LABORATORY NOTEBOOK

LABORATORY NOTEBOOK

NOTEBOOK NO. 821

CONTINUED FROM NOTEBOOK NO. CONTINUED TO NOTEBOOK NO.

ASSIGNED TO:

NAME Debashis Basu

SIGNATURE Debashis Basu DATE

DATE ISSUED BY

PHONE EMAIL

COMPANY

DEPARTMENT

ADDRESS

CITY STATE ZIP

DATE NOTEBOOK COMPLETED NUMBER OF PAGES FILLED IN

NOTES:

CNWRA
CONTROLLED
COPY 821

LABORATORY NOTEBOOK GUIDELINES†

LABORATORY NOTEBOOK INTRODUCTION

Using a Laboratory Notebook to record ideas, inventions, experimentation records, observations and all work details is a vital part of any laboratory process. Careful attention to how you keep your Laboratory Notebook can have a positive impact on the patent outcome of a pending discovery or invention.

Following are some overall recommendations to help you keep more efficient and accurate Laboratory Notebook entries. Remember, however, that these are simply a suggested set of guidelines. Only your attorney can supply the exact guidelines she would like you to follow to satisfy specific legal requirements. That is why we recommend that you consult your legal counsel.

RECORDING DATA

Your Laboratory Notebook is a vital record of your work whether it is for patent purposes, legal records or documenting drug research under FDA guidelines. The Laboratory Notebook can help you prove:

a. Exact details and dates of conception
b. Details and dates of reduction to practice
c. Diligence in reducing your invention to practice
d. Details regarding the structure and operation of your invention
e. Experimentation observations and results
f. A chronological record of your work
g. Other work details

Follow a few simple rules of thumb

1. Always record entries legibly, neatly and in permanent ink.
2. Immediately enter into your notebook and date all original concepts, data and observations, using separate headings to differentiate each.
3. Record all concepts, results, references and other information in a systematic and orderly manner. (Language, charts and numbering systems should be maintained consistently throughout.)
4. It is acceptable to make your entries brief. Always, however, include enough details for someone else to successfully duplicate the work you have recorded.
5. Label all figures and calculations.
6. Never, under any circumstances, remove pages from your notebook.

Remember to treat your Laboratory Notebook as a legal document: It records the chronological history of your activities. The following guidelines should help you maintain the consistent and accurate entries needed for future legal purposes.

1. Start entries at the top of the first page, and always make successive, dated entries, working your way to the bottom of the last page.
2. After completing a page, sign it before continuing to the next page.
3. Make sure that you record the date of each entry clearly and unambiguously.
4. Never let anyone other than yourself write in your Notebook (excluding witness signatures, discussed later).
5. Never leave blank spaces, and never erase or remove material you have added. Simply draw lines through any blank spaces at the same time you are making your entries.
6. Do not erase errors. Just draw a single line through any erroneous entry, then add your initials. Enter the correct entry nearby.
7. You can supplement your entries with supporting material (e.g., test-result printouts and other documentation). But you must permanently affix the material onto a page in its proper chronological location.
8. Never rely solely on any supplemental attachment. Always include your own entry describing the attachment and add any conclusions that you might draw from its substance.
9. Occasionally, secondary sources might be too large or inappropriate to attach directly to your notebook. In this case, you can add all secondary sources to an ancillary record maintained precisely for this purpose. However, always remember to write a description of these secondary sources, clearly and unambiguously, in your notebook.

DOCUMENTING PATENT ACTIVITIES

A primary purpose of a Laboratory Notebook is the support of documenting work that may be patentable. To support patent activities, it is necessary to provide clear, concise, chronological entries with specific dates. To rely on these dates, you must have at least one non-inventor corroborate that the events actually happened and that he or she understood your invention by signing and dating the "Disclosed to and Understood by" signature blocks.

Your Laboratory Notebook should help you document and prove:

1. *Conception Date*—The date that you knew your invention would solve the problem.
2. *Date of reduction to practice*—The moment that you made a working embodiment of your invention.
3. *Diligence in reducing your invention to practice*—Diligence refers to your intent and conscious effort to make a working embodiment. You are not required to rush, or even to take the most efficient development strategy. But your Notebook must include details relating to your diligent activities. These are dates and facts that show what activities you have conducted to reduce the invention to practice, and when such activities were conducted. Since you may still be diligent despite periods of not working on reducing your invention to practice, always remember to provide reasonable excuses for these periods of inactivity by supplying facts relating to why there was no activity during the period in question. (e.g., unavailability of test conditions or equipment).
4. *How to make and use your invention*—provide documentation details sufficient to teach a colleague how to make and use your invention.
5. *The best mode of practicing your invention*—document the best way to practice your invention.

A non-inventor colleague should corroborate each of these events/facts by signing the "Disclosed to and Understood by" on the relevant pages.

† BookFactory provides these sample guidelines "AS IS" without any warranty.

TABLE OF CONTENTS

BOOK NO.

PAGE	SUBJECT	DATE
65		
66		
67		
68		
69		
70		
71		
72		
73		
74		
75		
76		
77		
78		
79		
80		
81		
82		
83		
84		
85		
86		
87		
88		
89		
90		
91		
92		

Purpose:

The purpose of this notebook is to document the different subroutines and modules of the codes "MFIX". In addition, the formulation of the code, the equations that are being used, and all relevant information are documented. All the relevant references are also documented.

Tasks :

1. Explanation of the different modules of the MFIX code. Simple description of all the subroutines is provided.
2. MFIX governing equations are closure models are provided from relevant technical publications.
3. Summary of the Governing Equations and Constitutive Relations are also provided.
4. All relevant references are also provided in the scientific notebook.

The documentation of the subroutines and modules were based on the following technical papers available in open literature (technical journals and reports). These papers and reports have the detailed formulation of the MFIX code.

1. Chandrasekharan, B. K., Lee, Loni van der, Hulme, Ian., & Kantzas, A., "A simulation and Experimental Study of the Hydrodynamics of a Bubbling Fluidized Bed of Linear Low Density Polyethylene Using Bubble Properties and Pressure Fluctuations", Macromolecular Materials and Engineering, Vol. 290, pp. 592-609, 2005.

2. Dartevelle, S., "Numerical modeling of geophysical granular flows: 1. A comprehensive approach to granular rheologies and geophysical multiphase flows" GEOCHEMISTRY GEOPHYSICS GEOSYSTEMS, VOL. 5, Q08003, doi:10.1029/2003GC000636, 2004.

3. Dartevelle, S., "Numerical modeling of geophysical granular flows: 2. Computer simulations of plinian clouds and pyroclastic flows and surges", GEOCHEMISTRY GEOPHYSICS GEOSYSTEMS, VOL. 5, Q08004, doi:10.1029/2003GC000637, 2004.

4. Syamlal, M., Rogers, W., and O'Brien, T. J., "MFIX documentation: Theory guide, DOE/METC-94/1004, DE9400,097,49 pp., U. S. Department of Energy, Washington, D. C.

Magma Modeling :-

- Using FLOW3D.

dbasu/texas/FLOW3D-run

Mixture Two-Phase.

- Trying the two-phase (drift-flux model).

* Just trying a 2-phase drift flux model.

* Drift-Flux Model can be used for everything whenever there is a two-phase flow involved.

* Some basic Computations

D.B 8/13/06

Using FLOW-3D & Drift ~~Flux~~ Flux Model

- To see the capability of FLOW-3D.

→ Let's focus on MFIX also.

The rest of the scientific notebook focuses only on the Subroutines/modules/formulations of the MFIX code.

MFIX MODULES (WHAT EACH MODULE DOES)

These codes are mainly for specifying machine constants, CPU time etc. → Mainly Related to Hardware.

```
-rw-r--r-- 1 dbasu sunuser 10192 Jul 27 12:50 AIX.F
-rw-r--r-- 1 dbasu sunuser 53488 Jul 27 12:50 BLAS.F
-rw-r--r-- 1 dbasu sunuser 9571 Jul 27 12:50 CM.F
-rw-r--r-- 1 dbasu sunuser 11342 Jul 27 12:50 IRIXF90.F
-rw-r--r-- 1 dbasu sunuser 9926 Jul 27 12:50 LINUX.F
-rw-r--r-- 1 dbasu sunuser 10060 Jul 27 12:50 LINUX_intel.F
-rw-r--r-- 1 dbasu sunuser 1217219 Jul 27 12:50 ODEPACK.F
-rw-r--r-- 1 dbasu sunuser 9996 Jul 27 12:50 OSF1.F
-rw-r--r-- 1 dbasu sunuser 10163 Jul 27 12:50 PC.F
-rw-r--r-- 1 dbasu sunuser 10048 Jul 27 12:50 PGNT.F
-rw-r--r-- 1 dbasu sunuser 9591 Jul 27 12:50 ULTRIX.F
```

- AIX.F → Set the machine constants.
- BLAS.F → Constant times a vector plus a vector.
- CM.F → Set the machine [redacted] constants. D.B 8/23/06
- IRIXF90.F → Set the machine constants.
- LINUX.F → Set the machine constants.
- LINUX_intel.F → Set the machine constants. (Modify to make it work for Absoft 7.5 Profortran compiler)
- ODEPACK.F → Livermore solver for ordinary Differential Equations. It solves the initial-value problem for stiff or non-stiff systems.
- OSF1.F → Program for setting the machine const
- PC.F → Set the machine constants.
- PGNT.F → Set the machine constants.
- ULTRIX.F → Set the machine contants.

[Adjusting the zeros in diff-momentum equations]

```
-rw-r--r-- 1 dbasu sunuser 4244 Jul 27 12:50 adjust_a_u_g.f
-rw-r--r-- 1 dbasu sunuser 4415 Jul 27 12:50 adjust_a_u_s.f
-rw-r--r-- 1 dbasu sunuser 4065 Jul 27 12:50 adjust_a_v_g.f
-rw-r--r-- 1 dbasu sunuser 4304 Jul 27 12:50 adjust_a_v_s.f
-rw-r--r-- 1 dbasu sunuser 4097 Jul 27 12:50 adjust_a_w_g.f
-rw-r--r-- 1 dbasu sunuser 4285 Jul 27 12:50 adjust_a_w_s.f
```

- adjust_a_u_g.f → Handles the special case of the center coeff. in U-g momentum eq. becoming zero.
- adjust_a_u_s.f → Handles the special case of the center coeff. in U_s momentum eq. becoming zero.
- adjust_a_v_g.f → Handles the special case of the center coeff. in V-g momentum eq. becoming zero.
- adjust_a_v_s.f → Handles the special case of the center coeff. in V_s momentum eq. becoming zero.
- adjust_a_w_g.f → Handles the special case of the centers coeff. in W-g momentum eq. becoming zero.
- adjust_a_w_s.f → Handles the special case of the center coeff. in W_s momentum eq. becoming zero.

[Adjusting different parameters]

```
-rw-r--r-- 1 dbasu sunuser 6754 Jul 27 12:50 adjust_dt.f
-rw-r--r-- 1 dbasu sunuser 4869 Jul 27 12:50 adjust_eps.f
-rw-r--r-- 1 dbasu sunuser 2808 Jul 27 12:50 adjust_leq.f
-rw-r--r-- 1 dbasu sunuser 2631 Jul 27 12:50 adjust_rop.f
```

- adjust_dt.f → Automatically adjust the time step.
- adjust_eps.f → Eliminate the solid phases that occupy very small fractions of the computational cell volume.
- adjust_leq.f → Adjusts Linear Equation solver method and iterations.
- adjust_rop.f → Removes small negative values of density.

Debashis Basu | Debashis Basu | 8/23/2006

INVENTOR SIGNATURE | PRINTED NAME | DATE

DISCLOSED TO AND UNDERSTOOD BY | PRINTED NAME | DATE

(Boundary Conditions)

```
-rw-r--r-- 1 dbasu sunuser 2236 Jul 27 12:50 adjust_theta.f
-rw-r--r-- 1 dbasu sunuser 14664 Jul 27 12:50 allocate_arrays.f
-rw-r--r-- 1 dbasu sunuser 1199 Jul 27 12:50 ambm_mod.f
-rw-r--r-- 1 dbasu sunuser 109 Jul 27 12:50 b_force1.inc
-rw-r--r-- 1 dbasu sunuser 1937 Jul 27 12:50 b_force2.inc
-rw-r--r-- 1 dbasu sunuser 25859 Jul 27 12:50 b_m_p_star.f
-rw-r--r-- 1 dbasu sunuser 14227 Jul 27 12:50 bc_mod.f
-rw-r--r-- 1 dbasu sunuser 16255 Jul 27 12:50 bc_phi.f
-rw-r--r-- 1 dbasu sunuser 31609 Jul 27 12:50 bc_theta.f
-rw-r--r-- 1 dbasu sunuser 2348 Jul 27 12:50 bound_x.f
-rw-r--r-- 1 dbasu sunuser 1047 Jul 27 12:50 boundfunijk3_mod.f
-rw-r--r-- 1 dbasu sunuser 1040 Jul 27 12:50 boundfunijk_mod.f
-rw-r--r-- 1 dbasu sunuser 2975 Jul 27 12:50 cal_d.f
-rw-r--r-- 1 dbasu sunuser 5333 Jul 27 12:50 calc_cell.f
-rw-r--r-- 1 dbasu sunuser 11075 Jul 27 12:50 calc_coeff.f
-rw-r--r-- 1 dbasu sunuser 45547 Jul 27 12:50 calc_d.f
-rw-r--r-- 1 dbasu sunuser 3054 Jul 27 12:50 calc_dif_g.f
-rw-r--r-- 1 dbasu sunuser 3162 Jul 27 12:50 calc_dif_s.f
-rw-r--r-- 1 dbasu sunuser 3905 Jul 27 12:50 calc_drag.f
-rw-r--r-- 1 dbasu sunuser 9511 Jul 27 12:50 calc_e.f
-rw-r--r-- 1 dbasu sunuser 7620 Jul 27 12:50 calc_gama.f
-rw-r--r-- 1 dbasu sunuser 51826 Jul 27 12:50 calc_grbdry.f
-rw-r--r-- 1 dbasu sunuser 10909 Jul 27 12:50 calc_k_cp.f
```

- adjust_theta.f → Remove small negative values of theta caused by linear solvers.
- allocate_arrays.f → Allocating the arrays.
- ambm_mod.f → Module ambm
- b_force1.inc → Double Precision declaration (Body force)
- b_force2.inc → Include file for all body force statement functions.
- b_m_p_star.f → Determine P_star when there is an interface at east.
- bc_mod.f → Common block containing Boundary Conditions Data.
- bc_phi.f → Set up the phi boundary conditions.
- bc_theta.f → Implementation of Johnson and Jackson Boundary conditions for the Pseudo thermal temperature.
- bound_x.f → Bound the value of x array.
- boundfunijk3_mod.f → Integer function BOUND_FUNIJK3
- boundfunijk_mod.f → Integer function BOUND_FUNIJK
- cal_d.f → Calculates shear velocity. D.B 8/23/06
- calc_cell.f → Calculate the cell index for a reactor location
- calc_coeff.f → Calculates all physical and transport properties, reaction rates & exchange rates.
- calc_d.f → Calculate coefficients linking velocity correction to pressure correction - east.
- calc_dif_g.f → Calculates the effective diffusivity of fluid phase.
- calc_dif_s.f → Calculates the effective diffusivity of solid phases.
- calc_drag.f → Calculates gas-solids & solids-solids drag terms.
- calc_e.f → Calculate coefficients linking velocity correction to pressure correction. → SIMPLE (???)
- calc_gama.f → Calculates gas-solid heat transfer coefficient.
- calc_grbdry.f → Calculates hw & cw for kinetic theory boundary conditions.
- calc_k_cp.f → Calculate & store dPodEP_S.

(Very important. Calculates Important Parameters.)

ALL property & other stuff calculations

Debashis Basu | DEBASHIS BASU | 8/23/06

INVENTOR SIGNATURE | PRINTED NAME | DATE

DISCLOSED TO AND UNDERSTOOD BY | PRINTED NAME | DATE

```
-rw-r--r--  1 dbasu  sunuser  3555 Jul 27 12:50 calc_k_g.f
-rw-r--r--  1 dbasu  sunuser  5549 Jul 27 12:50 calc_k_s.f
-rw-r--r--  1 dbasu  sunuser  6413 Jul 27 12:50 calc_mflux.f
-rw-r--r--  1 dbasu  sunuser  12589 Jul 27 12:50 calc_mu_g.f
-rw-r--r--  1 dbasu  sunuser  47993 Jul 27 12:50 calc_mu_s.f
-rw-r--r--  1 dbasu  sunuser  2996 Jul 27 12:50 calc_mw.f
-rw-r--r--  1 dbasu  sunuser  6694 Jul 27 12:50 calc_outflow.f
-rw-r--r--  1 dbasu  sunuser  12239 Jul 27 12:50 calc_p_star.f
-rw-r--r--  1 dbasu  sunuser  46779 Jul 27 12:50 calc_resid.f
-rw-r--r--  1 dbasu  sunuser  45568 Jul 27 12:50 calc_s_ddot_s.f
-rw-r--r--  1 dbasu  sunuser  3164 Jul 27 12:50 calc_trd_g.f
-rw-r--r--  1 dbasu  sunuser  3247 Jul 27 12:50 calc_trd_s.f
-rw-r--r--  1 dbasu  sunuser  58211 Jul 27 12:50 calc_u_friction.f
-rw-r--r--  1 dbasu  sunuser  5084 Jul 27 12:50 calc_vol_fr.f
-rw-r--r--  1 dbasu  sunuser  35800 Jul 27 12:50 calc_xsi.f
-rw-r--r--  1 dbasu  sunuser  6594 Jul 27 12:50 check_ab_m.f
-rw-r--r--  1 dbasu  sunuser  7018 Jul 27 12:50 check_convergence.f
-rw-r--r--  1 dbasu  sunuser  6550 Jul 27 12:50 check_data_01.f
-rw-r--r--  1 dbasu  sunuser  3962 Jul 27 12:50 check_data_02.f
-rw-r--r--  1 dbasu  sunuser  10651 Jul 27 12:50 check_data_03.f
-rw-r--r--  1 dbasu  sunuser  8324 Jul 27 12:50 check_data_04.f
-rw-r--r--  1 dbasu  sunuser  7516 Jul 27 12:50 check_data_05.f
-rw-r--r--  1 dbasu  sunuser  26453 Jul 27 12:50 check_data_06.f
```

- calc_k_g.f → Calculate the effective conductivity of fluid phase.
- calc_k_s.f → Calculate the effective conductivity of solid phase.
- calc_mflux.f → Calculates the convection fluxes. (Master Routine)
- calc_mu_g.f → Calculate the effective viscosity for a turbulent flow, sum of molecular & eddy viscosities. (V.V. Sinh)
- calc_mu_s.f → Calculate the granular stress tensor. (V.V. Sinh)
- calc_mw.f → Calculate the average molecular weight of gas.
- calc_outflow.f → Calculate the mass and volumetric outflow rates from an outflow boundary.
- calc_p_star.f → Calculate p_star in cells where solids continuity is solved.
- calc_resid.f → Calculates residuals for continuity equations.
- calc_s_ddot_s.f → Calculate del.u, S:S and S_XX, S_YY or S_ZZ at the boundary for use in frictional boundary condition. D.B. 8/22/06
- calc_trd_g.f → Calculate the trace of gas phase rate of strain tensor.
- calc_trd_s.f → Calculate the trace of solids phase rate of strain tensor.
- calc_u_friction.f → Calculate Rho and eco for kinetic theory and frictional boundary conditions.
- calc_vol_fr.f → Calculate volume fractions of phases used for pressure corrections.
- calc_xsi.f → Calculate the convection weighting (determining) factors for higher order discretization.
- check_ab_m.f → Check the entries of the sparse matrix coefficients and the source vector, if sre is set true.
- check_convergence.f → Monitor Convergence.
- check_data_01.f → Check the run control namelist section
- check_data_02.f → Check the output control namelist section
- check_data_03.f → Check the geometry and discretization namelist section. [Grid Generation]
- check_data_04.f → Check the solid phase input section.
- check_data_05.f → Check the gas phase input section. D.B. 8/22/06
- check_data_06.f → Check the initial conditions input section.

Calculations related to different thing.

Checking Different - Factors -

Debashis Basu. | DEBASHIS BASU | 8/23/2006
INVENTOR SIGNATURE | PRINTED NAME | DATE

DISCLOSED TO AND UNDERSTOOD BY | PRINTED NAME | DATE


```
-rw-r--r--  1 dbasu  sunuser  41992 Jul 27 12:50 check_data_07.f
-rw-r--r--  1 dbasu  sunuser  9069 Jul 27 12:50 check_data_08.f
-rw-r--r--  1 dbasu  sunuser  7114 Jul 27 12:50 check_data_09.f
-rw-r--r--  1 dbasu  sunuser  18364 Jul 27 12:50 check_data_20.f
-rw-r--r--  1 dbasu  sunuser  31910 Jul 27 12:50 check_data_30.f
-rw-r--r--  1 dbasu  sunuser  35917 Jul 27 12:50 check_mass_balance.f
-rw-r--r--  1 dbasu  sunuser  932 Jul 27 12:50 check_mod.f
-rw-r--r--  1 dbasu  sunuser  8801 Jul 27 12:50 check_one_axis.f
-rw-r--r--  1 dbasu  sunuser  3180 Jul 27 12:50 check_plane.f
drwxr-xr-x  4 dbasu  sunuser  512 Jul 27 12:50 chem
-rw-r--r--  1 dbasu  sunuser  3938 Jul 27 12:50 chischeme_mod.f
-rw-r--r--  1 dbasu  sunuser  6673 Jul 27 12:50 cn_extrapol.f
-rw-r--r--  1 dbasu  sunuser  651 Jul 27 12:50 coeff_mod.f
drwxr-xr-x  2 dbasu  sunuser  1024 Jul 27 12:50 cohesion
-rw-r--r--  1 dbasu  sunuser  4428 Jul 27 12:50 compare.f
-rw-r--r--  1 dbasu  sunuser  5920 Jul 27 12:50 constant_mod.f
-rw-r--r--  1 dbasu  sunuser  249 Jul 27 12:50 cont_mod.f
-rw-r--r--  1 dbasu  sunuser  35590 Jul 27 12:50 conv_dif_phi.f
-rw-r--r--  1 dbasu  sunuser  31569 Jul 27 12:50 conv_dif_u_g.f
-rw-r--r--  1 dbasu  sunuser  32183 Jul 27 12:50 conv_dif_u_s.f
-rw-r--r--  1 dbasu  sunuser  31414 Jul 27 12:50 conv_dif_v_g.f
-rw-r--r--  1 dbasu  sunuser  32142 Jul 27 12:50 conv_dif_v_s.f
-rw-r--r--  1 dbasu  sunuser  30816 Jul 27 12:50 conv_dif_w_g.f
```

- check_data_07.f → Check Boundary conditions, specifications, convert physical locations to i,j,k's, compute BC-AREA.
- check_data_08.f → Check internal surface specifications, and convert physical locations to i,j,k's.
- check_data_09.f → Check chemical reactions specifications.
- check_data_20.f → Check whether the sum of volume fractions is 1.0 and Eg >= E*. (Volume fraction cheque.)
- check_data_30.f → Check whether the sum of reaction rates is zero and the sum of mass fraction is 1.0 Eg >= E*.
- check_mass_balance.f → Check Global Species and elemental balance.
- check_mod.f → MODULE CHECK.
- check_one_axis.f → Check Geometry data for one axis.
- check_plane.f → Make sure the flow boundary condition of internal surface is a plane.
- chem → Directory for chemical composition/chemistry etc. (Directory.)
- chischeme_mod.f → Determine factors for Chi-Scheme of Darwish and Moukalled to ensure consistency of equation sets.
- cn_extrapol.f → Performs the explicit extrapolation phase when Crank-Nicholson based second order accurate time integration is activated. (Related to time integration.)
- coeff_mod.f → MODULE COEFF.
- cohesion → DIRECTORY FOR PARTICLE COHESION. (Directory.)
- compare.f → Common block containing physical constants and constants used in the numerical technique.
- constant_mod.f → Logical function compare. [Returns .TRUE if values V1, V2 are equal].
- cont_mod.f → Indicates whether the continuity equation needs to be solved.
- conv_dif_phi.f → Determine convection diffusion terms for a scalar ϕ. The off-diagonal coefficients calculated here must be positive. The center coeff. and the source vector are negative. (Important Subroutine.)
- conv_dif_u_g.f → Determine the convection diffusion terms for U-s momentum equation. (Important Subroutine.)
- conv_dif_u_s.f → Determine the convection diffusion terms for U-s momentum equations.
- conv_dif_v_g.f → Determine the convection diffusion
- conv_dif_v_s.f → Determine the convection diffusion terms for v-s momentum equations.
- conv_dif_w_g.f → Determine the convection diffusion terms for w-g momentum equations.

Very Important Subroutines.
(Determines the convection diffusion terms for different momentum equations).

Debashis Basu | DEBASHIS BASU | 8/23/2006
INVENTOR SIGNATURE | PRINTED NAME | DATE

DISCLOSED TO AND UNDERSTOOD BY | PRINTED NAME | DATE

Imp ← -rw-r--r-- 1 dbasu sunuser 31306 Jul 27 12:50 conv_dif_w_s.f → Determine the convection diffusion terms for W-s momentum equations.

Imp ← -rw-r--r-- 1 dbasu sunuser 6454 Jul 27 12:50 conv_pp_g.f → Determine the convection terms for pressure correction equation. (Cell explaine (in a big way) in the top.

Imp. Module subroutine. ← -rw-r--r-- 1 dbasu sunuser 12390 Jul 27 12:50 conv_rop.f → Calculate the face value of density used for calculating convective fluxes. (Module subroutine)

Determining the convective terms for different continuity equations. ←
-rw-r--r-- 1 dbasu sunuser 11698 Jul 27 12:50 conv_rop_g.f → Determine convection terms for a continuity equation.
-rw-r--r-- 1 dbasu sunuser 12289 Jul 27 12:50 conv_rop_s.f → Determine convection terms for solids continuity equation.
-rw-r--r-- 1 dbasu sunuser 23995 Jul 27 12:50 conv_source_epp.f → Determine convection terms for solids volume fraction correction equation.

-rw-r--r-- 1 dbasu sunuser 5832 Jul 27 12:50 copy_a.f → Copy A-vel-g to A-m.

Related to identifying cell & neighbour. ✓ ← -rw-r--r-- 1 dbasu sunuser 10485 Jul 27 12:50 corner.f → Identify wall cells with more than one fluid cell as a neighbour. No heat mass, momentum or energy transfer.

-rw-r--r-- 1 dbasu sunuser 331 Jul 27 12:50 corner_mod.f → Module for corner.

Important for the gas - solid phases & velocities. ←
-rw-r--r-- 1 dbasu sunuser 10002 Jul 27 12:50 correct_0.f → Correct the fluid pressure and gas and solids velocities.
-rw-r--r-- 1 dbasu sunuser 4422 Jul 27 12:50 correct_1.f → Correct the solids volume fraction.

Calculates the cp values of all materials & components. ←
-rw-r--r-- 1 dbasu sunuser 1962 Jul 27 12:50 cp_fun1.inc → This is an include file containing all CP functions.
-rw-r--r-- 1 dbasu sunuser 3089 Jul 27 12:50 cp_fun2.inc → This is an include file containing all CP functions.

Discrete element simulation related files. ←
drwxr-xr-x 2 dbasu sunuser 1536 Jul 27 12:50 des → Directory for "DISCRETE ELEMENT SIMULATION".
-rw-r--r-- 1 dbasu sunuser 3503 Jul 27 12:50 dgtsl.f → Given a tri-diagonal matrix and a right hand side, will find a solution

Related to diffusive fluxes across internal surfaces. ←
-rw-r--r-- 1 dbasu sunuser 5699 Jul 27 12:50 dif_u_is.f → Remove Diffusive Fluxes Across Internal Surfaces.
-rw-r--r-- 1 dbasu sunuser 5325 Jul 27 12:50 dif_v_is.f → Remove Diffusive Fluxes Across Internal Surfaces.
-rw-r--r-- 1 dbasu sunuser 5185 Jul 27 12:50 dif_w_is.f → Remove Diffusive Fluxes Across Internal Surfaces.

Higher order discretization. ←
-rw-r--r-- 1 dbasu sunuser 19643 Jul 27 12:50 discretize.f → A collection of functions for higher-order discretization.
-rw-r--r-- 1 dbasu sunuser 4522 Jul 27 12:50 display_resid.f → Display Residuals.

drwxr-xr-x 3 dbasu sunuser 512 Jul 27 12:50 dmp_modules ⇒ Directory related to distributed memory parallel modules. ✓

Directory with some related files. ←
drwxr-xr-x 2 dbasu sunuser 512 Jul 27 12:50 dqmom → This is directory that has some rk automation, also source population equation & Gauss & also odeint.f.

-rw-r--r-- 1 dbasu sunuser 16601 Jul 27 12:50 drag_gs.f

Very Important code. Calculates the gas-solids drag coefficient.

→ Based on the different relationship of Syamlal, Brien & Gidaspow & Wen Yu models.

Debashis Basu — INVENTOR SIGNATURE; DEBASHIS BASU — PRINTED NAME; 8/24/2006 — DATE

DISCLOSED TO AND UNDERSTOOD BY — PRINTED NAME — DATE

-rw-r--r-- 1 dbasu sunuser 1687 Jul 27 12:50 drag_mod.f → Common Block containing Drag Arrays.

-rw-r--r-- 1 dbasu sunuser 5847 Jul 27 12:50 drag_ss.f → Computes the coefficient of drag between solids phase m and solids phase 1.

-rw-r--r-- 1 dbasu sunuser 1192 Jul 27 12:50 energy_mod.f → Defines parameters related to Heat Transfer coefficients. (Heat of reaction, Radiation temp. etc.).

-rw-r--r-- 1 dbasu sunuser 5322 Jul 27 12:50 eosg.f → Equation of state for gas.

-rw-r--r-- 1 dbasu sunuser 188 Jul 27 12:50 ep_s1.inc → Solids volume fraction function

-rw-r--r-- 1 dbasu sunuser 49 Jul 27 12:50 ep_s2.inc → one line. Defining some ε. May be solid volume fraction.

-rw-r--r-- 1 dbasu sunuser 2965 Jul 27 12:50 equal.f → Loop on the number of solid phases to set a variable equal to the value or negative value of another variable.

-rw-r--r-- 1 dbasu sunuser 3865 Jul 27 12:50 error_routine.f → Assist in printing error messages during input processing.

-rw-r--r-- 1 dbasu sunuser 2568 Jul 27 12:50 exchange.f → Calculates interphase mass, momentum and energy exchange.

-rw-r--r-- 1 dbasu sunuser 605 Jul 27 12:50 exit.f → Exiting MFIX.

-rw-r--r-- 1 dbasu sunuser 9178 Jul 27 12:50 fldvar_mod.f → Common block containing field variables data.

-rw-r--r-- 1 dbasu sunuser 24478 Jul 27 12:50 flow_to_vel.f → Convert volumetric and mass flow rates to velocities

-rw-r--r-- 1 dbasu sunuser 913 Jul 27 12:50 fun_avg1.inc → Dummy Arguments.

-rw-r--r-- 1 dbasu sunuser 1369 Jul 27 12:50 fun_avg2.inc → Arithmatic Averages

-rw-r--r-- 1 dbasu sunuser 13716 Jul 27 12:50 function.inc → Function for generating the local 3-D array index IJK from the 1-D indices I, J and K.

-rw-r--r-- 1 dbasu sunuser 2533 Jul 27 12:50 function3.inc

-rw-r--r-- 1 dbasu sunuser 126 Jul 27 12:50 function_mod.f → function for generating the local 3-D array index IJK from the 1-D indices I, J and K.

-rw-r--r-- 1 dbasu sunuser 953 Jul 27 12:50 funits_mod.f → Module funits. Which this processor should write the log file in Radial DMP mode. Distributed

-rw-r--r-- 1 dbasu sunuser 13361 Jul 27 12:50 g_0.f → Common block containing geometry and discretization data

-rw-r--r-- 1 dbasu sunuser 12271 Jul 27 12:50 geometry_mod.f → Compute area of boundary surfaces.

-rw-r--r-- 1 dbasu sunuser 4646 Jul 27 12:50 get_bc_area.f → Read a variety input data

-rw-r--r-- 1 dbasu sunuser 10622 Jul 27 12:50 get_data.f

-rw-r--r-- 1 dbasu sunuser 4211 Jul 27 12:50 get_eq.f → complete the filling of coeff-matrix A.

Mostly related to different functions & other equation etc.

PS → Pressure coeff (6)
110
40

110 40

38
22
46
46

Debashis Basu — INVENTOR SIGNATURE; DEBASHIS BASU — PRINTED NAME; 8/24/2006 — DATE

DISCLOSED TO AND UNDERSTOOD BY — PRINTED NAME — DATE

3/12/2006 2:00

```
-rw-r--r-- 1 dbasu sunuser 17292 Jul 27 12:50 get_flow_bc.f
-rw-r--r-- 1 dbasu sunuser 2895 Jul 27 12:50 get_hloss.f
-rw-r--r-- 1 dbasu sunuser 8250 Jul 27 12:50 get_is.f
-rw-r--r-- 1 dbasu sunuser 7669 Jul 27 12:50 get_philoss.f
-rw-r--r-- 1 dbasu sunuser 3329 Jul 27 12:50 get_smass.f
-rw-r--r-- 1 dbasu sunuser 7836 Jul 27 12:50 get_stats.f
-rw-r--r-- 1 dbasu sunuser 9511 Jul 27 12:50 get_walls_bc.f
-rw-r--r-- 1 dbasu sunuser 7047 Jul 27 12:50 ic_mod.f
-rw-r--r-- 1 dbasu sunuser 4909 Jul 27 12:50 in_bin_512.f
-rw-r--r-- 1 dbasu sunuser 5404 Jul 27 12:50 in_bin_512i.f
-rw-r--r-- 1 dbasu sunuser 4942 Jul 27 12:50 indices_mod.f
-rw-r--r-- 1 dbasu sunuser 3656 Jul 27 12:50 init_ab_m.f
-rw-r--r-- 1 dbasu sunuser 4595 Jul 27 12:50 init_fvars.f
-rw-r--r-- 1 dbasu sunuser 15950 Jul 27 12:50 init_namelist.f
-rw-r--r-- 1 dbasu sunuser 2500 Jul 27 12:50 init_resid.f
-rw-r--r-- 1 dbasu sunuser 4665 Jul 27 12:50 is_mod.f
-rw-r--r-- 1 dbasu sunuser 20822 Jul 27 12:50 iterate.f
-rw-r--r-- 1 dbasu sunuser 20104 Jul 27 12:50 k_epsilon_prop.f
-rw-r--r-- 1 dbasu sunuser 101450 Jul 27 12:50 leq_bicgs.f
-rw-r--r-- 1 dbasu sunuser 24400 Jul 27 12:50 leq_gmres.f
-rw-r--r-- 1 dbasu sunuser 3459 Jul 27 12:50 leq_sor.f
-rw-r--r-- 1 dbasu sunuser 1071 Jul 27 12:50 leqsol_mod.f
-rw-r--r-- 1 dbasu sunuser 2799 Jul 27 12:50 line_too_big.f
```

- get_flow_bc.f → find and validate i, j, k locations for FLOW BC's.
- get_hloss.f → Determine the total heat loss from the reactor.
- get_is.f → Find and validate i, j, k locations for IS's.
- get_philoss.f → Determine the total loss of scalar φ through reactor walls.
- get_smass.f → Determine the weight of solids in the reactor.
- get_stats.f → Get the statistics for stalled or diverged iterations.
- get_walls_bc.f → Find and validate i, j, k locations for wall BC's.
- ic_mod.f → Common block containing initial condition data.
- in_bin_512.f → Read an array in chunks of 512 bytes.
- in_bin_512i.f → Read an array in an array in chunks of 512 bytes.
- indices_mod.f → Common block containing arrays for index computation.
- init_ab_m.f → Initialize the sparse matrix coefficients and the source vector.
- init_fvars.f → Initialize all field variables as undefined.
- init_namelist.f → Initialize the namelist variables.
- init_resid.f → Initidize Residuals.
- is_mod.f → Include file for internal surface specifications.
- iterate.f → this module controls the iterations for solving equations version 2.0 (very important subroutines)
- k_epsilon_prop.f → Calculates diffusion coefficient and sources for k & ε equations.
- leq_bicgs.f → compute Residual for linear System (Big subroutine.)
- leq_gmres.f → Solves system of linear system equation using GMRES. (Solvers using GMRES, OR etc.)
- leq_sor.f → Successive over-relaxation method.
- leqsol_mod.f → Module leq sol.
- line_too_big.f → Return an error condition if input data is located first column MAXCOL in the data input file.


```
-rw-r--r-- 1 dbasu sunuser 2545 Jul 27 12:50 location.f
-rw-r--r-- 1 dbasu sunuser 3862 Jul 27 12:50 location_check.f
-rw-r--r-- 1 dbasu sunuser 10060 Jul 27 12:50 machine.f
-rw-r--r-- 1 dbasu sunuser 593 Jul 27 12:50 machine_mod.f
-rwxr-xr-x 1 dbasu sunuser 54664 Jul 27 12:50 make_mfix
-rw-r--r-- 1 dbasu sunuser 5390 Jul 27 12:50 make_upper_case.f
-rw-r--r-- 1 dbasu sunuser 6397 Jul 27 12:50 mark_phase_4_cor.f
-rw-r--r-- 1 dbasu sunuser 667 Jul 27 12:50 matrix_mod.f
-rw-r--r-- 1 dbasu sunuser 10152 Jul 27 12:50 mfix.f
-rw-r--r-- 1 dbasu sunuser 262 Jul 27 12:50 mfix_directory_path.inc
-rw-r--r-- 1 dbasu sunuser 150812 Jul 27 12:50 mfix_l.make
-rw-r--r-- 1 dbasu sunuser 150812 Jul 27 12:50 mfix_u.make
-rw-r--r-- 1 dbasu sunuser 3108 Jul 27 12:50 mflux_mod.f
-rw-r--r-- 1 dbasu sunuser 4183 Jul 27 12:50 mod_bc_i.f
-rw-r--r-- 1 dbasu sunuser 4170 Jul 27 12:50 mod_bc_j.f
-rw-r--r-- 1 dbasu sunuser 4236 Jul 27 12:50 mod_bc_k.f
-rw-r--r-- 1 dbasu sunuser 70 Jul 27 12:50 mpif.h
-rw-r--r-- 1 dbasu sunuser 4769 Jul 27 12:50 namelist.inc
-rw-r--r-- 1 dbasu sunuser 3352 Jul 27 12:50 open_file.PC
-rw-r--r-- 1 dbasu sunuser 3338 Jul 27 12:50 open_file.f
-rw-r--r-- 1 dbasu sunuser 8500 Jul 27 12:50 open_files.f
-rw-r--r-- 1 dbasu sunuser 3538 Jul 27 12:50 out_array.f
-rw-r--r-- 1 dbasu sunuser 3330 Jul 27 12:50 out_array_c.f
```

- location.f → find the cell center location in x, y, z direction for the given index I2.
- location_check.f → check calculated a given cell locations for consistency.
- machine.f → Set the machine constants. (LINUX ABSOFT only) (Related to machines)
- machine_mod.f → Module machine. (Related to machines)
- make_mfix → MFIX script file for obtaining user-defined files and executing 'make'. → This is the makefile.
- make_upper_case.f → change lowercase characters to uppercase in input file.
- mark_phase_4_cor.f → For each cell, mark the phase whose continuity will be used to form pressure correction equations. (Important related to pressure correction)
- matrix_mod.f → Module matrix.
- mfix.f → Main module of the mfix code. (Main Module.)
- mfix_directory_path.inc → File containing path to mfix/model directory.
- mfix_l.make → list all files and subroutines that are needed by MFIX to run, including executables.
- mfix_u.make → Almost similar to mfix_l.make
- mflux_mod.f → Module for mass fluxes and densities at faces.
- mod_bc_i.f → Modify the "I" values for the b.c. plane. (Modifying the B.C. values in i, j, k planes.)
- mod_bc_j.f → modify the "j" values for the b.c. plane.
- mod_bc_k.f → modify the "k" values for the b.c. plane.
- mpif.h → For MPI run.
- namelist.inc → Namelist in every subroutine and subroutines. Namelist & input data related.
- open_file.PC → opening a file (opening files & writing outputs.)
- open_file.f → opening a file.
- open_files.f → open all the files for this run.
- out_array.f → Print out a 3-D array to standard output.
- out_array_c.f → Print out a 3D array to standard output.

output types & how to get the outputs etc. etc.

```
-rw-r--r--  1 dbasu  sunuser  4247 Jul 27 12:50 out_array_k.f
```
→ Print out a 2D (constant k-plane) array to standard output.

```
-rw-r--r--  1 dbasu  sunuser  4097 Jul 27 12:50 out_array_kc.f
```
→ Print out a 2D (constant k plane) array to standard output (character)

```
-rw-r--r--  1 dbasu  sunuser  3646 Jul 27 12:50 out_bin_512.f
```
→ write out an array in chunks of 512 bytes

```
-rw-r--r--  1 dbasu  sunuser  3668 Jul 27 12:50 out_bin_512i.f
```
→ write out an array in chunks of 512 bytes (Integer words).

```
-rw-r--r--  1 dbasu  sunuser  3668 Jul 27 12:50 out_bin_512r.f
```
→ write out an array in chunks of 512 bytes (real words).

```
-rw-r--r--  1 dbasu  sunuser  3160 Jul 27 12:50 out_bin_r.f
```
→ write out a time dependent restart variable.

```
-rw-r--r--  1 dbasu  sunuser  5527 Jul 27 12:50 output_mod.f
```
→ Common block containing data for output control.

```
-rw-r--r--  1 dbasu  sunuser  398 Jul 27 12:50 parallel_mod.f
```
→ Module Parallel. Definitions for parallel programming.

Parameters Related.

```
-rw-r--r--  1 dbasu  sunuser  2013 Jul 27 12:50 param1_mod.f
```
→ Module Param1.

```
-rw-r--r--  1 dbasu  sunuser  4083 Jul 27 12:50 param_mod.f
```
→ Module Param.

Parse calculations. Used for kinetics, stoichiometry etc.

```
-rw-r--r--  1 dbasu  sunuser  16649 Jul 27 12:50 parse_line.f
```
→ Parse Input line.

```
-rw-r--r--  1 dbasu  sunuser  708 Jul 27 12:50 parse_mod.f
```
→ Used for parse matrix

```
-rw-r--r--  1 dbasu  sunuser  4676 Jul 27 12:50 parse_resid_string.f
```
→ Initialize Residuals.

```
-rw-r--r--  1 dbasu  sunuser  36089 Jul 27 12:50 parse_rxn.f
```
→ Parse Input line.

Important Subroutine

```
-rw-r--r--  1 dbasu  sunuser  26617 Jul 27 12:50 partial_elim.f
```
→ Do partial elimination for scalar quantities.

```
-rw-r--r--  1 dbasu  sunuser  920 Jul 27 12:50 pgcor_mod.f
```
→ Module pgcor. Variables for pressure correction equation.

Important Subroutine

```
-rw-r--r--  1 dbasu  sunuser  11048 Jul 27 12:50 physical_prop.f
```
→ Calculates physical property that vary with time.

```
-rw-r--r--  1 dbasu  sunuser  6313 Jul 27 12:50 physprop_mod.f
```
→ Common block containing physical property data.

```
-rw-r--r--  1 dbasu  sunuser  1325 Jul 27 12:50 pscor_mod.f
```
→ Module PSCOR. → variables for solids pressure correction equation.

Sources Terms Related to Radiation

```
-rw-r--r--  1 dbasu  sunuser  204 Jul 27 12:50 radtn1.inc
```
→ Statement functions for ~~source terms~~ in the energy equations. → set up for Radiation.

```
-rw-r--r--  1 dbasu  sunuser  730 Jul 27 12:50 radtn2.inc
```
→ Source terms for the energy equation. The source term is linearized as S = S_C - S_P * T.

```
-rw-r--r--  1 dbasu  sunuser  4821 Jul 27 12:50 read_database.f
```
→ Read thermochemical database.

```
-rw-r--r--  1 dbasu  sunuser  12835 Jul 27 12:50 read_namelist.f
```
→ Read in the namelist variables.

→ Reading all variables according to their names.

Debashis Basu
DEBASHIS BASU
8/25/06
PRINTED NAME
DATE
DISCLOSED TO AND UNDERSTOOD BY
PRINTED NAME
DATE

This is about reading the restart data.

```
-rw-r--r--  1 dbasu  sunuser  47259 Jul 27 12:50 read_res0.f
```
→ Read the initial restart records. (namelist data).

```
-rw-r--r--  1 dbasu  sunuser  8918 Jul 27 12:50 read_res1.f
```
→ Read in the time-dependent restart records.

```
-rw-r--r--  1 dbasu  sunuser  2241 Jul 27 12:50 remove_comment.f
```
→ Removing comments.

Resetting with old value.

```
-rw-r--r--  1 dbasu  sunuser  3814 Jul 27 12:50 reset_new.f
```
→ Reset the new variables with the stored previous time-step values of field variables.

```
-rw-r--r--  1 dbasu  sunuser  2187 Jul 27 12:50 residual_mod.f
```
→ MODULE "Residual".

Important Subroutine Related to reaction rates.

```
-rw-r--r--  1 dbasu  sunuser  10977 Jul 27 12:50 rrates.f
```
→ Calculates reaction rates for various reactions in cell i,j,k.

```
-rw-r--r--  1 dbasu  sunuser  11487 Jul 27 12:50 rrates0.f
```
→ Calculate reaction rates for various reactions in cell i,j,k using information from the data file.

```
-rw-r--r--  1 dbasu  sunuser  2187 Jul 27 12:50 rrates_init.f
```
→ Initialize Reaction Rate Arrays.

```
-rw-r--r--  1 dbasu  sunuser  8341 Jul 27 12:50 run_mod.f
```
→ common block containing run control data.

Important Subroutine

```
-rw-r--r--  1 dbasu  sunuser  4513 Jul 27 12:50 rxns_mod.f
```
→ RXNS module. Related to reaction rates, phase change etc.

Related to solids related terms.

```
-rw-r--r--  1 dbasu  sunuser  455 Jul 27 12:50 s_pr1.inc
```
→ coefficient in Jackson's model. → voidage of random close-packed bed.

```
-rw-r--r--  1 dbasu  sunuser  2694 Jul 27 12:50 s_pr2.inc
```
→ Compute solids pressure and its inverse.

```
-rw-r--r--  1 dbasu  sunuser  38 Jul 27 12:50 sc_p_gl.inc
-rw-r--r--  1 dbasu  sunuser  91 Jul 27 12:50 sc_p_g2.inc
```
→ Double Precision Scale, Unscale definition & defining.

Useful for modification of any additional scalar variable given.

```
-rw-r--r--  1 dbasu  sunuser  3218 Jul 27 12:50 scalar_prop.f
```
→ Calculate diffusion coefficient and sources for user-defined scalars.

Scalars related programs.

```
-rw-r--r--  1 dbasu  sunuser  1525 Jul 27 12:50 scalars_mod.f
```
→ Number of scalar equations solved

```
-rw-r--r--  1 dbasu  sunuser  524 Jul 27 12:50 scales_mod.f
```
→ Module scales.

```
-rw-r--r--  1 dbasu  sunuser  3179 Jul 27 12:50 seek_comment.f
```
→ Determine if (and where) a comment character appears in a data input file.

```
-rw-r--r--  1 dbasu  sunuser  2152 Jul 27 12:50 seek_end.f
```
→ Determine where trailing blanks begin in a line.

These two modules are related to the boundary conditions.

```
-rw-r--r--  1 dbasu  sunuser  14555 Jul 27 12:50 set_bc0.f
```
→ This module does the initial setting of all boundary conditions.

```
-rw-r--r--  1 dbasu  sunuser  15487 Jul 27 12:50 set_bc1.f
```
→ Set transient flow boundary conditions.

```
-rw-r--r--  1 dbasu  sunuser  5250 Jul 27 12:50 set_constants.f
```
→ This module sets all the constants.

```
-rw-r--r--  1 dbasu  sunuser  5830 Jul 27 12:50 set_constprop.f
```
→ This module sets all the constant physical properties.

→ Setting up of all the constants and all the constant physical properties.

Debashis Basu
INVENTOR SIGNATURE
DEBASHIS BASU
PRINTED NAME
8/25/06
DATE
DISCLOSED TO AND UNDERSTOOD BY
PRINTED NAME
DATE

1 06002-01-302

```
-rw-r--r--  1 dbasu  sunuser  17667 Jul 27 12:50 set_flags.f
-rw-r--r--  1 dbasu  sunuser   9226 Jul 27 12:50 set_fluidbed_p.f
-rw-r--r--  1 dbasu  sunuser  11061 Jul 27 12:50 set_geometry.f
-rw-r--r--  1 dbasu  sunuser   3672 Jul 27 12:50 set_geometry1.f
-rw-r--r--  1 dbasu  sunuser   9708 Jul 27 12:50 set_ic.f
-rw-r--r--  1 dbasu  sunuser  11003 Jul 27 12:50 set_increments.f
-rw-r--r--  1 dbasu  sunuser  10017 Jul 27 12:50 set_increments3.f
-rw-r--r--  1 dbasu  sunuser   4373 Jul 27 12:50 set_index1.f
-rw-r--r--  1 dbasu  sunuser   8514 Jul 27 12:50 set_index1a.f
-rw-r--r--  1 dbasu  sunuser   4584 Jul 27 12:50 set_index1a3.f
-rw-r--r--  1 dbasu  sunuser   2606 Jul 27 12:50 set_l_scale.f
-rw-r--r--  1 dbasu  sunuser   5220 Jul 27 12:50 set_max2.f
-rw-r--r--  1 dbasu  sunuser   2949 Jul 27 12:50 set_mw_mix_g.f
-rw-r--r--  1 dbasu  sunuser  22539 Jul 27 12:50 set_outflow.f
-rw-r--r--  1 dbasu  sunuser   3677 Jul 27 12:50 set_ro_g.f
-rw-r--r--  1 dbasu  sunuser  17276 Jul 27 12:50 set_wall_bc.f
-rw-r--r--  1 dbasu  sunuser   3541 Jul 27 12:50 shift_dxyz.f
-rw-r--r--  1 dbasu  sunuser   5030 Jul 27 12:50 solve_continuity.f
-rw-r--r--  1 dbasu  sunuser   8642 Jul 27 12:50 solve_energy_eq.f
-rw-r--r--  1 dbasu  sunuser   4387 Jul 27 12:50 solve_epp.f
-rw-r--r--  1 dbasu  sunuser   6655 Jul 27 12:50 solve_granular_energy.f
-rw-r--r--  1 dbasu  sunuser  12575 Jul 27 12:50 solve_k_epsilon_eq.f
-rw-r--r--  1 dbasu  sunuser   4922 Jul 27 12:50 solve_lin_eq.f
```

Important subroutine for FLUIDIZED BED computations.

Geometry related subroutines

Setting the index of neighboured cells.

Subroutine for turbulent length scale.

Important for Boundary Condition at outflow

Important for wall Boundary Condition

Volume fraction

Energy equations

V. V. Imp.

- set_flags.f → This module assigns a flag to a cell to identify its type.
- set_fluidbed_p.f → Set the pressure field inside the bed assuming a fluidized bed with gravity acting in -ve y-direction.
- set_geometry.f → Calculate X, X_E, oX, oX_E → Setup up the geometry.
- set_geometry1.f → Calculate cell volume & faces
- set_ic.f → This module sets all the initial conditions.
- set_increments.f → This module creates increments to be stored in the array store_increment which is added to cell index ijk to find the effective indices of neighbors.
- set_increments3.f → Same as above.
- set_index1.f → Same as above.
- set_index1a.f → Set the indices of the neighbors of cell ijk by brute force.
- set_index1a3.f → Set the indices of the neighbors of cell ijk by brute force.
- set_l_scale.f → Initialize the length scale for turbulence model.
- set_max2.f → calculate IMAX2, JMAX2, IMAX1, JMAX1, KMAX1 etc.
- set_mw_mix_g.f → calculate gas mixture molecular weight.
- set_outflow.f → Set specified pressure outflow BC for a specified range of cells.
- set_ro_g.f → Initialize the gas densities. (Initialize the gas densities)
- set_wall_bc.f → Set the wall boundary condition.
- shift_dxyz.f → shift the data in the dx, dy, dz arrays from 1:IMAX to IMIN1:IMAX1 etc.
- solve_continuity.f → Solve for volume fractions.
- solve_energy_eq.f → Solve the energy equations.
- solve_epp.f → Solve solids volume fraction correction equation.
- solve_granular_energy.f → Solves granular energy equations.
- solve_k_epsilon_eq.f → Solves K & ε equations for a turbulent flow
- solve_lin_eq.f → Interface for Linear Equation Solvers.

Debashis Basu | DEBASHIS BASU | 8/26/06
INVENTOR SIGNATURE | PRINTED NAME | DATE

DISCLOSED TO AND UNDERSTOOD BY | PRINTED NAME | DATE


```
-rw-r--r--  1 dbasu  sunuser   4785 Jul 27 12:50 solve_pp_g.f
-rw-r--r--  1 dbasu  sunuser   8682 Jul 27 12:50 solve_scalar_eq.f
-rw-r--r--  1 dbasu  sunuser   8509 Jul 27 12:50 solve_species_eq.f
-rw-r--r--  1 dbasu  sunuser  15908 Jul 27 12:50 solve_vel_star.f
-rw-r--r--  1 dbasu  sunuser   6531 Jul 27 12:50 source_granular_energy.f
-rw-r--r--  1 dbasu  sunuser   5849 Jul 27 12:50 source_phi.f
-rw-r--r--  1 dbasu  sunuser  12693 Jul 27 12:50 source_pp_g.f
-rw-r--r--  1 dbasu  sunuser   5472 Jul 27 12:50 source_rop_g.f
-rw-r--r--  1 dbasu  sunuser   5743 Jul 27 12:50 source_rop_s.f
-rw-r--r--  1 dbasu  sunuser  32662 Jul 27 12:50 source_u_g.f
-rw-r--r--  1 dbasu  sunuser  39831 Jul 27 12:50 source_u_s.f
-rw-r--r--  1 dbasu  sunuser  29592 Jul 27 12:50 source_v_g.f
-rw-r--r--  1 dbasu  sunuser  39475 Jul 27 12:50 source_v_s.f
-rw-r--r--  1 dbasu  sunuser  33033 Jul 27 12:50 source_w_g.f
-rw-r--r--  1 dbasu  sunuser  42625 Jul 27 12:50 source_w_s.f
-rw-r--r--  1 dbasu  sunuser    211 Jul 27 12:50 species_indices.inc
-rw-r--r--  1 dbasu  sunuser    421 Jul 27 12:50 tau_g_mod.f
-rw-r--r--  1 dbasu  sunuser    439 Jul 27 12:50 tau_s_mod.f
-rw-r--r--  1 dbasu  sunuser   6727 Jul 27 12:50 tau_u_g.f
-rw-r--r--  1 dbasu  sunuser   7690 Jul 27 12:50 tau_u_s.f
-rw-r--r--  1 dbasu  sunuser   5763 Jul 27 12:50 tau_v_g.f
-rw-r--r--  1 dbasu  sunuser   6594 Jul 27 12:50 tau_v_s.f
-rw-r--r--  1 dbasu  sunuser   6888 Jul 27 12:50 tau_w_g.f
```

Subroutines for solving.

very important.

Calculates all the source terms in the different equations.

For gas phase.

For solids phase.

Calculates the cross terms in the gradient of stress in the different momentum equations.
→ Gas
→ Solids.

- solve_pp_g.f → Solve fluid pressure correction equation.
- solve_scalar_eq.f → Solve the scalar transport equations.
- solve_species_eq.f → Solve species mass balance equations.
- solve_vel_star.f → Solve starred velocity components. [But this one also has corrections/modification by Dartevelle & Jay Boyalakuntla]
- source_granular_energy.f → Calculate the source terms in the granular energy equation.
- source_phi.f → Determine source term for the phi equation.
- source_pp_g.f → Determine the source terms for pressure correction equation. The off-diagonal terms are positive. The center coeff & source vect are negative.
- source_rop_g.f → Determine the source term for continuity equation (for gas)
- source_rop_s.f → Determine the source term for solids continuity equation
- source_u_g.f → Determine the source terms for U-g momentum equations.
- source_u_s.f → Determine the source terms for U-s momentum equation.
- source_v_g.f → Determine the source terms for the V-g momentum equation.
- source_v_s.f → Determine the source terms for the V-s momentum equations.
- source_w_g.f → Determine the source terms for the W-g momentum equation.
- source_w_s.f → Determine the source terms for the W-s momentum equation.
- species_indices.inc → Gas Phase Species Indices.
- tau_g_mod.f → Module tau_g.
- tau_s_mod.f → Module tau_s.
- tau_u_g.f → Cross terms in the gradient of stress in U-g momentum
- tau_u_s.f → Cross terms in the gradient of stress in U-s momentum
- tau_v_g.f → Cross terms in the gradient of stress in V-g momentum
- tau_v_s.f → Cross terms in the gradient of stress in V-s momentum
- tau_w_g.f → Cross terms in the gradient of stress in W-g momentum
- tau_w_s.f → Cross terms in the gradient of stress in W-s momentum

Debashis Basu | DEBASHIS BASU | 8/28/06
INVENTOR SIGNATURE | PRINTED NAME | DATE

DISCLOSED TO AND UNDERSTOOD BY | PRINTED NAME | DATE

```
-rw-r--r--  1 dbasu   sunuser   7234 Jul 27 12:50 tau_w_s.f
-rw-r--r--  1 dbasu   sunuser   6934 Jul 27 12:50 test_lin_eq.f
drwxr-xr-x  2 dbasu   sunuser    512 Jul 27 12:50 thermochemical
-rw-r--r--  1 dbasu   sunuser    477 Jul 27 12:50 time_cpu_mod.f
-rw-r--r--  1 dbasu   sunuser  19199 Jul 27 12:50 time_march.f
-rw-r--r--  1 dbasu   sunuser    854 Jul 27 12:50 tmp_array1_mod.f
-rw-r--r--  1 dbasu   sunuser   2337 Jul 27 12:50 tmp_array_mod.f
-rw-r--r--  1 dbasu   sunuser   4202 Jul 27 12:50 toleranc_mod.f
-rw-r--r--  1 dbasu   sunuser    546 Jul 27 12:50 trace_mod.f
-rw-r--r--  1 dbasu   sunuser   2583 Jul 27 12:50 transfer.f
-rw-r--r--  1 dbasu   sunuser   3372 Jul 27 12:50 transport_prop.f
-rw-r--r--  1 dbasu   sunuser   1166 Jul 27 12:50 turb_mod.f
-rw-r--r--  1 dbasu   sunuser   2235 Jul 27 12:50 undef_2_0.f
-rw-r--r--  1 dbasu   sunuser  12932 Jul 27 12:50 under_relax.f
-rw-r--r--  1 dbasu   sunuser   4175 Jul 27 12:50 update_old.f
-rw-r--r--  1 dbasu   sunuser    859 Jul 27 12:50 ur_facs_mod.f
-rw-r--r--  1 dbasu   sunuser   2321 Jul 27 12:50 usr0.f
-rw-r--r--  1 dbasu   sunuser   2337 Jul 27 12:50 usr1.f
-rw-r--r--  1 dbasu   sunuser   2410 Jul 27 12:50 usr2.f
-rw-r--r--  1 dbasu   sunuser   2224 Jul 27 12:50 usr3.f
-rw-r--r--  1 dbasu   sunuser   1823 Jul 27 12:50 usr_init_namelist.f
-rw-r--r--  1 dbasu   sunuser    619 Jul 27 12:50 usr_mod.f
-rw-r--r--  1 dbasu   sunuser   1720 Jul 27 12:50 usr_write_out0.f
```

- tau_w_s.f → Cross terms in the gradient of stress in w-s momentum.
- test_lin_eq.f → Routine for testing the accuracy of linear equation solver.
- thermochemical (directory) → Directory for thermochemical calculations & properties.
- time_cpu_mod.f → Module for time_cpu.
- time_march.f (very important subroutine. controls time marching.) → This is a very important subroutine. This controls the time integration/time marching.
- tmp_array1_mod.f → Module_tmp_array1
- tmp_array_mod.f → Module_tmp_array
- toleranc_mod.f → Specify all tolerance parameters.
- trace_mod.f → Module trace.
- transfer.f → Transfer the result of linear equation solver to variable array.
- transport_prop.f (This is a very important subroutine) → Calculate transport properties that vary with time. ⇒ This is a very important
- turb_mod.f → Module turb.
- undef_2_0.f → Change undefined values to zero. Otherwise linear equation solver does not work.
- under_relax.f (under relaxing u momentum, v momentum, w ") → Under relaxing scalar equations.
- update_old.f → update the stored previous-time-step values of certain field variables. (u, v, w, ρ, P, e etc)
- ur_facs_mod.f → MODULE ur_facs.
- usr0.f, usr1.f, usr2.f, usr3.f (These are modules for user-defined subroutines.) → User defined function subroutine. These are all user defined function subroutine
- usr_init_namelist.f → Initialize user-defined namelist variables.
- usr_mod.f → Module usr.
- usr_write_out0.f → Echo user-defined input.

Debashis Basu — DEBASHIS BASU — 9/5/06 DATE

DISCLOSED TO AND UNDERSTOOD BY — PRINTED NAME — DATE


```
-rw-r--r--  1 dbasu   sunuser   1747 Jul 27 12:50 usr_write_out1.f
-rw-r--r--  1 dbasu   sunuser    647 Jul 27 12:50 usrnlst.inc
-rw-r--r--  1 dbasu   sunuser   9708 Jul 27 12:50 utilities.f
-rw-r--r--  1 dbasu   sunuser   5418 Jul 27 12:50 vavg_u_g.f
-rw-r--r--  1 dbasu   sunuser   3568 Jul 27 12:50 vavg_u_s.f
-rw-r--r--  1 dbasu   sunuser   5430 Jul 27 12:50 vavg_v_g.f
-rw-r--r--  1 dbasu   sunuser   3264 Jul 27 12:50 vavg_v_s.f
-rw-r--r--  1 dbasu   sunuser   5424 Jul 27 12:50 vavg_w_g.f
-rw-r--r--  1 dbasu   sunuser   3545 Jul 27 12:50 vavg_w_s.f
-rw-r--r--  1 dbasu   sunuser   5901 Jul 27 12:50 vf_gs_x.f
-rw-r--r--  1 dbasu   sunuser   5900 Jul 27 12:50 vf_gs_y.f
-rw-r--r--  1 dbasu   sunuser   5916 Jul 27 12:50 vf_gs_z.f
-rw-r--r--  1 dbasu   sunuser    899 Jul 27 12:50 visc_g_mod.f
-rw-r--r--  1 dbasu   sunuser   3201 Jul 27 12:50 visc_s_mod.f
-rw-r--r--  1 dbasu   sunuser    266 Jul 27 12:50 vshear_mod.f
-rw-r--r--  1 dbasu   sunuser   4948 Jul 27 12:50 write_ab_m.f
-rw-r--r--  1 dbasu   sunuser   4790 Jul 27 12:50 write_ab_m_var.f
-rw-r--r--  1 dbasu   sunuser   2489 Jul 27 12:50 write_error.f
-rw-r--r--  1 dbasu   sunuser   2742 Jul 27 12:50 write_header.f
-rw-r--r--  1 dbasu   sunuser  40088 Jul 27 12:50 write_out0.f
-rw-r--r--  1 dbasu   sunuser  12291 Jul 27 12:50 write_out1.f
-rw-r--r--  1 dbasu   sunuser   3019 Jul 27 12:50 write_out3.f
-rw-r--r--  1 dbasu   sunuser  17966 Jul 27 12:50 write_res0.f
```

- usr_write_out1.f → write user defined output.
- usrnlst.inc → user defined namelist related.
- utilities.f → Have a logical function isnan. Also have modules (MAX-VEL-INLET) & CHECK-VEL-SOUND
- vavg_u_g.f → volume average u_g.
- vavg_u_s.f → Volume average U_s.
- vavg_v_g.f → Volume average V_g.
- vavg_v_s.f → Volume average V_s.
- vavg_w_g.f → Volume average W_g.
- vavg_w_s.f → Volume average W_s.
- (Related to checking velocities. Related to volume averaging. → gas → solid)
- vf_gs_x.f → Calculate the average drag coefficient at $i+1/2, j, k$ & multiply with u momentum cell volume.
- vf_gs_y.f → Calculate the average drag coefficient at $i, j+1/2, k$ and multiply with v-momentum cell volume.
- vf_gs_z.f → Calculate the average drag coeff. at $i, j, k+1/2$ and multiply with w-momentum cell vol.
- (Average drag efficient - multiply th cell volume.)
- visc_g_mod.f → Module visc_g → Related to Gas viscosity.
- visc_s_mod.f → Module visc_s → Related to Solid viscosity.
- vshear_mod.f → Module vshear.
- write_ab_m.f → Write the sparse matrix coefficients and the source vector.
- write_ab_m_var.f → write the sparse matrix coefficients and the source vector.
- (All most same both of them. But some differ in the variable.)
- write_error.f → write an error message.
- write_header.f → Read & verify input data, open files.
- write_out0.f → echo user input
- write_out1.f → write out the field variables to standard output.
- write_out3.f → To write CPU time used by the code.
- (writing out different things.)
- write_res0.f → write out the initial restart records.

Debashis Basu — DEBASHIS BASU — 9/5/06

INVENTOR SIGNATURE — PRINTED NAME — DATE

DISCLOSED TO AND UNDERSTOOD BY — PRINTED NAME — DATE

1 BOOK NO. 06002.01.302 PROJECT

```
-rw-r--r--  1 dbasu   sunuser   8024 Jul 27 12:50 write_res1.f  → write out the time d
                                                                  restart records.
-rw-r--r--  1 dbasu   sunuser   3445 Jul 27 12:50 write_spx0.f  → write out the initial re
                                                                  records (REAL)
-rw-r--r--  1 dbasu   sunuser  13025 Jul 27 12:50 write_spx1.f  → write out the time depe
                                                                  records.
-rw-r--r--  1 dbasu   sunuser   4565 Jul 27 12:50 write_table.f  → to write a table of D>
                                                                  and cell wall locations.
-rw-r--r--  1 dbasu   sunuser   1780 Jul 27 12:50 write_usr0.f  → write initial part of u
                                                                  output.
-rw-r--r--  1 dbasu   sunuser   1953 Jul 27 12:50 write_usr1.f  → write user defined output.
-rw-r--r--  1 dbasu   sunuser   1149 Jul 27 12:50 xerbla.f  → Subroutine XERBLA.
                                                                  Modules.
-rw-r--r--  1 dbasu   sunuser    327 Jul 27 12:50 xsi1.inc  → Special inline for xsi.
-rw-r--r--  1 dbasu   sunuser    398 Jul 27 12:50 xsi2.inc  → Special inline for xsi.
-rw-r--r--  1 dbasu   sunuser    915 Jul 27 12:50 xsi_array_mod.f  → Module xsi_array.
-rw-r--r--  1 dbasu   sunuser   2450 Jul 27 12:50 zero_array.f  → zero out an array.
-rw-r--r--  1 dbasu   sunuser   5111 Jul 27 12:50 zero_norm_vel.f  → Set the velocity co
                                                                  normal to a wall
[dbasu@texas] /home/dbasu/mfix/model {13} #
```

Debashis Basu | DEBASHIS BASU | 09/08/2006
INVENTOR SIGNATURE | PRINTED NAME | DATE

DISCLOSED TO AND UNDERSTOOD BY | PRINTED NAME | DATE

604 Macromolecular Materials and Engineering B. K. Chandrasekaran, L. van der Lee, I. Hulme, A. Kantzas

Information potentially subject to copyright protection was redacted from this page through page 46 of this scientific notebook. The redacted material is an article from the Macromolecular Materials and Engineering Journal, 2005, 290, 592-609, entitled A Simulation and Experimental Study of the Hydrodynamics of a Bubbling Fluidized Bed, by B.K. Chandrasckaram,, L. van der Lee, I. Helme, and A. Kantzas

Debashis Basu | DEBASHIS BASU | 9/9/06
INVENTOR SIGNATURE | PRINTED NAME | DATE

DISCLOSED TO AND UNDERSTOOD BY | PRINTED NAME | DATE

See page 17.

Debashis Basu. DEBASHIS BASU
INVENTOR SIGNATURE PRINTED NAME DATE

DISCLOSED TO AND UNDERSTOOD BY PRINTED NAME DATE

REFERENCES :

1. Chandrasekharan, B. K., Lee, Loni van der, Hulme, Ian., & Kantzas, A., "A simulation and Experimental Study of the Hydrodynamics of a Bubbling Fluidized Bed of Linear Low Density Polyethylene Using Bubble Properties and Pressure Fluctuations", Macromolecular Materials and Engineering, Vol. 290, pp. 592-609, 2005.

2. Dartevelle, S., "Numerical modeling of geophysical granular flows: 1. A comprehensive approach to granular rheologies and geophysical multiphase flows" GEOCHEMISTRY GEOPHYSICS GEOSYSTEMS, VOL. 5, Q08003, doi:10.1029/2003GC000636, 2004.

3. Dartevelle, S., "Numerical modeling of geophysical granular flows: 2. Computer simulations of plinian clouds and pyroclastic flows and surges", GEOCHEMISTRY GEOPHYSICS GEOSYSTEMS, VOL. 5, Q08004, doi:10.1029/2003GC000637, 2004.

4. Syamlal, M., Rogers, W., and O'Brien, T. J., "MFIX documentation: Theory guide, DOE/METC-94/1004, DE9400,097,49 pp., U. S. Department of Energy, Washington, D. C.

Debashis Basu. DEBASHIS BASU
INVENTOR SIGNATURE PRINTED NAME DATE

DISCLOSED TO AND UNDERSTOOD BY PRINTED NAME DATE