Multiphase Flow with Interphase eXchanges Version MFIX-2007-4 (Date: 12/31/2007)

Notice

Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed or represents that its use would not infringe privately owned rights.

- MFIX is provided without any user support for applications in the user's immediate organization. It should not be redistributed in whole or in part.
- The use of MFIX is to be acknowledged in any published paper based on computations using this software by citing the MFIX theory manual. Some of the submodels are being developed by researchers outside of NETL. The use of such submodels is to be acknowledged by citing the appropriate papers of the developers of the submodels.
- The authors would appreciate receiving any reports of bugs or other difficulties with the software, enhancements to the software, and accounts of practical applications of this software.

Table of Contents

1. Setting Up and Running MFIX on a UNIX/LINUX Workstation

Please read this file carefully and if you still have problems with installation or use, send an email to mfix-help@lists.mfix.org if you have any. Before sending a question, please subscribe to mfix-help mailing list at http://lists.mfix.org/wws/subrequest/mfix-help. Once subscribed, please search the archives at http://lists.mfix.org/wws/arc/mfix-help as your question might be already answered.

1.1 Creating MFIX Directory

To install MFIX (version m.n) from the tar file, go to the home directory and type

```
gunzip -d mfix.tar.gz
tar xvf mfix.tar
```

This procedure will create the directory mfix, which contains six subdirectories: *doc, model, post_mfix, tests, tutorials, and tools*:

Subdirectory	Contents
model	MFIX source files
post_mfix	Post_mfix source files
tutorials	Several example problems and corresponding pdf files describing some of the tutorials exists, which are useful for a new user
tests	test simulations used to verify the code during development
tools	development tools
doc	MFIX manuals in pdf format

The postprocessing code *ani_mfix* is used to view MFIX results as animations. A linux binary is available on the download page.

1.2 Installing the MFIX Code

For convenience, create aliases by adding the following lines to the *.login* or the .cshrc file (assuming MFIX was installed in the home directory):

```
alias ani ~/mfix/ani_mfix/ani_mfix
alias post ~/mfix/post_mfix/post_mfix
alias mkmfix sh ~/mfix/model/make_mfix
```

Note that makefile should be invoked as “sh make_mfix.” Erroneously invoking it as “make_mfix” will cause the user defined files not to be used for the build.

Then the post-processors can be activated from any directory by typing **ani** or **post**, and the mfix make file can be activated by typing **mkmfix**.

For the first time, the user should logout and login or type: **source .login** so that the aliases are defined. This step needs to be done only once.

1.3 Starting a New Run

Create a separate subdirectory for each run.

If you want to modify MFIX subroutines, do the following. Copy the subroutine from the *mfix/model* directory into the run directory. Modify the file copied to the run directory. Do not modify the files in the *mfix/model* directory.

Create an MFIX executable file by typing **mkmfix**. The MFIX executable file *mfix.exe* will be created and copied into the run-directory.

The following options are available for different compilations:

1. For Shared Memory Executable, input >y= for the SMP option.
2. For Distributed Memory Executable , input >y= for the DMP option.
3. For Debug option, input > y= for the debug option.
4. To force recompilation of user defined routines, input >y= at this prompt. In most cases it is not necessary to select this option. This option is needed only in the rare instances when the run directory contains a module file (e.g., *run_mod.f*) and another file (e.g., *usr0.f*), in which the corresponding use statement has been inserted (e.g., use run). The makefile will not be aware of this dependency (*usr0* depends upon *run*). So, in this example, changes made in *run_mod.f* will not cause the (necessary) recompilation of *usr0.f*, unless this option is selected.
5. Choose the appropriate compiler selection options.

If all the above options are unselected, an optimized serial version executable is produced for that particular platform.

Write an MFIX data file and name it *mfix.dat*. The key words are listed at the end of this file. Examples are available in the *mfix/tutorials* directory.

Run MFIX by typing **mfix.exe**, or run in batch mode by typing **mfix.exe &**, or run in batch mode with the screen output redirected to a file (FULL_LOG = .TRUE. in the data file), for example, *screen.log* , by typing **mfix.exe > screen.log &. For csh or tcshrc shell users, mfix.exe |& tee screen.log & will enable both directing the screen output to a file and displaying the output**

on the screen at the same time or one can perform 'tail –f screen.log.' (Note that the file *screen.log* can become very large.)

If the code was compiled as a SMP version, then you will be prompted to enter the number of processors to be used for the run.

Run the DMP version using **mpirun -np <Number of Processors> mfix.exe** . This command may be different on other machines (e.g., **mpprun -n <Number of Processors> mfix.exe**). Recent versions of MPI library (i.e., MPICH2 on Linux clusters) require mpi daemons running on the compute nodes prior to the launch of any mpi executable. Please make sure the appropriate MPI initialization procedures are followed and the simple MPI examples run successfully to verify MPI setup prior to the launch of MFIX executable in DMP mode.

MFIX results can be retrieved, even while the run is in progress. To activate visualization, run *ani_mfix* by typing **ani**. When *ani_mfix* is used the first time for a run, you will be prompted for the name of the restart file. Enter the name of the restart file paying attention to the case (e.g., *BUB01.RES*).

The program will create several control files: *ani_mfix.con*, *ani_mfix.ini*, *ani_mfix.mm*, *script.001*.

To retrieve and manipulate data and to create special restart files, run the post_processor *post_mfix* by typing **post**. *post_mfix* will prompt the user for the run name. Enter the run name (e.g., *BUB01*).

1.4 Modifying the Post-Processing Codes

To make *post_mfix*, first change the directory to the *mfix/post_mfix* directory. If user-defined post-processing is required, modify the *usr_post.f* file in the directory. Then type **make_post**. Note that the *mfix/model* is required for creating a *post_mfix* executable.

2. Setting Up and Running MFIX on Windows Workstation

2.1 Installing the MFIX Code on Windows OS

This is not very well supported; please see our download page (http://www.mfix.org/members/download.php) for special remarks (http://www.mfix.org/members/wininst.html) and also earlier postings on mfix-help mailing list.

2.2 Installing the MFIX Code using Cygwin on Windows OS

1. Install Cygwin from cygwin.com. To get more familiar with Linux/Unix - you can do a google search on unix primer (http://www.google.com/search?q=unix++primer) or you can see the top hit (http://bignosebird.com/unix.shtml)

2. Check that Devel and Editors are installed (To the right of the words will be "default"; Left button click on the word "default" until it says "install")
3. Make sure that make and ex are installed

   - $ which ex
     /usr/bin/ex

     $ which make
     /usr/bin/make

4. Now install g95 from g95.org

   - Check that g95 is available (by typing g95 on console, you should get 'g95: no input files')

5. Get development version of MFIX from http://www.mfix.org/members/download_develop.php
6. issue this command "tar xzvf mfix.tar.gz"
7. go to tutorials/fluidBed1 by typing "cd mfix-old/tutorials/fluidBed1"
8. Issue the command "sh ../../model/make_mfix"

   - This should start the compilation process and depending on the computer this might take a long time.
   - At the end you should have the executable mfix.exe in this directory. To list the files you can use the 'ls' command

9. Now issue the command "nohup ./mfix.exe > out1&" - this will launch the program in the background.
10. You can see the output from the out file by issuing command "tail -f out1". This would stop as soon as the program finished executing.
11. You can visualize the output using animate mfix for windows and that can be downloaded from http://www.mfix.org/members/download_develop.php. There are other options like paraview (http://www.mfix.org/members/develop/paraview/Instructions.php)
12. Any questions, email mfix-help@lists.mfix.org

3. MFIX at Run Time

3.1 MFIX Output and Messages

MFIX output is stored in nine *.*SPx* binary files. The restart info is periodically written to the *.*RES* binary file. The text file *.*OUT* echos the input, shows the numerical cell distribution, and, if OUT_DT is defined, prints the field variables at the specified intervals.

The text file *.*LOG* contains run information. For the DMP version, a LOG file is created for each of the processors and they are numbered as *Name###.LOG*.

Messages about the run are written to the *.LOG* file. The progress of the run will be displayed at the terminal as shown below if the data file specifies FULL_LOG = .TRUE.

Time = 0.92965 Dt = 0.13930E-02 CPU time left = 54.672 s

Nit	P0	P1	U0	V0	U1	V1	Max Res
1	1.5E-02	2.5E+01	1.3E-03	3.3E-03	3.2E-03	6.3E-03	P0
2	1.	5.0E-02	2.8E-03	6.3E-03	2.5E-03	3.8E-03	P1
3	0.1	2.1E-02	1.2E-03	3.7E-03	1.1E-03	1.8E-03	P0
4	5.2E-02	8.1E-03	6.5E-04	2.1E-03	4.8E-04	8.6E-04	P0
5	3.0E-02	3.9E-03	4.1E-04	1.3E-03	2.5E-04	4.3E-04	P0

The first line shows the time, the time-step, and the CPU time remaining to complete the run. The CPU time remaining is not accurate, especially in the beginning of the run. For each time-step, the normalized residuals for various equations are written out every iteration.

MFIX uses a variable time step, which is automatically adjusted within user-defined limits to reduce the run time. At large values of Dt, the iterations may not converge. When this happens, the time step size is successively reduced until convergence is obtained. Messages about divergence and recovery are displayed on the terminal and in the *.LOG* file, if FULL_LOG = .TRUE.

The subsequent lines display the iteration number, the normalized residuals for various equations (e.g., gas continuity, solids continuity, x and y gas momentum, and x and y solids momentum), and the equation with the maximum residual. The residuals P0 and P1 are normalized only when Nit>1. The residuals displayed can be selected with the keyword RESID_STRING.

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 (1) incorrect format for the name-list input, (2) unknown (misspelt) variable name, or (3) the dimension of the name-list item is too small. For example, if the dimension of DX is set as 5000 (DIM_I in *param_mod.f*), and if the input data file contains an entry DX(5001), 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 it is certain that giving a default value is reasonable.

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.

Every NLOG (see Appendix) 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 step, MFIX will print out the number of iterations during the previous time step and the total solids inventory in the reactor.

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 step.

A message is written to the *.LOG* file whenever the *.RES* and *.SPx* files are written. This message also shows an approximate value of the cumulative disk space usage in megabytes.

3.2 Restarting a Run

A run is restarted by rerunning MFIX after typing RUN_TYPE = 'restart_1' in *mfix.dat*. The old *.OUT* file will be overwritten. The *.LOG* messages will be appended to the old *.LOG* file.

There are three other types of restarts; refer to the Appendix for details.

3.3 When the Run Does Not Converge

Initial non-convergence: Ensure that the initial conditions are physically realistic. If in the initial time step, the run displays NaN (Not-a-Number) for any residual, reduce the initial time step, since automatic time step reduction will become ineffective. If time step reductions do not help, recheck the problem setup.

Holding the time step constant (DT_FAC=1) and ignoring the stalling of iterations (DETECT_STALL=.FALSE.) may help in overcoming initial nonconvergence. Often a better initial condition will aid convergence. For example, using a hydrostatic rather than a uniform pressure distribution as the initial condition will aid convergence in fluidized-bed simulations.

If there are computational regions where the solids tend to compact (i.e., solids volume fraction less than EP_star), the convergence can be improved by reducing UR_FAC(2) below the default value of 0.5.

Convergence is often difficult with higher order discretization methods. First order upwinding may be used to overcome initial transients and then the higher order method may be turned on. Also, higher-order methods such as van Leer and minmod give faster convergence than methods such as superbee and ULTRA-QUICK.

4. Keywords in Input Data File (*mfix.dat*)

[ ] indicates the default value.

The symbols used in the table are as follows:

Dimension	Description	Type	Description
1	Cell number in x, y, or z direction	C	Character
M	Solids-phase number	DP	Double Precision
N	Species number	I	Integer
Ic	Initial condition number	L	Logical
Bc	Boundary condition number		
Is	Internal surface number		
Usr	User-defined output number		

4.1 Run Control

Keyword (dimension)	Type	Description
RUN_NAME	C	Name used to create 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 (g, cm, s, cal).
SI		All input and output in SI units (kg, m, s, J).
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. (Do not use)
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. (Do not use)
TIME	DP	Start-time of the run.
TSTOP	DP	Stop-time of the run.
DT	DP	Starting time step. If DT is not defined, a steady-state calculation will be performed.
DT_MAX [1.0]	DP	Maximum time step.
DT_MIN [1E-6]	DP	Minimum time step.
DT_FAC [0.9]	DP	Factor for adjusting time step. Should be less than 1. Use a value of 1 to hold the time step constant.
DETECT_STALL	L	
[.TRUE.]		Reduce time step if the residuals sum does not decrease.
.FALSE.		Do not reduce time step for stalled iterations.
MODEL_B	L	Momentum equations.
[.FALSE.]		Model A
.TRUE.		Model B
MOMENTUM_X_EQ(m)	L	(m=0 indicates gas phase)
[.TRUE.]		Solve X-momentum equations of phase m.
.FALSE.		Do not solve X-momentum equations of phase m. Beware of inconsistencies when the momentum equations are turned off; e.g., 2-D developing flow with only Y-momentum should not specify no-slip-walls.
MOMENTUM_Y_EQ(m)	L	(m=0 indicates gas phase)
[.TRUE.]		Solve Y-momentum equations of phase m.

.FALSE.		Do not solve Y-momentum equations of phase m.
MOMENTUM_Z_EQ(m)	L	(m=0 indicates gas phase)
[.TRUE.]		Solve Z-momentum equations of phase m.
.FALSE.		Do not solve Z-momentum equations of phase m.
ENERGY_EQ	L	
[.TRUE.]		Solve energy equations
.FALSE.		Do not solve energy equations
SPECIES_EQ(m)	L	(m=0 indicates gas phase)
[.TRUE.]		Solve species equations of phase m. To solve species equation with no chemical reactions, copy the file mfix/model/rrates.f into run directory and remove the first two executable lines as explained in the comments. No other change is needed in that file.
.FALSE.		Do not solve species equations of phase m.
GRANULAR_ENERGY [.FALSE.]	L	Use the granular energy transport equation (pde) as opposed to the algebraic (alg) equation formulation.
SIMONIN [.FALSE.]	L	Use Simonin model (see ~mfix/doc/Simonin_Ahmadi_Models.pdf for details).
AHMADI [.FALSE.]	L	Use Ahmadi model (see ~mfix/doc/Simonin_Ahmadi_Models.pdf for details).
JENKINS [.FALSE.]	L	Use Jenkins small frictional boundary condition (see ~mfix/doc/Simonin_Ahmadi_Models.pdf for details).
FRICTION [.FALSE.]	L	Use the Schaeffer model when .FALSE., or use the Princeton model when .TRUE.

The combination of the keywords GRANULAR_ENERGY and FRICTION invokes different solids stress models as shown below:

GRANULAR_ENERGY = .FALSE.

EP_g < EP_star > Schaeffer

EP_g >= EP_star > viscous (algebraic)

GRANULAR_ENERGY = .TRUE.

FRICTION = .TRUE.

EP_s(IJK,M) > EPS_f_min > Princeton + viscous (pde)

EP_s(IJK,M) < EPS_f_min > viscous (pde)

FRICTION = .FALSE.

EP_g < EP_star > Schaeffer + viscous (pde)

EP_g >= EP_star > viscous (pde)

Keyword (dimension)	**Type**	**Description**
SAVAGE	I	For a term appearing in the frictional stress model invoked with FRICTION = .TRUE.
0		Use S:S in the frictional stress model.
[1]		Use an alternate form suggested by Savage.
2		An appropriate combination of the above two forms.
SCHAEFFER	L	
[.TRUE.]		If set to false with FRICTION = .FALSE., then the model will not have any frictional viscosity.
BLENDING_STRESS [.FALSE.]	L	This will turn on the blending function to blend the Schaeffer stresses with that of kinetic theory around ε^*. The default is hyperbolic tangent function for blending (TANH_BLEND = .TRUE.) and one could also utilize a scaled and truncated sigmoidal function (SIGM_BLEND=.TRUE.).
YU_STANDISH	L	
[.FALSE.]		Use Yu and Standish correlation to compute maximum packing for polydisperse systems.
FEDORS_LANDEL	L	
[.FALSE.]		Use Fedors and Landel correlation to compute maximum packing for a binary (only) mixture of powders.
CALL_USR	L	
.TRUE.		Call user-defined subroutines.
[.FALSE.]		Do not call user-defined subroutines.

NRR [0]	I	The number of user defined chemical reactions stored in the *.SPA file. See Section 4.10 Chemical Reactions.
k-ε Equation for Gas-Phase Turbulence		The numerical parameters (like under-relaxation) are the same as the ones for SCALAR (index = 9) Mu_gmax, which is the maximum value of gas turbulent viscosity must be set in order to use the K-Epsilon model. There is no default for this value and the code will not run if Mu_gmax is not set. A value Mu_gmax =1.E+03 is recommended (it is only used in calc_mu_g.f) All walls must be defined (NSW, FSW or PSW) in order to use standard wall functions. If a user does not specify a wall type, he/she will not obtain the typical turbulent profile in wall-bounded flows.
K_Epsilon [.FALSE.]	L	When activated the k-ε turbulence model (for single-phase flow) is solved using standard wall functions.
KT_TYPE [UNDEFINED]	C	Solids phase stress model. Models include the kinetic theories of Iddir & Arastoopour (2005): IA_NONEP. To specify a kinetic theory GRANULAR_ENERGY must be .TRUE. If KT_TYPE is left undefined (default) the viscous model is based on the theory of Lun et al. (1984). The IA theory was modified to include the effects of interstitial fluid on two transport coefficients (conductivity and viscosity) as well as a consistency check to ensure that the stresses of two or more identical solids phases will add to that of a single solids phase. These modifications can be turned off by changing the default value of switch_IA to .FALSE. in constant_mod.f. Note that for the IA theory, the definition of granular temperature includes particle mass (unlike the Lun et al. (1984) theory).

DRAG_TYPE [SYAM_OBRIEN]	C	Drag models include: SYAM_OBRIEN, GIDASPOW, GIDASPW_PCF, GIDASPOW_BLEND, GIDASPOW_BLEND_PCF, WEN_YU, WEN_YU_PCF, KOCH_HILL, KOCH_HILL_PCF, BVK. (see drag_gs.f for details) In the model SYAM_OBRIEN two additional parameters may be specified: drag_c1 and drag_d1 (see section 4.2) The extension _PCF following the specified drag model indicates that the polydisperse correction factor proposed by Beetstra et al. (2007) is implemented with that model.
RDF_TYPE [LEBOWITZ]	C	Radial distribution function at contact for polydisperse systems include: LEBOWITZ, MODIFIED_LEBOWITZ, MANSOORI and MODIFIED_MANSOORI. (Do not specify any RDF for monodisperse systems. “Carnahan-Starling” is the only option available.)

4.2 Physical Parameters

Keyword (dimension)	**Type**	**Description**
C(100)	DP	User defined constants.
C_NAME(100)	C	Name of user-defined constants (20 characters long). These character strings are used only to identify user-defined constants (c) in the .OUT file.
C_e	DP	Coefficient of restitution for particle-particle collisions. (MFIX 1.94 keyword _e_).
e_w [1.0]	DP	Coefficient of restitution for particle-wall collisions.
PHIP [0.6]	DP	Specularity coefficient associated with particle-wall collisions.
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.

Phi_w [0.0]	DP	Angle of internal friction (in degrees) at walls. Set this value to non-zero (phi_w = 11.31 means tan_phi_w = 0.2) when using Jenkins boundary condition.
eps_f_min [0.5]	DP	Minimum solids fraction above which friction sets in. (when FRICTION = .TRUE.)
EP_S_MAX(MMAX) [1.0 – ep_star]	DP	Maximum solids volume fraction at packing for polydisperse systems (more than one solids phase used). The value of EP_star may change during the computation if solids phases with different particle diameters are specified and Yu_Standish or Fedors_Landel correlations are used.
SEGREGATION_SLOPE_COEFFICIENT [0.0]	DP	Used in calculating the initial slope of segregation: see Gera et al. (2003) - recommended value 0.3. Increasing this coefficient results in decrease in segregation of particles in binary mixtures.
L_scale0 [0.0]	DP	Value of turbulent length initialized. This may be overwritten in specific regions with the keyword IC_L_scale.
Mu_gmax	DP	Maximum value of the turbulent viscosity of the fluid.
V_ex	DP	Excluded volume in Boyle-Massoudi stress.
[0.0]		B-M stress is turned off.
P_ref [0.0]	DP	Reference pressure.
P_scale [1.0]	DP	Scale factor for pressure.
GRAVITY	DP	Gravitational acceleration.
[980.7]		By default, the gravity force acts in the _ve y-direction. Modify file b_force2.inc to change the body force term.
Drag_c1 and Drag_d1 drag_c1 [0.8] drag_d1 [2.65]	DP	Quantities for calibrating Syamlal-O'Brien drag correlation using Umf data. These are determined using the Umf spreadsheet (http://www.mfix.org/members/develop/umf.xls). When these are undefined the default values for these constants are used.

4.3 Numerical Parameters

Keyword (dimension)	Type	Description
MAX_NIT [500]	I	Maximum number of iterations.
NORM_g [Use the residual from the first iteration.]	DP	Factor to normalize the gas continuity equation residual. Setting Norm_g = 0 invokes a normalization method based on the dominant term in the continuity equation. This setting may speed up calculations, especially near a steady state and for incompressible fluids. But the number of pressure iterations may need to be increased, LEQ_IT(1), to ensure mass balance.
NORM_s [Use the residual from the first iteration.]	DP	Factor to normalize the solids continuity equation residual. Setting Norm_s = 0 invokes a normalization method based on the dominant term in the continuity equation. This setting may speed up calculations, especially near a steady state. But the number of pressure iterations may need to be increased, LEQ_IT(2), to ensure mass balance.
TOL_RESID [1E-3]	DP	Maximum residual at convergence (continuity+momentum).
TOL_RESID_Th [1E-4]	DP	Maximum residual at convergence (granular energy).
TOL_RESID_T [1E-4]	DP	Maximum residual at convergence (energy).
TOL_RESID_X [1E-4]	DP	Maximum residual at convergence (species balance).
TOL_RESID_Scalar [1E-4]	DP	Maximum residual at convergence (scalar balances.)
TOL_DIVERGE [1E+4]	DP	Minimum residual for declaring divergence. When the fluid is incompressible, the velocity residuals take large values in the second iteration (e.g., 1E+8) and then drop down to a low value in the third iteration (e.g., 0.1). In such cases, it is desirable to increase this setting.

Max_Inlet_Vel_Fac [1]	DP	The code declares divergence if the velocity anywhere in the domain exceeds a maximum value. This maximum value is automatically determined from the boundary values. The user may scale the maximum value by adjusting this scale factor.

The next keywords LEQ_IT, LEQ_METHOD, LEQ_SWEEP, LEQ_TOL, UR_FAC, and DISCRETIZE are dimensioned for the nine types of equations:

Index	Equation Type
1	gas pressure
2	solids volume fraction
3	gas and solids u-momentum
4	gas and solids v-momentum
5	gas and solids w-momentum
6	Temperature
7	species mass fractions
8	granular temperature
9	user-defined scalar

For example, LEQ_IT(3) = 10 will make MFIX use 10 linear equation iterations while solving the gas and solids u-momentum equation (Equation Type =3).

Keyword (dimension)	**Type**	**Description**

LEQ_IT(9) [20 for 1 and 2] [5 for 2-5] [15 for 6-9]	I	Number of iterations in the linear equation solver. The nine values are for the nine types of equations noted above. The same convention holds for LEQ_METHOD, UR_FAC, and DISCRETIZE. If the residual of an equation is less than the convergence criterion, MFIX makes LEQ_IT equal to the lesser of 5 and the user-defined value and LEQ_METHOD equal to 1.
LEQ_METHOD(9) [2 for all equations]	I	The method used in the linear equation solver: 1. SOR 2. BiCGSTAB 3. GMRES 5. CG
LEQ_SWEEP(9) [RSRS]	C	The sweep direction used in the linear equation solver. The sweep direction for preconditioning line relaxation; e.g., if LEQ_SWEEP = "ISIS", 1 sweep with do IK loop followed by send_recv (repeated twice), or if LEQ_SWEEP = "RSRS",1 red-black sweep with do IK loop followed by send_recv (repeated twice), or if LEQ_SWEEP = "ASAS",1 red-black sweep with do IK loop followed by do JK followed by IJ followed by send_recv (repeated twice). Only used by BiCGSTAB and CG.
LEQ_TOL(9) [1.0D-4]	DP	The tolerance, if used, in linear equation solvers. Only used by BiCGSTAB and CG.
LEQ_PC(9) [LINE]	C	The preconditioner used for the sweeps in the linear solver LINE - Line relaxation DIAG - Diagonal Scaling NONE - No preconditioner
UR_FAC(9) [0.8 for 1, 6, 9] [0.5 for 2, 3, 4, 5, 8] [1.0 for 7]	DP	Under relaxation factors for seven types of equations. Reducing UR_FAC(2) will help convergence in problems in which the solids tend to pack.
DEF_COR [.FALSE.]	L	If true, use deferred correction method for implementing higher order discretization. Otherwise, use down-wind factor method (default).
DISCRETIZE(9)	I	Discretization scheme for seven types of equations.

[0]		First-order upwinding.
1		First-order upwinding (using down-wind factors).
2		Superbee (recommended method).
3		SMART.
4		ULTRA-QUICK.
5		QUICKEST (does not work).
6		MUSCL.
7		van Leer.
8		Minmod.
FPFOI [.FALSE.]	L	Four point fourth order interpolation and is upstream biased. If this scheme is chosen and discretize(*) < 2, discretize(*) is defaulted to 2. If you chose this scheme, set the C_FAC value between 0 and 1.
C_FAC [UNDEFINED]	DP	Factor used in the universal limiter (when FPFOI is set .TRUE.) and can be any value in the set (0,1). The choice of 1 will give (diffusion) first order upwinding and as this value becomes closer to 0 the scheme becomes more compressive.
CN_ON [.FALSE.]	L	Implicit Euler based temporal discretization scheme employed (first order accurate in time).
.TRUE.		Crank-Nicholson based temporal disretization scheme employed (second order accurate in time).
Chi_Scheme [.FALSE.]	L	Chi-Scheme, proposed by Darwish and Moukalled (2003), is activated. This scheme guarantees that the set of differenced species mass balance equations has the property that the sum of mass fractions add up to one. When a flux limiter is used with (higher order) spatial discretization schemes it is not guaranteed that the mass fractions add up to one. This problem may be rectified by activating the chi-scheme.

UR_F_gs [1]	DP	The implicitness calculation of the gas-solids drag coefficient may be underrelaxed by changing UR_F_gs, which takes values between 0 to 1: UR_F_gs = 1 (update F_gs every iteration) UR_F_gs = 0 (update F_gs every time step)

4.4 Geometry and Discretization

For 2D simulations, the thickness of the third direction specified should be exact if mass or volumetric flow rates, rather than 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	(Do not use.)
[.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. Enter values from DX(0) to DX(IMAX-1). (Use uniform mesh size with higher-order discretization methods. Also in cylindrical coordinates DX should be kept uniform for strict momentum conservation.)
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	(Do not use.)
[.FALSE]		y direction is considered.
.TRUE.		y direction is not considered.
JMAX	I	Number of cells in the y direction.
DY (1)	DP	Cell sizes in the y direction. Enter values from DY(0) to DY(IMAX-1). (Use uniform mesh size with second-order discretization methods.)

YLENGTH	DP	Reactor length in the y direction.
NO_K	L	
[.FALSE.]		z (θ) direction is considered.
.TRUE.		z (θ) direction is not considered.
KMAX	I	Number of cells in the z (θ) direction.
DZ (1)	DP	Cell sizes in the z (θ) direction. Enter values from DZ(0) to DZ(IMAX-1). (Use uniform mesh size with second-order discretization methods.)
ZLENGTH	DP	Reactor length in the z (θ) direction.
CYCLIC_X	L	Flag for making the x-direction cyclic without pressure drop. No other boundary conditions for the x-direction should be specified.
[.FALSE.]		No cyclic condition at X-boundary.
.TRUE.		Cyclic condition at X-boundary.
CYCLIC_X_PD	L	Flag for making the x-direction cyclic with pressure drop. If the keyword Flux_g is given a value this becomes a cyclic boundary condition with specified mass flux. No other boundary conditions for the x-direction should be specified.
[.FALSE.]		No cyclic condition at X-boundary.
.TRUE.		Cyclic condition with pressure drop at X-boundary.
DELP_X	DP	Fluid pressure drop across XLENGTH when a cyclic boundary condition with pressure drop is imposed in the x-direction.
CYCLIC_Y	L	Flag for making the y-direction cyclic without pressure drop. No other boundary conditions for the y-direction should be specified.
[.FALSE.]		No cyclic condition at Y-boundary.
.TRUE.		Cyclic condition at X-boundary.

CYCLIC_Y_PD	L	Flag for making the y-direction cyclic with pressure drop. If the keyword Flux_g is given a value this becomes a cyclic boundary condition with specified mass flux. No other boundary conditions for the y-direction should be specified.
[.FALSE.]		No cyclic condition at Y-boundary.
.TRUE.		Cyclic condition with pressure drop at Y-boundary.
DELP_Y	DP	Fluid pressure drop across YLENGTH when a cyclic boundary condition with pressure drop is imposed in the y-direction.
CYCLIC_Z	L	Flag for making the z-direction cyclic without pressure drop. No other boundary conditions for the z-direction should be specified.
[.FALSE.]		No cyclic condition at Z-boundary.
.TRUE.		Cyclic condition at Z-boundary.
CYCLIC_Z_PD	L	Flag for making the z-direction cyclic with pressure drop. If the keyword Flux_g is given a value this becomes a cyclic boundary condition with specified mass flux. No other boundary conditions for the z-direction should be specified.
[.FALSE.]		No cyclic condition at Z-boundary.
.TRUE.		Cyclic condition with pressure drop at Z-boundary.
DELP_Z	DP	Fluid pressure drop across ZLENGTH when a cyclic boundary condition with pressure drop is imposed in the z-direction.
SHEAR	L	If .TRUE. imposes a mean shear on the flow field as a linear function of _x_ coordinate. This feature should only be used when CYCLIC_X=.TRUE. Also, the keyword V-sh needs to be set.
V_sh	DP	Specifies the mean _y_ velocity component at the eastern boundary of the domain (V_sh), and the mean _y_ velocity (-V_sh) at the western boundary of the domain.

Flux_g [UNDEFINED]	DP	If a value is specified (in units of g/cm^2.s), the domain-averaged gas flux is held constant at that value in simulations over a periodic domain. A pair of boundaries specified as “periodic with fixed pressure drop” is then treated as “periodic with fixed mass flux.” Even for this case a pressure drop must also be specified, which is used as the initial guess in the simulations.

4.5 Gas Phase

Keyword (dimension)	Type	Description
RO_g0	DP	Specified constant gas density. This value may be set to zero to make the drag zero and to simulate granular flow in a vacuum. For this case, users may turn off solving for gas momentum equations to accelerate convergence.
MU_g0	DP	Specified constant gas viscosity.
K_g0	DP	Specified constant gas conductivity.
C_pg0	DP	Specified constant gas specific heat.
DIF_g0	DP	Specified constant gas diffusivity.
MW_AVG	DP	Average molecular weight of gas.
MW_g (n)	DP	Molecular weight of gas species n.

4.6 Solids Phase

Keyword (dimension)	Type	Description
MMAX [1]	I	Number of solids phases.
D_p0 (m)	DP	Initial particle diameters, same as the old D_P(m).
RO_s (m)	DP	Particle densities.

NMAX (m)	I	Number of species in phase m. Note that m=0 indicates gas phase.
MU_s0	DP	Specified constant granular viscosity. If this value is specified, then the kinetic theory calculation is turned off and P_s = 0 and Lambda_s = -2/3 MU_s0.
K_s0	DP	Specified constant solids conductivity.
C_ps0	DP	Specified constant solids specific heat.
DIF_s0	DP	Specified constant solids diffusivity.
MW_s (m,n)	DP	Molecular weight of solids phase-m, species n.
EP_star	DP	Packed bed void fraction.
CLOSE_PACKED (m) [.TRUE.]	L	Indicates whether the solids phase forms a packed bed with a void fraction EP_star.

4.7 Initial Conditions

Each initial condition (IC) is specified over a rectangular region (or pie-shaped for cylindrical coordinates) that corresponds to the scalar numerical grid. These are 3D regions: X_w X_e, Y_s Y_n, and Z_t 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 1.9 pi to 0.1 pi as 1.9 pi to 2 pi and 0 to 0.1 pi.

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 wall.
IC_I_e (ic)	I	i index of the east-most wall.
IC_J_s (ic)	I	j index of the south-most wall.
IC_J_n (ic)	I	j index of the north-most wall.
IC_K_b (ic)	I	k index of the bottom-most wall.
IC_K_t (ic)	I	k index of the top-most wall.
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 bulk density (rop_s = ro_s x ep_s) of solids phase-m in the IC region. Users need to specify this IC only for polydisperse flow (MMAX > 1). Users must make sure that summation of (IC_ROP_s(ic,m) / RO_s(m)) over all solids phases is equal to (1.0 – IC_EP_g(ic)).
IC_T_g (ic)	DP	Initial gas phase temperature in the IC region.
IC_T_s (ic, m)	DP	Initial solids phase-m temperature in the IC region.
IC_Theta_m (ic, m)	DP	Initial solids phase-m granular temperature in the IC region.
IC_GAMA_Rg (ic) [0]	DP	Gas phase radiation coefficient in the IC region. Modify file radtn2.inc to change the source term.
IC_T_Rg (ic)	DP	Gas phase radiation temperature in the IC region.

IC_GAMA_Rs (ic, m) [0]	DP	Solids phase-m radiation coefficient in the IC region. Modify file radtn2.inc to change the source term.
IC_T_Rs (ic, m)	DP	Solids phase-m radiation temperature 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) [0]	DP	Initial mass fraction of gas species n.
IC_X_s (ic, m, n) [0]	DP	Initial mass fraction of solids phase-m, species n.
IC_SCALAR (ic, n) [0]	DP	Initial value of Scalar n.

4.8 Boundary Conditions

Boundary conditions (BC) are specified over flow planes or 2D surfaces that are normal to one of the coordinate directions and coincide 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 to specify obstacles in the flow domain. Obstacles are 3D regions, just as for the IC regions: X_w X_e, Y_s Y_n, and Z_t 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.
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. To specify as the west, south, or bottom end of the computational region, add a layer of wall cells to the west, south, or bottom of the PI cells. Users need to specify all scalar quantities and velocity components. The specified values of fluid and solids velocities are only used initially as MFIX computes these values at this inlet boundary.

P_OUTFLOW or PO		Outflow to a boundary at a specified constant pressure. To specify as the west, south, or bottom end of the computational region, add a layer of wall cells to the west, south, or bottom of the PO cells.
FREE_SLIP_WALL or FSW		Velocity gradients at the wall vanish. If BC_JJ_PS is equal to 1, the Johnson-Jackson boundary condition is used for solids. A FSW is equivalent to using a PSW with h_w=0.
NO_SLIP_WALL or NSW		All components of the velocity vanish at the wall. If BC_JJ_PS is equal to 1, the Johnson-Jackson boundary condition is used for solids. A NSW is equivalent to using a PSW with v_w=0 and h_w undefined.
PAR_SLIP_WALL or PSW		Partial slip at the wall implemented as dv/dn + h_w (v – v_w) = 0, where n is the normal pointing from the fluid into the wall. The coefficients h_w and v_w should be specified. For free slip set h_w = 0. For no slip leave h_w undefined (h_w=∞) and set v_w = 0. To set h_w = ∞, leave it unspecified. If BC_JJ_PS is equal to 1, the Johnson-Jackson boundary condition is used for solids.

Specifications for WALL boundary conditions:

Keyword (dimension)	**Type**	**Description**
Momentum Equation		Partial slip at the wall implemented as dv/dn + h_w (v - v_w) = 0, where n is the normal pointing from the fluid into the wall. If the Johnson and Jackson partial slip boundary condition is not used (i.e., BC_JJ_PS(bc) = 0), the coefficients h_w and v_w should be specified. For free slip set h_w = 0. For no slip leave h_w undefined (h_w = ∞) and set v_w = 0. To set h_w = ∞, leave it unspecified.
BC_hw_g (bc) [∞]	DP	Gas phase h_w for partial slip boundary.
BC_hw_s (bc, m) [∞]	DP	Solids phase h_w for partial slip boundary.

BC_Uw_g (bc)	DP	Gas phase U_w for partial slip boundary.
BC_Uw_s (bc, m)	DP	Solids phase U_w for partial slip boundary.
BC_Vw_g (bc)	DP	Gas phase V_w for partial slip boundary.
BC_Vw_s (bc, m)	DP	Solids phase V_w for partial slip boundary.
BC_Ww_g (bc)	DP	Gas phase W_w for partial slip boundary.
BC_Ww_s (bc, m)	DP	Solids phase W_w for partial slip boundary.
BC_JJ_PS (bc)	I	1: Use Johnson and Jackson partial slip bc. 0: Do not use Johnson and Jackson partial slip bc.
[0]		If granular energy transport equation is not solved: (GRANULAR_ENERGY=.FALSE.).
[1]		If granular energy transport equation is solved: (GRANULAR_ENERGY=.TRUE..
Granular Energy Equation		The granular energy boundary condition is implemented as $dT/dn + h_w (T - T_w) = c$, where n is the normal pointing from the fluid into the wall. If the Johnson and Jackson partial slip boundary condition is not used (i.e., BC_JJ_PS(bc) = 0), the coefficients h_w and c should be specified. For specified heat flux set $h_w=0$ and give a value for c. For specified temperature boundary condition leave h_w unspecified ($h_w=\infty$) and give a value for T_w.
BC_Thetaw_m (bc, m)	DP	T_w for granular energy bc.
BC_hw_Theta_m (bc, m) [∞]	DP	H_w for granular energy bc.
BC_C_Theta_m (bc, m)	DP	c for granular energy bc.

Gas and Solids Energy Equations		The thermal boundary condition implemented as $dT/dn + h_w (T - T_w) = c$, where n is the normal pointing from the fluid into the wall. The coefficients h_w, T_w, and c should be specified. $H_w = 0$ => specified heat flux; $h_w = \infty$ => specified temperature boundary condition. To set $h_w = \infty$, leave it unspecified and give a value for Tw.
BC_hw_T_g (bc) [∞]	DP	Gas phase h_w for heat transfer.
BC_hw_T_s (bc, m) [∞]	DP	Solids phase h_w for heat transfer.
BC_Tw_g (bc)	DP	Gas phase T_w for heat transfer.
BC_Tw_s (bc, m)	DP	Solids phase T_w for heat transfer.
BC_C_T_g (bc)	DP	Gas phase C for heat transfer.
BC_C_T_s (bc, m)	DP	Solids phase C for heat transfer.
Gas and Solids Species Equations		The species diffusion boundary condition is implemented as $dX/dn + h_w (X - X_w) = c$, where n is the normal pointing from the fluid into the wall. The coefficients h_w, X_w, and c should be specified. Hw = 0 => specified species diffusion flux; $h_w = \infty$ => specified species concentration at the boundary. To set $h_w = \infty$, leave it unspecified and give a value for X_w.
BC_hwX_g (bc, n) [∞]	DP	The species diffusion boundary condition is implemented as $dX/dn + h_w (X - X_w) = c$, where n is the normal pointing from the fluid into the wall. The coefficients h_w, X_w, and c should be specified. $H_w = 0$ => specified species diffusion flux; $h_w = \infty$ => specified species concentration at the boundary. To set $h_w = \infty$, leave it unspecified and give a value for X_w. Gas phase h_w for mass transfer.
BC_hw_X_s (bc, m, n) [∞]	DP	Solids phase h_w for mass transfer.
BC_Xw_g (bc, n)	DP	Gas phase X_w for mass transfer.

BC_Xw_s (bc, m, n)	DP	Solids phase X_w for mass transfer.
BC_C_X_g (bc, n)	DP	Gas phase C for mass transfer.
BC_C_X_s (bc, m, n)	DP	Solids phase C for mass transfer.
Scalar Transport Equations		The scalar boundary condition is implemented as dS/dn + h_w (S - S_w) = C, where n is the normal pointing from the fluid into the wall. The coefficients h_w, S_w, and c should be specified. Hw = 0 => specified species diffusion flux; h_w = ∞ => specified species concentration at the boundary. To set h_w = ∞ , leave it unspecified and give a value for S_w.
BC_hw_Scalar (bc, n) [∞]	DP	h_w for scalar transfer at the boundary.
BC_ScalarW (bc, n)	DP	X_w for scalar transfer at the boundary.
BC_C_Scalar (bc, n)	DP	C for scalar transfer at the boundary.

Specifications for FLOW boundary conditions:

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	Bulk density of solids phase at the BC plane.
BC_T_g (bc)	DP	Gas phase temperature at the BC plane.
BC_T_s (bc, m)	DP	Solids phase-m temperature at the BC plane.
BC_Theta_m (bc, m)	DP	Solids phase-m granular temperature at the BC plane.
BC_X_g (bc, n) [0]	DP	Mass fraction of gas species n at the BC plane.
BC_X_s (bc, m, n) [0]	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 restarting, 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 conditions, BC_DT_0 is the time period to average and print 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	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_1 (bc)	DP	The interval when normal velocity is equal to BC_Jet_g1.
BC_Jet_g1 (bc)	DP	Value of normal velocity during the interval BC_DT_1.

4.9 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. When IS_Type is specified for such regions, 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 surface.
SEMIPERMEABLE or SP		Only gas flows through the surface. Solids velocity is zero or a user-specified fixed value.
IS_PC (is, 2) (*, 1) = 1.E32 (*, 2) = 0.0	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 the inertial resistance coefficient (0.0).
IS_VEL_S (is, m)	DP	Value of fixed solids velocity through semipermeable surfaces.

4.10 Output Control

Keyword (dimension)	Type	Description
RES_DT	DP	Interval at which restart (.RES) file is updated.
SPX_DT (11)	DP	Interval at which .SPX files are updated.
.SP1		Void fraction (EP_g).
.SP2		Gas pressure, solids pressure (P_g, P_star).

.SP3		Gas velocity (U_g, V_g, W_g).
.SP4		Solids velocity (U_s, V_s, W_s).
.SP5		Solids density (ROP_s).
.SP6		Gas and solids temperature (T_g, T_s1, T_s2).
.SP7		Gas and solids mass fractions (X_g, X-s).
.SP8		Granular temperature (G).
.SP9		User defined scalars.
.SPA		Reaction Rates. (See section 4.11)
.SPB		Turbulence quantities (k and ε).
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 [25]	I	Interval in number of time steps at which .LOG file is written.
FULL_LOG [.FALSE.]	L	If true, display the residuals on the screen and messages about convergence on the screen and in the .LOG file.
RESID_STRING (8)	C	Specify residuals to be printed as 4-character strings. First character specifies the field variable: P - pressure, R - density, U - u velocity, V - v velocity, W - w velocity, T - temperature, X - species mass fraction, G - Granular temperature. The second number specifies the phase (0 for gas). The last two numbers specify the species index; e.g., 'P0' - gas pressure, 'R1' - solids phase 1 density, 'X001'- gas phase, species 1; 'X203' - solids phase 2, species 3; 'K0' - k-ε residuals.
REPORT_MASS_BALANCE_DT [Undefined]	DP	If a value is defined, say 0.1 s, an overall species mass balance is performed and reported in the LOG file. The over all mass balance calculations may slightly slow down the run.

4.11 Chemical Reactions

Chemical reactions can be specified in two ways: in the data file, or in the subroutine rrates. If reactions are specified in the data file, MFIX uses the subroutine rrates0 to calculate all the required quantities. The user need not modify rrates0. If the chemical reactions are not specified in the data file, the user must program them into the subroutine rrates.

To define chemical reactions through the data file, first define species names with the following key word.

Keyword (dimension)	**Type**	**Description**
SPECIES_NAME	C	Names of gas and solids phase species. The first NMAX(0) are the names of gas species. The next NMAX(1) are the names of solids phase-1 species, and so on. All the names should be unique.

The reactions are specified in a reaction block beginning with the string @(RXNS) and ending with the string @(END). The reactions are entered in the following format: reaction name: reaction scheme. r1: CH4 + 2 O2 → CO2 + 2H2O

The reaction rate in g-mol/(m^3.s)is then entered in the following format, starting with the keyword rate:

Keyword	**Name**	**Phase index for Temp** (m)	**Preexponential Factor** (A)	**Temperature Exponent** (n)	**Activation Temperature** (E/R)	**Concentration Dependence**
Rate	r1:	0	6.7E12	0	24358.	[O2]^1.3 [CH4]^0.2

The index m is used to identify the phase temperature to be used in the rate expression, to assign the enthalpy change caused by the reaction, and to determine the phase volume fraction to be used in the rate expression. m=0 is fluid phase, m=1 is solids phase 1, and so on. The rest of the input appears in the rate expression as follows:

rate = A * T_g**n * exp(-(E/R)/T_g)) * EP_g * (RO_g * X_O2/MW_O2)**1.3 * (RO_g * X_CH$/MW_CH4)**0.2

The rate expression may be written on multiple lines. Reverse reactions are denoted by a '<' in the reaction scheme.

The enthalpy change caused by the reaction in cal/g-mol is expressed as follows with the keyword DH. The default value is zero.

Enthalpy Change Keyword	**Reaction Name**	**Enthalpy Change Caused by Reaction** (cal/g-mol)
DH	r1:	-191759

The enthalpy change is assigned to the phase identified by the index m in the rate expression.

The following is an example of an athermal reversible reaction.

```
@( RXNS Start chemical reactions input )
A2R: A <--> R
rate A2R: 0 1.2E17 0 5837 [A]
rate A2R<: 0 2.5E41 0 14897 [R]
DH A2R: 0.0
@( END End chemical reactions input )
```

All the reaction schemes must be specified in the data file. All the reaction rate expressions and enthalpy changes need not be specified in the data file, however. If a rate expression or an enthalpy change is missing from the data file, that information must be added to the subroutine rrates0.

To write out reaction rates to SPx file:

1. In mfix.dat set NRR to the desired number of reaction rates to be written out to the file *.SPA. This number will be less than or equal to the total number of reactions.
2. In rrates.f assign the reaction rates to the variable ReactionRates as follows, for example:

```
ReactionRates(ijk, 1) = <rate of some chemical reaction>
ReactionRates(ijk, 2) = <rate of another chemical reaction>
...
ReactionRates(ijk, NRR) = <rate of another chemical reaction>
```

4.12 Thermochemical Properties

The directory mfix/model/thermochemical contains the database of Burcat and Ruscic (2005) and routines for reading the database. With linkage to this database the users need not manually enter data for molecular weight, specific heat, and heats of reactions. Instead the users will need to enter the names of the species (keyword species_name) in the datafile. MFIX reads the necessary thermochemical data from files in the following order: 1. mfix.dat 2. BURCAT.THR file in the run directory 3. mfix/model/thermochemical/BURCAT.THR. The species names are case sensitive and should match the names in BURCAT.THR exactly (except for trailing blanks and tabs); alternatively aliases can be defined for common species, such as O2, in read_therm.f.

See mfix/tests/thermo for a sample case.

The format of BURCAT.THR file resembles CHEMKIN format, but with several notable differences. All thermochemical data must start below a line that starts with THERMO DATA. The examples (in parenthesis) refer to the sample dataset given at the end. Each entry starts with the unique CAS identifier (74-82-8) for the species, followed by several lines of comments. The data section starts with the species name in columns 1-18 (CH4 RRHO). Common species names could be followed by strings (RRHO) that identify the method used to determine the coefficients. The numbers toward the end of the line are the temperature limits (200 to 6000 K) where the property calculation is valid and the molecular weight (16.04246). Unlike CHEMKIN the common temperature for the high and low temperature branches are not recorded; it is always 1000 K. The next three lines give the fourteen coefficients (seven each for high and low temperature branches) and the enthalpy of formation at 298 K (which is also not included in CHEMKIN format). All the coefficients and the enthalpy of formation are normalized with the gas constant R. The normalized specific heat is given by Cp/R = A5T4 + A4T3 + A3T2 + A2T + A1. For other details see Burcat and Ruscic (2005) report in the file mfix/modelthermochemical/intro.pdf. There are two sets of A coefficients: The low temperature coefficients (AL) that should be used for temperatures in the range Tlow to 1000 K and the high temperature coefficients (AH) that should be used for temperatures in the range 1000K to Thigh. The coefficients are stored in a fixed format (E15.0) as follows:

AH1 AH2 AH3 AH4 Ah5

AH6 AH7 AL1 AL2 AL3

AL4 AL5 AL6 AL7 Formation enthalpy at 298K

For a number of species the lower temperature limit is set at 300 K, 2 degrees above the reference temperature (298 K) used for formation enthalpy calculation. For those species MFIX relaxes the lower limit for Cp calculations to 298 K (see read_database.f) to enable heat of reaction calculation.

Sample dataset:

```
74-82-8
CH4  METHANE  Same as the Anharmonic but calculated Using the RRHO method rather
than the NRRAO2.  Max Lst Sq Error Cp @ 6000. K 0.62%.
CH4   RRHO        g 8/99C  1.H  4.   0.   0.G   200.000  6000.000  B  16.04246 1
 1.91178600E+00 9.60267960E-03-3.38387841E-06 5.38797240E-10-3.19306807E-14    2
-1.00992136E+04 8.48241861E+00 5.14825732E+00-1.37002410E-02 4.93749414E-05    3
-4.91952339E-08 1.70097299E-11-1.02453222E+04-4.63322726E+00-8.97226656E+03    4
```

The database reader is set up such that the database is read only if necessary. One exception is that if gas species properties are needed the reader will look for solids species as well, whether or not needed. Any species name can be specified to get around this problem; e.g., O2, Al2O3.

4.13 User-Defined Subroutines

The user may modify any *.f or *.inc file in MFIX. To modify a file, first copy it from the mfix/model directory into the run directory. Modify only this copy in the run directory; do **NOT** modify the original files in mfix/model. Then, invoke the 'sh make_mfix' command from the run directory. The make_mfix messages will identify the files from the run directory used to create the MFIX executable. All the (MFIX and non-MFIX) *.inc files from the run directory will be used to create the MFIX executable. Only MFIX *.f files from the run directory will be used, however. Non-MFIX *.f files in the run directory will be ignored. To use new Fortran files, include them in one of the MFIX *.f files.

The following is a list of MFIX files that are usually modified to include chemical reactions and user defined scalars:

rrates.f	Chemical reaction rates and heats of reaction.
transport_prop.f	Transport properties.
physical_prop.f	Physical properties.
scalar_prop.f	Properties and source terms in scalar transport equations.

The following routines are used for writing user-defined output:

write_usr0.f	Called once during the run. Can be used for opening user-defined files.
write_usr1.f	Called at intervals defined by USR_DT.

To activate the calls to the following three routines, set call_usr = .TRUE. in the data file:

usr0.f	A subroutine that is called once every run, just before the time-loop begins.
usr1.f	A subroutine that is called once every timestep.
usr2.f	A subroutine that is called once every iteration.
usr3.f	A subroutine that is called once every run, after the time-loop ends.
usrnlst.inc	List of user-defined keywords. These may be used to enter data through the input data file mfix.dat.
usr_init_namelist.f	Initialize user-defined keywords.

usr_mod.f	User-defined module. Include "Use usr" to use user-defined variables in this module. If allocatable arrays are defined in this module, allocate them in usr0.f.

4.14 Parallelization Controls

Keyword (dimension)	Type	Description
NODESI [1]	I	Number of grid blocks in x-direction.
NODESJ [1]	I	Number of grid blocks in y-direction.
NODESK [1]	I	Number of grid blocks in z-direction.
MINIMIZE_DOTPRODUCTS [FALSE]	L	The dot products calculations in BiCGStab are minimized and this should only have impact on parallel runs
SOLVER_STATISTICS [FALSE]	L	Print out additional statistics for parallel runs
DEBUG_RESID [TRUE]	L	If set to false, the residuals are grouped into far fewer global collectives and this should only have impact on parallel runs

NODESI * NODESJ * NODESK must be the same as the number of processors specified using the mpirun (or equivalent command). Otherwise the code will return with an error.

The parallel performance depends on several things and one has to evaluate different options before choosing the right strategy for the problem at hand. For e.g. if the J direction is the strongest coupled direction, the preconditioning for the linear solver will be poor if there is decomposition in that direction. However, since decomposing in all the directions reduces the processor grid surface area to the volume, the communication cost will be less for the same computational grid. The preconditioners are chosen with the keyword LEQ_PC. In addition to LINE relaxation, one can choose the “DIAG” or “NONE” preconditioners that reduces interprocessor communications but would increase the number of linear equation solver iterations. The DIAG and NONE choices for preconditioners may be appropriate for all equations except the continuity (or pressure and volume fraction correction) equations. The parallel performance is greatly dependent on the choices stated here, and some trial an error may be required to determine the right combination of decomposition direction and the choice of preconditioners to get the best performance in production runs.

4.15 Discrete Element Model (DEM)

VARIABLE	TYPE	DESCRIPTION
DISCRETE_ELEMENT [F]	L	Variable to decide to do DEM for solids or not. Must be TRUE to do DEM.
DES_CONTINUUM_COUPLED [F]	L	Decided whether gas and solids are coupled together in flow.
WALLDTSPLIT [F]	L	Treats wall interaction also as a two-particle interaction but accounting for the wall properties. Must be TRUE for DEM.
TSUJI_DRAG [F]	L	To use Tsuji's drag correlation
PARTICLES [UNDEFINED_I]	I	Number of particles
DTSOLID_FACTOR [UNDEFINED]	DP	DT_FLUID/DTSOLID; number of solid marches per fluid time step.
DES_INTERP_ON [FALSE]	L	To turn on the interpolation suite to calculate the individual drag forces on the particles based on the particle locations rather than cell averages.
Boundary Conditions (Not needed for MFIX-DEM coupled simulations)		
DES_PERIODIC_WALLS [F]	L	Periodic wall boundary condition is imposed on any pair of walls.
DES_PERIODIC_WALLS_X [F]	L	Direction of periodicity: X
DES_PERIODIC_WALLS_Y [F]	L	Direction of periodicity: Y
DES_PERIODIC_WALLS_Z [F]	L	Direction of periodicity: Z
INLET_OUTLET [F]	L	If inlet-outlet boundary condition is imposed.
INLET_OUTLET_X [F]	L	Direction of Inlet-Outlet: X
INLET_OUTLET_Y [F]	L	Direction of Inlet-Outlet: Y
INLET_OUTLET_Z [F]	L	Direction of Inlet-Outlet: Z
Neighbor Search Parameters		
DES_NEIGHBOR_SEARCH [1]	I	1= N-Square search; 2 = Quadtree/Octree search; 4 = Cell linked-list based algorithm
NEIGHBOR_SEARCH_N [1]	I	Number of time solid steps when neighbor search is performed

MN [10]	I	Maximum number of neighbors
QLM [1]	I	Number of levels to traverse “up” to move a particle to its new quad
QLN [1]	I	Number of levels to traverse “up” to perform particle neighbor search
INIT_QUAD_COUNT [UNDEFINED_I]	I	Count to initialize quadtree
NEIGHBOR_SEARCH_RAD_RATIO [1000]	DP	Ratio of the particle radius and the distance (imaginary sphere radius) to be traveled by a particle before a neighbor search is performed
MQUAD_FACTOR [1.1]	DP	Factor to create quadtree arrays based on the number of particles
FACTOR_RLM [1.2]	DP	FACTOR_RLM is used in cell-linked list to determine neighbors. Default value is used to populate the neighbor list with more particles than what are actually in contact. It is done to avoid calling neighbor search at every DEM time step.
Particle-particle and Particle-wall contact parameters		
KN [UNDEFINED]	DP	Normal spring constant for inter-particle collision
KT [UNDEFINED]	DP	Tangential spring constant for inter-particle collision
KN_W [UNDEFINED]	DP	Normal spring constant for particle-wall collision
KT_W [UNDEFINED]	DP	Tangential spring constant for particle-wall collision
ETA_DES_N [UNDEFINED]	DP	Normal damping coefficient for inter-particle collision
ETA_DES_T [UNDEFINED]	DP	Tangential damping coefficient for inter-particle collision
ETA_N_W [UNDEFINED]	DP	Normal damping coefficient for particle-wall collision
ETA_T_W [UNDEFINED]	DP	Tangential damping coefficient for particle-wall collision
MEW [UNDEFINED]	DP	Particle friction coefficient
MEW_W [UNDEFINED]	DP	Wall friction coefficient
DES_F [UNDEFINED]	DP	Frequency of bottom wall oscillation

DES_GAMMA [UNDEFINED]	DP	Acceleration amplitude of the bottom wall
Output and Restart Control		
PRINT_DES_DATA [F]	L	Option to print DEM output
P_TIME [UNDEFINED_I]	DP	Time interval to print DEM output when doing only granular flow simulation

4.16 ISAT and Direct Integration for Chemical Reactions

Input Variables:

CALL_DI [F]	L	Variable to decide if chemical reactions are time-split and solved using direct integration (DI) with ODE solver. Do not call time-splitting and DI
CALL_ISAT [F]	L	Variable to decide if chemical reactions are time-split and solved using ISAT. Do not use time-splitting and ISAT.
CALL_GROW [F]	L	Variable to decide if particle growth due to chemical reactions is calculated. Do not do particle growth calculations.
ISATdt	DP	Time step for ISAT simulations (usually the value is less than the average time step in MFIX).

If "CALL_DI=.TRUE." or "CALL_ISAT=.TRUE.", MFIX uses DI or ISAT to calculate the chemical reactions using the time-splitting method. The ODE solver used here is ODEPACK (www.netlib.org/odepack).

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

If "CALL_ISAT=.TRUE.," user must have the ISATAB library and link it in the make_mfix file by specifying '-l{path to the library}/libraryname' at the end of the link options. Figure 7 of the ISAT manual in the document directory shows the flow of MFIX using ISAT. The user must provide ISATdt to keep the high performance of ISAT.

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

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

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

mchem_mod.f	Sets the global variables.
misat_table_init.f	Sets the controlling parameters for ISATAB.
mchem_odepack_init.f	Sets the controlling parameters for ODEPACK.
fex.f	The subroutine provides the source terms for equations in the ODE solver and source terms of reactions rates. The source terms have the following order: $[\rho_g, T_g, X_g], [\varepsilon_{s1}, T_{s1}, X_{s1}, d_1], \cdots, [\varepsilon_{sm}, T_{sm}, X_{sm}, d_m]$.

	If the variables are constant during the simulations, the source terms should be set to zero. For example, for isothermal chemical reactions, the source terms of T_g and T_{sm} are zero.
calc_jacobian.f	Provides the Jacobian matrix for ODE solver.
transport_prop.f	Provides transport properties.
physical_prop.f	Provides physical properties.

Example using ISAT

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

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

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

$$SiH_2 + SiH_4 \leftrightarrow Si_2H_6.$$

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

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

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

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

The following files are modified to simulate this case:

mchem_mod.f	Sets the global variables.
misat_table_init.f	Controlling parameters for ISATAB are set.
mchem_odepack_init.f	Controlling parameters for ODEPACK are set.
fex.f	The subroutine provides the source terms to ODE solver and sources terms of reactions. For this case, the order of ODEs is $[\rho_g, T_g, X_{SiH4}, X_{SiH2}, X_{H2}, X_{Si2H6}, X_{N2}], [\varepsilon_{s1}, T_{s1}, X_{Si}, X_{Al2O3}, d_1]$. As written in the code, the user should provide the source terms of reactions, which is RXN_source_g(*n*) for gas phase species *n* and RXN_source_s(*m,n*) for solid phase *m* and species *n*. Then the source terms of the ODEs are calculated. Note that N_2 and Al_2O_3 are inert species; therefore the source terms are set to zero.

calc_jacobian.f	The subroutine provides the Jacobian matrix for ODE solver. The example case uses the Jacobian matrix generated by ADIFOR (g_derives.f). The ADIFOR can be downloaded from the website (http://www-unix.mcs.anl.gov/autodiff/ADIFOR).
transport_prop.f	Provides the transport properties.
physical_prop.f	Provides the physical properties such as averaged molecular weight.

4.17 Direct Quadrature Method of Moments (DQMOM)

Input Variables:

CALL_DQMOM [F]	L	Variable to decide if the population balance equations are solved. Do not invoke DQMOM
Nscalar [0]	I	Number of solid phases to solve the population balance equations.
Aggregation_eff [0.0]	DP	Success-factor for aggregation.
Breakage_eff [0.0]	DP	Success-factor for breakage.

The new files added to MFIX are as follows:

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

Example using DQMOM

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

4.18 Cohesion Model in DEM

Cohesive interparticle forces represent an addition to the discrete-particle simulation. The raw code for this addition was located in the *mfix/model/cohesion* directory. These forces can be implemented using both a square-well model and a Hamaker van der Waals model. Cohesive forces are turned-on by setting the logical variable "USE_COHESION" equal to true in the mfix.dat file.

Square-Well model

The nature of the square-well model is determined by the well width (r_{outer}) and the well depth (D). Both of these parameters are specified in the *mfix.dat* file. Furthermore, the simulation is set up to use a separate set of square-well parameters with particle-particle interactions and particle-wall interactions. Generally, the well depth is set to be twice as large for particle-wall interactions to be consistent with the increased surface contact that would exist in such interactions.

Within the square well model, cohesive interactions are implemented as instantaneous momentum impulses (J_*) resulting in a change in particle velocity as shown below:

$$m\vec{\boldsymbol{v}}_{i,post} = m\vec{\boldsymbol{v}}_{i,pre} - J_*\vec{\boldsymbol{k}}_{ij,*} \qquad (1)$$

where m is the particle mass, $\vec{v}_{i,pre}$ is the pre-interaction velocity, $\vec{v}_{i,post}$ is the post-interaction velocity and $\vec{k}_{ij,*}$ is the unit vector connecting the particle centers. This treatment leads to three types of cohesive interactions in the square-well model: approaching cohesive interactions, escaping cohesive interactions and capture cohesive interactions. The momentum impulse during each of these interactions can be calculated by solving the momentum balance along with the stipulation that the total kinetic energy of the two-particle system changes by the depth of the square well (D). This treatment leads to the following expressions for the momentum impulse in each type of interaction:

$$J_{coh,app} = \frac{m}{2}\left(\vec{k}_{ij,1}\cdot\vec{v}_{ij,a} - \sqrt{\frac{4D}{m} + \left(\vec{k}_{ij,1}\cdot\vec{v}_{ij,a}\right)^2}\right) \qquad (2)$$

$$J_{coh,esc} = \frac{m}{2}\left(\vec{\boldsymbol{k}}_{ij,3}\cdot\vec{\boldsymbol{v}}_{ij,c} + \sqrt{\left(\vec{\boldsymbol{k}}_{ij,3}\cdot\vec{\boldsymbol{v}}_{ij,c}\right)^2 - \frac{4D}{m}}\right) \qquad (3)$$

$$J_{coh,cap} = m\left(\vec{k}_{ij,3}\cdot\vec{v}_{ij,c}\right) \qquad (4)$$

Because cohesive interactions are implemented only once on approach and once as the particles depart, the identification of a square-well interaction cannot be done simply using the separation distance. Special book-keeping is employed to ensure that square-well cohesive interactions are implemented only during the time step when the well-widths first become overlapped for approaching interactions and when they first become un-overlapped for departing interactions. This book-keeping involves recording a "list of links" for each particle. Two particles are considered "linked" is their separation distance is less than the width of their square-wells (the square-wells overlap). The list of links is updated at every cohesive interaction. Cohesive square-well interactions are recognized using both the list of links and the particle separation distance at each. For example, an approaching cohesive interaction is only implemented if two particles overlap their square-wells, yet they are not on their respective list of links. The particles will be added to each list during the interaction and at the next time step, they may still have overlapped square-wells, but an interaction will not be implemented because they particles are already on their respective list of links. A similar algorithm is used to identify departing interactions.

Hamaker model

The Hamaker model for van der Waals forces predicts the cohesive force between equal-sized, spherical particles according to the following equation [1]:

$$F_{vdW} = \frac{Ar_{inner}}{12H^2} \quad (5)$$

where r_{inner} is the particle radius; A is the Hamaker constant, which is specific to a given material and has typical values on the order of 10^{-20} J; and H is the minimum surface-to-surface separation distance between two particles i and j:

$$H = \sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2} - 2r_{inner} \quad (6)$$

The corresponding cohesive force between a spherical particle and a flat wall is [1]:

$$F_{vdW,wall} = \frac{A_{wall} r_{inner}}{6H^2} \quad (7)$$

For both of these expressions, the cohesive force approaches infinity as the separation distance approaches zero. This singularity incurred at particle contact is avoided by introducing a "cutoff" distance, H_{cut}. For separation distances below this cutoff distance, the interparticle cohesive force is given by a surface adhesion force (F_{ad}) model [2].

$$F_{ad} = 2\pi r_{inner}\gamma \quad (8)$$

where γ is a constant surface energy per unit area. The contact surface energy is calculated at the beginning of each simulation based on the specified Hamaker constant and cutoff distance to ensure that the force is continuous at the cutoff distance. Thus, the cohesive force is maintained at a constant value for any separation distances (based on equation 6) below the minimum cutoff distance, which includes any "negative" separation distances that occur during actual particle contact.

Cohesive Interaction Search

Cohesive interactions are identified using a method separate from the functions used to identify particle contacts in the discrete-particle model. The simulation domain is divided up into search grids in order to reduce the number of particles that must be checked for interactions by each particle. The search grid is made up of several uniform boxes that are defined to be at least as wide as the largest length-scale in the cohesive forces. In this way, particles are only checked against other particles in their box and adjacent boxes. The size of the search boxes ensures that a particle could not have an interaction with particles that are located in boxes outside the adjacent boxes. The arrays recording the indices of the search box for each particle are updated at each time step (*update_search_grids.f*).

Cohesion Variables

VARIABLE	TYPE	DESCRIPTION
WELL_WIDTH(NPARTICLES)	DP	Square well depth associated with each particle
WELL_DEPTH(NPARTICLES)	DP	Square well width associated with each particle
MASTER_WELL_DEPTH	DP	Square-well depth read in from mfix.dat for simulations that use same well depth for all particle-particle interactions
MASTER_WALL_WELL_DEPTH	DP	Square-well depth for particle-wall interactions read in from mfix.dat for simulations that use same well width for all particle-wall interactions
RADIUS_RATIO	DP	Ratio of square-well width to particle radius
WALL_RADIUS_RATIO	DP	Ratio of wall square-well depth to particle radius
LINKS(MAXNEIGHBORS, NPARTICLES)	I	
USE_COHESION	L	Switch to turn cohesion on and off
SQUARE_WELL	L	Switch to turn square well on and off
IS_LINKED(NPARTICLES)	L	True if given particle has at lest one linked partner

COHESION_DEBUG	I	Flag to turn on output lines for debugging cohesive simulation
SEARCH_GRIDS(3)	I	Number of search grids in the x-, y-, and z-direction
PART_IN_GRID(100, 100, 100, 50)	I	Matrix of particles in each search grid
PART_GRID(4,NPARTICLES)	I	Matrix location of particle
MAX_PART_IN_GRID	I	Maximum number of particles in a search grid
SEARCH_GRID_SIZE(3)	DP	width of search grids
VAN_DER_WAALS	L	Flag to turn on the use Hamaker van der Waals forces
HAMAKER_CONSTANT	DP	Hamaker constant used for particle-particle interactions
VDW_INNER_CUTOFF	DP	Minimum separation distance below which van der Waals forces are calculated using a surface adhesion model
VDW_OUTER_CUTOFF	DP	Maximum separation distance above which van der Waals forces are not implemented
WALL_HAMAKER_CONSTANT	DP	Hamaker constant used in particle-wall interactions
WALL_VDW_INNER_CUTOFF	DP	Minimum separation distance below which van der Waals forces are calculated using a surface adhesion model (particle-wall interactions)
WALL_VDW_OUTER_CUTOFF	DP	Maximum separation distance above

		which van der Waals forces are not implemented (particle-wall interactions)
SURFACE_ENERGY	DP	Surface energy used to calculate cohesive force for low separation distances in van der Waals model (this variable is calculated at the beginning of each simulation to ensure the van der Waals force is continuous at the inner cutoff)
WALL_SURFACE_ENERGY	DP	Surface energy used to calculate cohesive force for low separation distances in van der Waals model (particle-wall interactions) (this variable is calculated at the beginning of each simulation to ensure the van der Waals force is continuous at the inner cutoff)