUM_R_s(IJK, M) = ZERO
          DO 5100 N = 1, NMAX(M)
            SUM_R_s(IJK, M) = SUM_R_s(IJK, M) + R_sp(IJK, M, N)
     &                           - RoX_sc(IJK, M, N) * X_s(IJK, M, N)
5100      CONTINUE
        ENDIF
5200  CONTINUE
C
C
C     Store R_temp values in an array.  Only store the upper triangle without
C     the diagonal of R_temp array.
C
      DO 5900 L = 0, MMAX
      DO 5900 M = L+1, MMAX
        LM = L + 1 + (M-1) * (M) / 2
        IF(R_temp(L,M) .NE. UNDEFINED) THEN
          R_phase(IJK, LM) = R_temp(L,M)
        ELSEIF(R_temp(M,L) .NE. UNDEFINED) THEN
          R_phase(IJK, LM) = -R_temp(M,L)
        ELSE
          CALL START_LOG
          WRITE(UNIT_LOG, 1000)L, M
          CALL END_LOG
          STOP
        ENDIF
5900  CONTINUE
C
C=======================================================================
C
      RETURN
1000  FORMAT(/1X,70('*')//' From: RRATES',
     &/' Message: Mass transfer between phases ', I2, ' and ', I2,
     &' (R_temp) not specified', /1X, 70('*')/)
      END
```

C.2 calc_physprop.f

```
CvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvC
C                                                                      C
C  Module name: CALC_PHYSPROP                                          C
C  Purpose: Calculate physical properties that vary with time          C
C                                                                      C
C  Author: M. Syamlal                                 Date: 17-JUL-92  C
C  Reviewer:                                          Date:            C
C                                                                      C
C  Revision Number: 1                                                  C
C  Purpose: To incorporate heat capacity changes resulting from        C
C           change in composition -- MGAS chemistry                    C
C  Author: S. Venkatesan                              Date: dd-mmm-yy  C
C  Reviewer:                                          Date: dd-mmm-yy  C
C                                                                      C
C  Literature/Document References:                                     C
C    Gunn, D.J., "Transfer of heat or mass to particles in fixed and   C
C      fluidised beds," Int. J Heat Mass Transfer, Vol. 21, pp 467-476,C
C      (1978).                                                         C
C    Perry, R.H., and C.H. Chilton, Chemical Engineer's Handbook, 5th  C
C      edition, McGraw-Hill Kogakusha, Tokyo, 1973.                    C
C    Bird, R.B., W.E. Stewart, E.N. Lightfoot, Transport Phenomena,    C
C      John Wiley & Sons, New York, 1960.                              C
C    Bauer, R., and E.U. Schlunder, "Effective radial thermal          C
C      conductivity of packings in gas flow: Part II: Thermal          C
C      conductivity of the packing fraction without gas flow," Int.    C
C      Chem. Eng., Vol. 18, 189-204 (1978).                            C
C                                                                      C
C  Variables referenced: ANY_SPECIES_EQ                                C
C  Variables modified: None                                            C
C                                                                      C
C  Local variables: TGX, TSX                                           C
C                                                                      C
C^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^C
C
      SUBROUTINE CALC_PHYSPROP
C
      IMPLICIT NONE
      INCLUDE 'param.inc'
      INCLUDE 'field_variables.inc'
      INCLUDE 'physical_prop.inc'
      INCLUDE 'geometry.inc'
      INCLUDE 'indices.inc'
      INCLUDE 'other_variables.inc'
      INCLUDE 'run_control.inc'
      INCLUDE 'usr_namelist.inc'
C
C  Statement function types
C                         constant-pressure heat capacities (Cp's)!
      DOUBLE PRECISION CPO2, CPCO, CPCO2, CPCH4, CPH2, CPH2O, CPH2S,
     &                 CPN2, CPNH3, CPTAR, CPC2H4, CPC2H6, CPC3H8,
     &                 CPC6H6, CPFC, CPVM, CPM, CPA
C
```

```
C  Local Variables
C
C
      DOUBLE PRECISION TGX, TS1X, TS2X, VMLEFT, DIFF, EP_g2
      DOUBLE PRECISION Sc1o3, UGC, VGC, WGC, USCM, VSCM, WSCM, VREL, Re
C
C  Statement functions
C
      INCLUDE 'functions_avg1.inc'
      INCLUDE 'functions.inc'
      INCLUDE 'functions_avg2.inc'
      CPO2(XXX)= (8.27 + 0.000258*XXX - 187700.0/XXX**2 )/32.
      CPCO(XXX)= ( 6.6 + 0.0012*XXX )/28.
      CPCO2(XXX)= ( 10.34 + 0.00274*XXX - 195500.0/XXX**2 )/44.
      CPN2(XXX)= ( 6.5 + 0.001*XXX )/28.
C
C     SPECIFIC HEAT OF COAL -- IGT Coal COnversion Systems Technical Data
C     Book (Sec. IA30.6, pp 3-4)
C
      CPFC(XXX)= -0.1315 + 1.341E-3*XXX - 1.087E-6*XXX*XXX
     &           + 3.06E-10*XXX*XXX*XXX
      CPA(XXX)= 0.1442 + 1.4E-4*XXX
C
C
C
      DO 500 K = 1, KMAX2
      DO 500 J = 1, JMAX2
      DO 500 I = 1, IMAX2
        IJK = FUNIJK (I, J, K)
        TGX  = T_g(IJK)
        TS1X  = T_s1(IJK)
        TS2X  = T_s2(IJK)
        IF( .NOT.WALL_AT(IJK) ) THEN
C
C     Gas viscosity (air) in g/cm.s
C     Perry & Chilton (1973) -- Formula 3-133 on page 3-248.
C
          IF(MU_g0 .EQ. UNDEFINED)
     &      MU_g (IJK) = 1.7D-4 * (T_g(IJK) / 273.0)**1.5
     &                   * (383. / (T_g(IJK) + 110.) )
C
C     Constant pressure spcific heat of air in cal/g.K
C     Perry & Chilton (1973) -- Table 3-174
C
          IF(ANY_SPECIES_EQ) THEN
            C_pg(IJK) =  X_g(IJK,1)*CPO2(TGX)  + X_g(IJK,2)*CPCO(TGX)
     &                 + X_g(IJK,3)*CPCO2(TGX) + X_g(IJK,7)*CPN2(TGX)
          ELSE
            C_pg(IJK) = 0.767 * CPN2(TGX) + 0.233 * CPO2(TGX)
          ENDIF
C
C     Specific heat of solids (Coal) in cal/g.K
C     Perry & Chilton(1973) -- Table 3-201 on page 3-136
C
```

```
          IF(ANY_SPECIES_EQ) THEN
            C_ps(IJK, 1) = X_s(IJK,1,1) * CPFC(TS1X)
     &                   + X_s(IJK,1,4) * CPA(TS1X)
            C_ps(IJK, 2) = X_s(IJK,2,1) * CPFC(TS2X)
     &                   + X_s(IJK,2,4) * CPA(TS2X)
          ELSE
            C_ps(IJK, 1) = 0.3
            C_ps(IJK, 2) = 0.3
          ENDIF
C
C  Gas conductivity (air) in cal/s.cm.K
C  Bird, Stewart, and Lightfoot (1960) -- Temperature dependence from formula
C  8.3-12 on p. 255 and conductivity value at 300 K from p. 263
C
          K_g(IJK) = 6.02D-5 * SQRT( T_g(IJK) / 300. )
C
C     Solids conductivity in cal/s.cm.K
C     From Bauer & Schlunder's (1978) theory an approximate average
C     value for the solids conductivity is 2.5*K_g
C
          K_s(IJK, 1) = 2.5 * K_g(IJK)
          K_s(IJK, 2) = 2.5 * K_g(IJK)
C
C     Wall heat transfer coefficient in cal/cm^3.s.K
C
          HWALL(IJK) = ZERO
          TWALL(IJK) = ZERO
C
C   Calculate Sherwood number for solids phases (Gunn 1978)
C
          EP_g2 = EP_g(IJK) * EP_g(IJK)
          DIFF = 4.26 * ((T_g(IJK)/1800.)**1.75) * 1013000. / P_g(IJK)
          Sc1o3 = (MU_g(IJK)/(RO_g(IJK) * DIFF))**(1./3.)
          UGC = AVG_U_g_W(IJK, I)
          VGC = AVG_V_g_S(IJK)
          WGC = AVG_W_g_B(IJK)
          DO 400 M = 1, MMAX
C
            USCM = AVG_U_s_W(IJK, I, M)
            VSCM = AVG_V_s_S(IJK, M)
            WSCM = AVG_W_s_B(IJK, M)
C
            VREL = SQRT((UGC - USCM)**2 + (VGC-VSCM)**2
     &                                  + (WGC-WSCM)**2 )
            Re = EP_g(IJK) * D_p(M) * VREL * RO_g(IJK) / MU_g(IJK)
            N_sh(IJK, M) = ( (7. - 10. * EP_g(IJK) + 5. * EP_g2)
     &                       *(ONE + 0.7 * Re**0.2 * Sc1o3)
     &                      + (1.33 - 2.4*EP_g(IJK) + 1.2*EP_g2)
     &                       * Re**0.7 * Sc1o3 )
400       CONTINUE
        ENDIF
500   CONTINUE
      RETURN
      END
```

C.3 usr_init_namelist.f

```
C
      SUBROUTINE USR_INIT_NAMELIST
C
      IMPLICIT NONE
      INCLUDE 'param.inc'
      INCLUDE 'usr_namelist.inc'
C
C
      PAFC = UNDEFINED
      PAA  = UNDEFINED
C
      RETURN
      END
```

C.4 usr0.f

```
C
      SUBROUTINE USR0
C
      IMPLICIT NONE
      INCLUDE 'param.inc'
C
C  Include files defining common blocks here
C
      INCLUDE 'physical_prop.inc'
      INCLUDE 'usr_namelist.inc'
      INCLUDE 'file_units.inc'
C
C  Function subroutines
C
      LOGICAL COMPARE
C
C  Define local variables here
C
      DOUBLE PRECISION SUM
C
C  Include files defining statement functions here
C
C
C  Insert user-defined code here
C
        IF(PAFC .EQ. UNDEFINED )
     &    CALL ERROR_ROUTINE ('USR0', 'PAFC not specified', 1, 1)
C
        IF(PAA .NE. UNDEFINED)THEN
          SUM = PAFC + PAA
          IF( .NOT.COMPARE(ONE,SUM) )THEN
            WRITE(UNIT_LOG,'(A,F10.5/A)')
     &        ' *** PAFC + PAA = ',SUM,' It should be equal to 1.0'
            CALL EXIT
          ENDIF
        ELSE
          PAA = 1.0 - PAFC
        ENDIF
C
C  Function of the ash-layer void fraction
      f_EP_A = (0.25 + 0.75 * ( 1.0 - PAA )) ** 2.5
C
      RETURN
      END
```

C.5 usr_namelist.inc

```
C
C  1. Define the user-defined namelist variables as DOUBLE PRECISION,
C     INTEGER, LOGICAL, or CHARACTER.
C  2. Include the variable in the appropriate common.
C  3. If variable, needs to be read from mfix.dat add the variable name
C     in the NAMELIST section.
C
C                      Sherwood number
      DOUBLE PRECISION N_Sh(DIMENSION_3, DIMENSION_M)
C
C                      Proximate Analysis
      DOUBLE PRECISION PAFC, PAA
C
C                      Function of ash-layer void fraction
      DOUBLE PRECISION f_EP_A
C
C     Double Precision variables
      COMMON / USR_DATA_DP /
     &       N_Sh, PAFC, PAA, f_EP_A
C
C
C
C      INTEGER
C
C     Integer variables
C      COMMON / USR_DATA_I /
C     &
C
C
C
C      LOGICAL
C     Logical variables
C      COMMON / USR_DATA_L /
C     &
C
C
C     Character variables
C      COMMON / USR_DATA_C /
C
C
      NAMELIST / USR_INPUT_DATA /
     &       PAFC, PAA
```

C.6 param.inc (initial part)

```
C
C 1.  Parameter input section:  The user may need to change this section
C       depending upon the size of the problem.
C
C                      Maximum of the number of cells in the x direction
      INTEGER          DIMENSION_I
      PARAMETER (DIMENSION_I = 21)
C
C                      Maximum of the number of cells in the y direction
      INTEGER          DIMENSION_J
      PARAMETER (DIMENSION_J = 62)
C
C                      Maximum of the number of cells in the z direction
      INTEGER          DIMENSION_K
      PARAMETER (DIMENSION_K = 1)
C
C                      Maximum number of computational cells
      INTEGER          DIMENSION_3
      PARAMETER (DIMENSION_3 = 1302)
C
C                      Maximum number of solids phases
      INTEGER          DIMENSION_M
      PARAMETER (DIMENSION_M = 2)
C
C                      Maximum number of gas species
      INTEGER          DIMENSION_N_g
      PARAMETER (DIMENSION_N_g = 4)
C
C                      Maximum number of solids species. For multiple solids
C                      phase use the maximum for all phases
      INTEGER          DIMENSION_N_s
      PARAMETER (DIMENSION_N_s = 2)
C
C===============================================================================
```

C.7 mfix.dat

```
#
#  Partial combustor
#
# Author:  E.F. Modeler            Date: 9-28-94
#
! 1. Run control
     Run_name       = 'pc01'
     Description    = 'Partial Combustor'
     Units          = 'cgs'
     Run_type       = 'new'

     Time           =  0.0
     Tstop          =  10.0
     DT             = 1.0E-4

     ENERGY_EQ      = .T.
     SPECIES_EQ     = .T.  .T.  .T.
     CALL_USR       = .T

! 2. User-defined constants
     C              =  800.                       1.0E2
     C_NAME         ='Transition Temp.' 'Pseudo-reaction const.'
     PAFC           = 0.6

! 3. Geometry
     Coordinates    = 'Cylindrical'

     IMAX           = 19
     DX             = 4*0.5, .59, .69, .81, .91, 11*1.0

     JMAX           = 60
     YLENGTH        = 300.0

     NO_K           = .T.

! 4. Gas-phase
     NMAX(0)        = 4
        !  Gas species   O2   CO   CO2   N2
     MW_g           = 32.  28.  44.   28.

! 5. Solids-phase
     MMAX           = 2
     D_P            = .025    .025
     RO_s           = 1.0     1.0

     NMAX(1)        = 2  2
     MW_s(1,1)      = 12.     12.
     MW_s(1,2)      = 56.     56.
     EP_star        = 0.4
     e              = 0.8
     C_f            = 0.1
     Phi            = 0.0

! 6. Initial conditions
     IC_X_w         =  0.0
     IC_X_e         = 16.0
     IC_Y_s         =  0.0
     IC_Y_n         =300.0

     IC_EP_g        =  0.54
     IC_P_Star      =  0.0
     IC_ROP_s(1,1)  =  0.46
     IC_ROP_s(1,2)  =  0.0

     IC_T_g         = 1200.0
     IC_T_s1        = 1200.0
     IC_T_s2        = 1200.0

     IC_U_g         =  0.0
     IC_V_g         = 30.0

     IC_U_s(1,1)    =  0.0
     IC_V_s(1,1)    =  1.0

     IC_X_g(1,1)    =  0.0
     IC_X_g(1,2)    =  0.0
     IC_X_g(1,3)    =  0.3
     IC_X_g(1,4)    =  0.7

     IC_X_s(1,1,1)  =  0.6
     IC_X_s(1,1,2)  =  0.4


! 7. Boundary conditions
                 !Jet         annulus        outlet
     BC_X_w         =  0.0         1.0          0.0
     BC_X_e         =  1.0         6.0          3.0
     BC_Y_s         =  0.0         0.0       300.0
     BC_Y_n         =  0.0         0.0       300.0

     BC_TYPE        = 'MI'         'MI'      'PO'

     BC_EP_g        = 0.9          1.0
     BC_P_g         = 3.04E7  3.04E7         3.04E7
     BC_ROP_s(1,1)  = 0.0
     BC_ROP_s(1,2)  = 0.1

     BC_T_g         = 300.0        300.0
     BC_T_s1        = 300.0
     BC_T_s2        = 300.0

     BC_U_g         =  0.0        0.0
     BC_MASSFLOW_g  = 20.2        4.0

     BC_U_s(1,2)    =  0.0        0.0
     BC_MASSFLOW_s(1,2)= 8.8       0.0

     BC_X_g(1,1)    =  0.2        0.0
     BC_X_g(1,2)    =  0.0        0.0
     BC_X_g(1,3)    =  0.0        0.0
     BC_X_g(1,4)    =  0.8        1.0

     BC_X_s(1,2,1)  =  0.6
     BC_X_s(1,2,2)  =  0.4

                         !  Define Obstacles
                !  1        2        3        4         5
     BC_X_w(20)     = 6.0      7.0      8.0      9.0      10.0
     BC_X_e(20)     = 7.0      8.0      9.0     10.0      11.0
     BC_Y_s(20)     = 0.0      0.0      0.0      0.0       0.0
     BC_Y_n(20)     =25.0     30.0     35.0     40.0      45.0
     BC_TYPE(20)    ='NSW'    'NSW'    'NSW'    'NSW'    'NSW'

                ! 6        7        8        9         10
     BC_X_w(25)     = 11.0    12.0     13.0     14.0      15.0
     BC_X_e(25)     = 12.0    13.0     14.0     15.0      16.0
     BC_Y_s(25)     =  0.0     0.0      0.0      0.0       0.0
     BC_Y_n(25)     =175.0   180.0    185.0    190.0     195.0
     BC_TYPE(25)    ='NSW'    'NSW'    'NSW'    'NSW'    'NSW'

               ! 11       12       13
     BC_X_w(30)     =  5.0      9.0    13.0
     BC_X_e(30)     =  9.0     13.0    16.0
     BC_Y_s(30)     =295.0    290.0   285.0
     BC_Y_n(30)     =300.0    300.0   300.0
     BC_TYPE(30)    ='NSW'    'NSW'    'NSW'

!8. Internal surfaces
     IS_X_w         =  1.0
     IS_X_e         =  1.0
     IS_Y_s         =  0.0
     IS_Y_n         = 25.0

     IS_TYPE        = 'IP'

!9. Output control
     RES_DT         =  0.01
     SPX_DT         = 7*0.1
     NLOG           =   100
```

Appendix D. Keywords in Input Data File

þ indicates the default value.

The symbols used in the table are as follows:

(dimension)	
l	Cell number in x, y, or z direction
m	Solids-phase number
n	Species number
ic	Initial condition number
bc	Boundary condition number
is	Internal surface number
usr	User-defined output number
Type	
C	Character
DP	Double Precision
I	Integer
L	Logical

D.1 Run Control

Keyword (dimension)	**Type**	**Description**
RUN_NAME	C	Name used for creating output files. The name should be legal after extensions are added to it., e.g., for run name BUB01, the output files *BUB01.LOG*, *BUB01.OUT*, *BUB01.RES*, etc., will be created.
DESCRIPTION	C	Problem description in 60 characters.
UNITS	C	Units for data input and output.
CGS		All input and output in CGS units.
RUN_TYPE	C	Type of run.
NEW		New run
RESTART_1		Normal restart run. Initial conditions from .*RES* file.
RESTART_2		Start a new run with initial conditions from a .*RES* file created from another run.
RESTART_3		Continue old run as in RESTART_1, but any input data not given in *mfix.dat* is read from the .*RES* file.
RESTART_4		Start a new run as in RESTART_2, but any input data not given in *mfix.dat* is read from the .*RES* file.
TIME	DP	Start-time of the run.

Keyword (dimension)	**Type**	**Description**
TSTOP	DP	Stop-time of the run.
DT	DP	Time step.
ENERGY_EQ	L	
þ *.TRUE.*		Solve energy equations.
.FALSE.		Do not solve energy equations.
SPECIES_EQ(m)	L	(m=0 indicates gas phase)
þ *.TRUE.*		Solve species equations of phase m.
.FALSE.		Do not species equations of phase m.
CALL_USR	L	
.TRUE.		Call user-defined subroutines.
þ *.FALSE.*		Do not call user-defined subroutines.

D.2 Physical and Numerical Parameters

C(100)	DP	User defined constants.
C_NAME(100)	C	Name of user-defined constants (20 characters long). These character strings are used only for identifying user- defined constants (C) in the .OUT file.
e	DP	Coefficient of restitution.
C_f	DP	Coefficient of friction between the particles of two solids phases.
Phi	DP	Angle of internal friction (in degrees). Set this value to zero to turn off plastic regime stress calculations.
L_scale0	DP	Value of turbulent length initialized. This may be overwritten in specific regions with the keyword IC_L_scale. þ Default value is 0.0.
Mu_gmax	DP	Maximum value of the turbulent viscosity of the fluid.

D.3 Geometry and Discretization

For 2D simulations the thickness of the third direction specified should be exact, if mass or volumetric flow rates, rather than the velocities, are specified at the boundaries.

Keyword (dimension)	**Type**	**Description**
COORDINATES	C	Coordinates used in the simulation.
CARTESIAN		Cartesian coordinates
CYLINDRICAL		Cylindrical coordinates
NO_I	L	
þ .FALSE.		x (r) direction is considered.
.*TRUE.*		x (r) direction is not considered.
IMAX	I	Number of cells in the x (r) direction.
DX (l)	DP	Cell sizes in the x (r) direction.
XMIN	DP	The inner radius in the simulation of an annular cylindrical region.
XLENGTH	DP	Reactor length in the x (r) direction.
NO_J	L	
þ .FALSE.		y direction is considered.
.*TRUE.*		y direction is not considered.
JMAX	I	Number of cells in the y direction.
DY (l)	DP	Cell sizes in the y direction.
YLENGTH	DP	Reactor length in the y direction.
NO_K	L	
þ .FALSE.		z (þ) direction is considered.
.*TRUE.*		z (þ) direction is not considered.
KMAX	I	Number of cells in the z (þ) direction.
DZ (l)	DP	Cell sizes in the z (þ) direction.
ZLENGTH	DP	Reactor length in the z (þ) direction.

Keyword (dimension)	**Type**	**Description**
CYCLIC_X	L	
þ .FALSE.		No cyclic condition at X-boundary.
.*TRUE.*		Cyclic condition at X-boundary.
CYCLIC_X_PD	L	
þ .FALSE.		No cyclic condition at X-boundary.
.*TRUE.*		Cyclic condition with pressure drop at X-boundary.
DELP_X	DP	Pressure drop across XLENGTH when a cyclic boundary condition with pressure drop is imposed in the X-direction.
CYCLIC_Y	L	
þ .FALSE.		No cyclic condition at Y-boundary.
.*TRUE.*		Cyclic condition at Y-boundary.
CYCLIC_Y_PD	L	
þ .FALSE.		No cyclic condition at Y-boundary.
.*TRUE.*		Cyclic condition with pressure drop at Y-boundary.
DELP_Y	DP	Pressure drop across YLENGTH when cyclic boundary condition with pressure drop is imposed in the Y-direction.
CYCLIC_Z	L	
þ .FALSE.		No cyclic condition at Z-boundary.
.*TRUE.*		Cyclic condition at Z-boundary.
CYCLIC_Z_PD	L	
þ .FALSE.		No cyclic condition at Z-boundary.
.*TRUE.*		Cyclic condition with pressure drop at Z-boundary.
DELP_Z	DP	Pressure drop across ZLENGTH when cyclic boundary condition with pressure drop is imposed in the Z-direction.

D.4 Gas Phase

RO_g0	DP	Specified constant gas density.
MU_g0	DP	Specified constant gas viscosity.
MW_AVG	DP	Average molecular weight of gas.
MW_g (n)	DP	Molecular weight of gas species n.

D.5 Solids Phase

MMAX	I	Number of solids phases.
D_p (m)	DP	Particle diameters.
RO_s (m)	DP	Particle densities.
NMAX (m)	I	Number of species in phase m. Note that m=0 indicates gas phase.
MW_s (m, n)	DP	Molecular weight of solids phase-m, species n.
EP_star	DP	Packed bed void fraction.

D.6 Initial Conditions

Each initial condition (IC) is specified over a rectangular (or pie-shaped, for cylindrical coordinates) region that corresponds to the scalar numerical grid. These are 3D regions, and X_w þ X_e, Y_s þ Y_n, and Z_t þ Z_b. The region is defined by the constant coordinates of each of the six faces, which may be specified as the physical coordinates or the cell indices. The physical coordinates are easier to specify than the cell indices. If cell sizes are not small enough to resolve a region specified using physical coordinates, MFIX will indicate this problem with an error message.

In cylindrical coordinates, when the theta direction crosses the 0 value, split that region into two regions: e.g., Split a region spanning 1.9þ to 0.1þ as 1.9þ to 2þ and 0 to 0.1þ.

Two initial condition regions may overlap. When an overlap occurs, MFIX uses the conditions specified for the higher IC number.

Keyword (dimension)	**Type**	**Description**
IC_X_w (ic)	DP	x coordinate of the west face.
IC_X_e (ic)	DP	x coordinate of the east face.
IC_Y_s (ic)	DP	y coordinate of the south face.
IC_Y_n (ic)	DP	y coordinate of the north face.
IC_Z_b (ic)	DP	z coordinate of the bottom face.
IC_Z_t (ic)	DP	z coordinate of the top face.
IC_I_w (ic)	I	i index of the west-most cell.
IC_I_e (ic)	I	i index of the east-most cell.
IC_J_s (ic)	I	j index of the south-most cell.
IC_J_n (ic)	I	j index of the north-most cell.
IC_K_b (ic)	I	k index of the bottom-most cell.
IC_K_t (ic)	I	k index of the top-most cell.

Keyword (dimension)	Type	Description
IC_TYPE (ic)	C	Type of initial condition. Mainly used in restart runs to overwrite values read from the .*RES* file by specifying it as 'PATCH'. The user needs to be careful when using the 'PATCH' option, since the values from the .*RES* file are overwritten and no error checking is done for the patched values.
IC_EP_g (ic)	DP	Initial void fraction in the IC region.
IC_P_g (ic)	DP	Initial gas pressure in the IC region. If this quantity is not specified, MFIX will set up a hydrostatic pressure profile, which varies only in the y-direction.
IC_P_star (ic)	DP	Initial solids pressure in the IC region. Usually this value is specified as zero.
IC_L_scale (ic)	DP	Turbulence length scale in the IC region.
IC_ROP_s (ic, m)	DP	Initial macroscopic density of solids phase-m in the IC region.
IC_T_g (ic)	DP	Initial gas phase temperature in the IC region.
IC_T_s1 (ic)	DP	Initial solids phase-1 temperature in the IC region.
IC_T_s2 (ic)	DP	Initial average temperature of solids phases 2 to M in the IC region.
IC_U_g (ic)	DP	Initial x-component of gas velocity in the IC region.
IC_U_s (ic, m)	DP	Initial x-component of solids-phase velocity in the IC region.
IC_V_g (ic)	DP	Initial y-component of gas velocity in the IC region.
IC_V_s (ic, m)	DP	Initial y-component of solids-phase velocity in the IC region.
IC_W_g (ic)	DP	Initial z-component of gas velocity in the IC region.
IC_W_s (ic, m)	DP	Initial z-component of solids-phase velocity in the IC region.
IC_X_g (ic, n)	DP	Initial mass fraction of gas species n.
IC_X_s (ic, m, n)	DP	Initial mass fraction of solids phase-m, species n.

D.7 Boundary Conditions

Boundary conditions (BC) are specified over flow planes or 2D surfaces that are normal to one of the coordinate directions and coinciding with a face of the scalar control-volume. The values for one of the three pairs of coordinates are equal. The surface is defined by the constant coordinates of each of the four edges, which can be specified with physical coordinates or cell indices, and the two equal values for the direction normal to the face, which can only be specified with physical coordinates. If cell sizes are not small enough to resolve a surface specified using physical coordinates, MFIX will indicate this problem with an error message.

A flow plane must have a wall cell (or an outside boundary) on one side and a flow cell on the other side.

The BC section is also used for specifying obstacles in the flow domain. Obstacles are 3D regions and just as for the IC regions X_w þ X_e, Y_s þ Y_n, and Z_t þ Z_b. By default the outside boundary is initialized as no-slip walls. For cylindrical coordinates the axis is initialized as a free-slip wall.

Two boundary surfaces must not intersect. Two obstacle regions may intersect.

Keyword (dimension)	**Type**	**Description**
BC_X_w (bc)	DP	x coordinate of the west face or edge.
BC_X_e (bc)	DP	x coordinate of the east face or edge.
BC_Y_s (bc)	DP	y coordinate of the south face or edge.
BC_Y_n (bc)	DP	y coordinate of the north face or edge.
BC_Z_b (bc)	DP	z coordinate of the bottom face or edge.
BC_Z_t (bc)	DP	z coordinate of the top face or edge.
BC_I_w (bc)	I	i index of the west-most cell.
BC_I_e (bc)	I	i index of the east-most cell.
BC_J_s (bc)	I	j index of the south-most cell.
BC_J_n (bc)	I	j index of the north-most cell.
BC_K_b (bc)	I	k index of the bottom-most cell.

Keyword (dimension)	**Type**	**Description**
BC_K_t (bc)	I	k index of the top-most cell.
BC_TYPE(bc)	C	Type of boundary:
DUMMY		The specified boundary condition is ignored. This is useful for turning off some boundary conditions without having to delete them from the file.
MASS_INFLOW or *MI*		Mass inflow rates for gas and solids phases are specified at the boundary.
MASS_OUTFLOW or *MO*		The specified values of gas and solids mass outflow rates at the boundary are maintained, approximately. This condition should be used (sparingly) for minor outflows, when the bulk of the outflow is occurring through other constant pressure outflow boundaries.
P_INFLOW or *PI*		Inflow from a boundary at a specified constant pressure. Not implemented at present.
P_OUTFLOW or *PO*		Outflow to a boundary at a specified constant pressure.
FREE_SLIP_WALL or *FSW*		Velocity gradients at the wall vanish.
NO_SLIP_WALL or *NSW*		All components of the velocity vanish at the wall.

For no-slip or free-slip walls none of the following needs to be specified.

Keyword (dimension)	**Type**	**Description**
BC_EP_g (bc)	DP	Void fraction at the BC plane.
BC_P_g (bc)	DP	Gas pressure at the BC plane.
BC_ROP_s (bc, m)	DP	Macroscopic density of solids phases at the BC plane.
BC_T_g (bc)	DP	Gas phase temperature at the BC plane.
BC_T_s1 (bc)	DP	Solids phase-1 temperature at the BC plane.
BC_T_s2 (bc)	DP	Solids phase-2 temperature at the BC plane.
BC_X_g (bc, n)	DP	Mass fraction of gas species n at the BC plane.

Keyword (dimension)	**Type**	**Description**
BC_X_s (bc, m, n)	DP	Mass fraction of solids phase-m, species n at the BC plane.
BC_U_g (bc)	DP	x-component of gas velocity at the BC plane.
BC_U_s (bc, m)	DP	x-component of solids-phase velocity at the BC plane.
BC_V_g (bc)	DP	y-component of gas velocity at the BC plane.
BC_V_s (bc, m)	DP	y-component of solids-phase velocity at the BC plane.
BC_W_g (bc)	DP	z-component of gas velocity at the BC plane.
BC_W_s (bc, m)	DP	z-component of solids-phase velocity at the BC plane.

For a mass inflow boundary, instead of specifying the normal velocity at a boundary, the gas and solids flow rates may be specified as the volumetric or mass flow rates. If the volumetric or mass flow rate is specified, MFIX will calculate the velocity normal to the boundary. The velocity calculated by MFIX, however, may differ from the velocity calculated based on the physical dimensions of the port because the simulated dimensions may not be exactly equal to the physical dimensions. Specify positive values for all the flow rates. MFIX will assign the correct sign to the computed velocity values.

If the mass or volumetric flow rate is specified for a mass outflow boundary condition, then at every interval BC_DT_0, MFIX will adjust the normal velocity so that the average computed-outflow rate is equal to the specified value. The user is cautioned, however, that if unrealistic mass flow rates are specified, the computations may become unstable. It is better to specify the velocity at the mass outflow boundary, if some amount of fluctuation in the mass outflow rate is tolerable.

Keyword (dimension)	**Type**	**Description**
BC_VOLFLOW_g(bc)	DP	Gas volumetric flow rate through the boundary.
BC_VOLFLOW_s(bc, m)	DP	Solids volumetric flow rate through the boundary.
BC_MASSFLOW_g(bc)	DP	Gas mass flow rate through the boundary.
BC_MASSFLOW_s(bc, m)	DP	Solids mass flow rate through the boundary.

MFIX allows the specification of a transient jet with its velocity fluctuating between two values. The jet conditions will override the steady condition specified for the normal velocity. Therefore, if there is no transient jet, do not specify any of the following, except BC_DT_0, which may be required for mass outflow conditions.

Keyword (dimension)	**Type**	**Description**
BC_DT_0 (bc)	DP	The interval at the beginning when the normal velocity at the boundary is equal to BC_Jet_g0. When doing a restart run this value and BC_Jet_g0 should be specified such that the transient jet continues correctly. MFIX does not store the jet conditions. For MASS_OUTFLOW boundary condition, BC_DT_0 is the time period for averaging and printing the outflow rates. The adjustment of velocities to get a specified mass or volumetric flow rate is based on the average outflow rate.
BC_Jet_g0 (bc)	DP	Value of normal velocity during the initial interval BC_DT_0.
BC_DT_h (bc)	DP	The interval when normal velocity is equal to BC_Jet_gh.
BC_Jet_gh (bc)	DP	Value of normal velocity during the interval BC_DT_h.
BC_DT_l (bc)	DP	The interval when normal velocity is equal to BC_Jet_gl.
BC_Jet_gl (bc)	DP	Value of normal velocity during the interval BC_DT_l.

D.8 Internal Surfaces

Internal surfaces (IS) are normal to one of the coordinate directions and coincide with one of the faces of the scalar control volume. One of the three pairs of coordinates is equal. The surface is defined by the constant coordinates of each of the four edges, which can be specified with physical coordinates or cell indices, and the two equal values for the direction normal to the face, which can only be specified with physical coordinates. If cell sizes are not small enough to resolve a surface specified using physical coordinates, MFIX will indicate this problem with an error message.

To specify a large number of internal surfaces in a region, a 3D region may be specified. Then add a prefix (X_, Y_, or Z_) to indicate the direction of the internal surfaces; e.g., X_IMPERMEABLE specifies impermeable internal surfaces parallel to the X coordinate.

Internal surfaces act as free-slip walls in stress computations. This default condition cannot be changed.

Keyword (dimension)	**Type**	**Description**
IS_X_w (is)	DP	x coordinate of the west face or edge.
IS_X_e (is)	DP	x coordinate of the east face or edge.
IS_Y_s (is)	DP	y coordinate of the south face or edge.
IS_Y_n (is)	DP	y coordinate of the north face or edge.
IS_Z_b (is)	DP	z coordinate of the bottom face or edge.
IS_Z_t (is)	DP	z coordinate of the top face or edge.
IS_I_w (is)	I	i index of the west-most cell.
IS_I_e (is)	I	i index of the east-most cell.
IS_J_s (is)	I	j index of the south-most cell.
IS_J_n (is)	I	j index of the north-most cell.
IS_K_b (is)	I	k index of the bottom-most cell.
IS_K_t (is)	I	k index of the top-most cell.
IS_TYPE(is)	C	Type of internal surface:
IMPERMEABLE or *IP*		No gas or solids flow through the internal surface.

Keyword (dimension)	Type	Description
SEMIPERMEABLE or *SP*		Only gas flows through the surface. Solids velocity is zero or a user-specified fixed value.
IS_PC (is, 2)	DP	1. permeability; 2. inertial resistance coefficient. These values need to be specified for semipermeable surfaces only. The thickness used for pressure drop computation is that of the momentum cell (DX_e, DY_n, or DZ_t). To turn off the resistance, use a large value for permeability (1.E32) and a small value for inertial resistance coefficient (0.0).
IS_VEL (is, m)	DP	Value of fixed solids velocity through semipermeable surfaces.

D.9 Output Control

RES_DT	DP	Interval at which restart (.*RES*) file is updated.
SPX_DT(7)	DP	Interval at which .*SPx* files are updated.
.SP1		þ$_g$ (EP_g)
.SP2		P_g, P^* (P_g, P_star)
.SP3		u_g, v_g, w_g (U_g, V_g, W_g)
.SP4		u_{sm}, v_{sm}, w_{sm} (U_s, V_s, W_s)
.SP5		þ$_{sm}$ þ$_{sm}$ (ROP_s)
.SP6		T_g, T_{s1}, T_{s2} (T_g, T_s1, T_s2)
.SP7		X_{gn}, X_{smn} (X_g, X_s)
OUT_DT	DP	Interval at which standard output (.*OUT*) file is updated.
USR_DT(5)	DP	Interval at which user-defined outputs are written from the subroutine WRITE_USR1.
NLOG	I	Interval in number of time steps at which .*LOG* file is written.