Proceedings of the Korean Nuclear Autumn Meeting
Yongpyong, Korea, October 2002

Development of Lagrangian Model for Fuel-Coolant Interaction

Sunchai Nilsuwankosit*, Jin Ho Song
Thermal Hydraulics Safety Research Team
Korea Atomic Energy Research Institute
P.O.Box 105, Yuseong, Daejeon, Korea, 305-600
Fax : +82-42-861-2574

Abstract

A numerical modules for simulating the motion of the Lagrangian particles and their interaction with the surrounding Eulerian fluids is currently being developed by the Thermal Hydraulics Safety Research Team at Korea Atomic Energy Research Institute (KAERI). The models employed for the modules are similar in concept to that used in TEXAS which is the Lagrangian-Eulerian one dimensional code. Unlike TEXAS, however, the motion of the particles as modeled for the modules can be simulated in one, two or three dimensions. In addition, the modules being developed are not intended to be used as a stand-alone code. Rather, the modules have been designed to be easily incorporated into the existing codes that are mainly based on the concept of the multiphase fluids. This will introduce the ability to model the discrete medium fields into the existing codes with the minimal effort.

Introduction

One distinctive feature of TEXAS, which is the computer code for analyzing the FCI process, is that it treats the molten fuel as being a stream of the discrete particles (Lagrangian particles, Lagrangian field) that are traveling through and interacting with the fluid (Eulerian fluid, Eulerian field), which is the binary mixture of the liquid water and the steam. This feature allows TEXAS to flexibly and effectively simulate the mixing and the fragmentation of the molten fuel during the process. With its previous success and since TEXAS only simulates the process in one dimension, it has been considered that the concept the Lagrangian particles should be extended to two and three dimensions. One possible option in this development is to directly modified/rewrite TEXAS so that it is capable of conducting the simulation in three dimensions. This option has already been pursued with the co-operation between the university of Wisconsin-Madison and KAERI. The other path that is being conducted here at KAERI is to develop a set of subroutines for simulating the Lagrangian particles, which can be inserted into the existing codes. In effect, this will introduce the Lagrangian particles in to calculation with the minimal change in the original codes. It is the development of this Lagrangian modules, which is primarily called **LeSiM** (**L**agrangian **e**xtensible **Si**mulation **M**odules), that is of interest and will be discussed in this report.

Models and Algorithms

In general, the concept of the Lagrangian particles is the same as that used in TEXAS. However, some modifications have been implemented. The most obvious modification is the three dimensional movement of the particles. Some other modifications are to fix the limitation found in the original implementation. For example, since the variable that records the number of particles per master group is now of real type with double precision, the number of the particles contained in a master group is greatly increased.

Compared with having the Lagrangian model built in to the code like in TEXAS, **LeSiM** may seem to lost an advantage in that the calculation of the Lagrangian field will have to be done separately after the calculation of the Eulerian fields. As the property of the particles is not update simultaneously with that of the fluids, this may result in the error of the calculation or, at least, the lost of the accuracy. This concern is especially troublesome if the calculation is to be done iteratively. However, consider the complexity of the calculation due to the coupling of the Lagrangian and the Eulerian fields, this disadvantage may not be as serious as it is first looked. In any case, to reduce the error caused by the separate calculations, **LeSiM** employs the Runge-Kutta method to calculate the movement of the particles and their interaction with the surrounding fluids. Other problems that may arise from implementing **LeSiM** in a code are that the code may require much more resource to run and may take much longer time to finish. These are to be expected even if the Lagrangian model is inherently built in since the calculation of the Lagrangian particles will obviously need more memory to store the property of the particles and more time for the calculation. With **LeSiM**, extra time is also needed for passing the information between the original code and its modules. Still, this added extra time should not be significant compared with the actual time needed for the calculation.

In general, the models and algorithms employed in **LeSiM** can be descried as in the following sections.

Models for motion and interactions

The motion of a particle is described simply by Newton' s second law,

$$\vec{F} = -r_p V_p \vec{a}$$

where $\vec{F}$ is the force acting on a particle, r_p and V_p are the density of a particle and $\vec{a}$ is the acceleration of a particle and is related to is velocity $\vec{v}$ and position $\vec{r}$ as

$$\begin{aligned} \vec{a} &= \frac{d}{dt}\vec{v} \\ &= \frac{d^2}{dt^2}\vec{r} \end{aligned}.$$

For the acting force, it is the summation of the pull by the gravity $\vec{G}$ and the drag $\vec{D}$ caused by the relative motion between a particle and the surrounding fluid.

The gravity pull is described as

$$\vec{G} = -\hat{k}\, gV\left(r_p - r_f\right)$$

where $\hat{k}$ is the vertical unit vector in the up direction, g is the gravity acceleration and r_f is the density of the fluid mixture.

The drag is described as

$$\vec{D} = f_D\left(\vec{v}_f - \vec{v}_p\right)$$

where f_D is the drag coefficient. The actual value of f_D depends on the property of each phase in the fluid and the flow regime. Therefore, it must be re-evaluated at every time step.

The other models needed are the model for the heat transfer process, the model for the fragmentation of the particles and, since a large number of particles are presented by a single master group, a model to predict how a group of particles expands its volume of appearance.

In general, the heat transfer process is modeled with the Newton' s law of cooling, with Q being the rate at which the heat is transferred from a particle,

$$Q = Ah_f \Delta T_{pf}$$

where A, h_f and ΔT_{pf} are the surface area of a particle, the heat transfer coefficient and the temperature difference between that of the sphere and the surrounding fluid.

For the fragmentation of a particle, the fragmentation rate depends much on the Weber number, rv^2D/s where r_f, v_f, D_f and s_p are the density, flow speed and the hydraulic diameter of the fluid and the surface tension of the particle. For the expansion size of a particle group, which is not employed in TEXAS, there is no clear definition since it is not a physical parameter but more of a statistical value. Unless all the particles are tracked, which is impossible if the fragmentation is involved, calculating the exact expansion size of a group is infeasible. Still, approximating the expansion size of a group can still be done under some specific condition. Some approximation models such as the diffusion theory or the uniform expansion can be applied. For **LeSiM**, the models for the expansion size of a group is being studied and has not been decided.

Algorithms for calculation

LeSiM has been written in Fortran90. It is consist of there groups of modules plus one separate program. These three groups of modules are as described below.

1. The first group is for the header files that define the common variables to be used, the information regarding the particles. The volumetric cells that contain the particles and the information on the fluids in each cell are recorded by the variables defined in these files. The following files are the header files.
   1.1 “***particle.inc***,” this file defines the common variables such as the position and the velocity of each master group. The mechanical and thermal parameters of each group are also included.
   1.2 “***testprp.inc***,” this file defines the variables for controlling the injection of the particles and the activation of the models for the heat transfer process and the fragmentation of the particles.
2. The second group is for the subroutines and the functions that control the flow of the information. The input and output process are also listed in this group. Each of these subroutines is in the following files.
   2.1 “***main.f90***,” this file contains the main subroutine (program) that control the flow of the simulation.
   2.2 “***readinput.f90***,” it contains the subroutine that read the information from a file. It can be also used to pass the information from the other subroutine if **LeSiM** is to be inserted and compiled together with the other code.
   2.3 “***openfile.f90***,” the subroutine in this file opens the data files for output.
   2.4 “*prn_rpt.f90*,” the subroutine in this file print the result from the simulation of LeSiM to the data files.
   2.5 “***pinject.f90***,” the subroutine in this file controls the injection of the particles in to the system for the simulation.
   2.6 “***chk_cll.f90***,” this file is the subroutine for checking the condition of the volumetric cells that contain the particles.
   2.7 “***chk_info.f90***,” in this file is the subroutine that check if the information is passed correctly.
   2.8 “***cpy2p0.f90***,” in this file is the subroutine that copies the variables for backup and for further calculation.
   2.9 “***closefile.f90***,” in this file is the subroutine that closes the data files.
   2.10 “***err_msg.f90***,” in this file is the subroutine for displaying the error message.
   2.11 “***pncell.f90***,” this file defines the function that checks whether a given particle is in the given cell.
   2.12 “***cpypp.f90***,” this file defines the subroutine for copying the variables.
3. The third group is for the subroutines, also the functions, that perform the actual simulation. They are contained in the following files.
   3.1 “***accel.f90***,” this is the file for the function that calculates the acceleration of a particle.
   3.2 “***accl_1.f90***,” this is the file for the function that calculate the acceleration of a particle along the first major axis. It is called from the function in “***accel.f90***.”
   3.3 “***accl_2.f90***,” this is the file for the function that calculate the acceleration of a particle along the second major axis. It is called from the function in “***accel.f90***.”
   3.4 “***accl_3.f90***,” this is the file for the function that calculate the acceleration of a particle along the third major axis. It is called from the function in “***accel.f90***.”
   3.5 “***chng_frate.f90***,” the file is for the subroutine that calculates the changing rate for the number of the particles in a group due to the fragmentation. If the change caused by the recombination of the particles must be modeled, it can be done with this subroutine.
   3.6 “***chng_rate.f90***,” the file is for the subroutine that calculates the changing rate for the number of the particles in a group with out the fragmentation. Even though the number of the particles is not expected to be increased without the fragmentation, it is still included for completeness. However, if the change caused by the recombination of the particles must be modeled, it can be done with this subroutine.

3.7 "***chr_frate.f90***," the file is for the subroutine that calculates the changing rate for the radius of a particle in a group due to the fragmentation. If the change caused by the recombination of the particles must be modeled, it can be done with this subroutine.

3.8 "***chr_frate.f90***," the file is for the subroutine that calculates the changing rate for the radius of a particle in a group without the fragmentation. If the change caused by the recombination of the particles must be modeled, it can be done with this subroutine.

3.9 "***chrg_frate.f90***," the file is for the subroutine that calculates the expansion rate for a particle group due to the fragmentation. If the change caused by the recombination of the particles must be modeled, it can be done with this subroutine.

3.10 "***chrg_frate.f90***," the file is for the subroutine that calculates the expansion rate for a particle group without the fragmentation. If the change caused by the recombination of the particles must be modeled, it can be done with this subroutine.

3.11 "***chtp_frate.f90***," the file is for the subroutine that calculates the changing rate for the temperature of a particle in a group, taken in to account the effect of the fragmentation. If the change caused by the recombination of the particles must be modeled, it can be done with this subroutine.

3.12 "***chtp_frate.f90***," the file is for the subroutine that calculates the changing rate for the temperature of a particle in a group, without the effect of the fragmentation. If the change caused by the recombination of the particles must be modeled, it can be done with this subroutine.

3.13 "***drag.f90***," this file is for the function that calculates the acceleration caused by the drag.

3.14 "***flwmp.f90***," this file is for the function that calculates the flow regime.

The additional program is also developed to help in presenting the result obtained from the simulation with **LeSiM**. This program runs separately and is not a crucial part of **LeSiM**. It may be excluded if the graphical presentation is not needed.

Calculations and Results

For the testing purpose, a number of calculations were conducted with **LeSiM** alone, without the interaction with the surrounding fluid, except the drag induced by the motion of the particles. The graphical results, also obtained with the separate program from **LeSiM**, from four of such calculations are given here to present the functionality of **LeSiM**. No fragmentation and heat transfer process were implemented in these results. For the system, while it may not be observable from the given figures, the system was composed of two parts. The upper part of the system was filled with the steam while the lower part was filled with the liquid water. This was done so test if the motion of the particles across the volumetric cells could be simulated successfully.

The results from four simulations are as given in Figure 1, 2, 3 and 4. Figure 1 shows the graphical display of an instance in which only one particle was injected into the system. In Figure 2, the calculation was for the case where 20 particles were injected periodically and systematically in to the system. In Figure 3, 20 particles were injected randomly, both in time and along the horizontal position, in to the system. Unlike the first 3 results whose the particle(s) has the initial velocity of zero, the fourth result was for the same system but with only five particles that were injected randomly, in time and space. Each particle also has random initial velocity. The graphical display of the result as obtained is shown in Figure 4.

In all cases, the simulations were found to run very smoothly and were finished without any difficulty. The accuracy of the results were, however, difficult to evaluate. It is expected that benchmarking with the data from the experiments will be performed. An additional calculation was conducted for the case where the cell was fully voided. In such case, the particles were assumed to conserve their energy. This was observed to be true, under the condition that the numerical round-off and truncation error were considered. Therefore, the algorithms as used in **LeSiM** are at least numerically correct. With the benchmarking, the structure of the modules and the models as used will be validated. When and if these are successfully tested, the next stage of the development is to apply **LeSiM** with the other fluid based codes for actual calculation of the FCI process.

Conclusions

The first stage of the development of **LeSiM** (Lagrangian extensible Simulation Modules) to be used with the existing fluid based code for the calculation of the FCI process has been completed. The modules is considered to perform as expected in the test calculations. The modules are then expected to be tested, benchmarked and modified based on the data from the experiments. This is the second stage of the development. Once they are successfully tested and modified, the modules is to be applied on the existing codes in order to actually simulate the FCI process. The implementation of the modules is considered the third stage of the whole development. while the testing and the actual simulation of the FCI experiments are the fourth and the final stage. With the speed at which the development has been progressing, the second and the third stage are expected to be finished within the next ten months. The definite time frame of the fourth stage has not been set. It was believed that if the modification the finished code is not too highly complicate, the fourth stage should be finished soon after the third stage is completed.

[*] Assistant Professor in Nuclear Engineering, Department of Nuclear Technology, Faculty of Engineering, Chulalongkorn University, Bangkok, Thailand (email:fntsns@eng.chula.ac.th)

References

1. Cho-Chone Chu, Michael L. Corradini, Ilpo Huhtiniemi, John Murphy, Jian Tang, Sunchai Nilsuwankosit, **A Code Manual for TEXAS-V: One Dimensional Transient Fluid Model Fuel-Coolant Interaction Analysis**, Nuclear Engineering and Engineering Physics, University of Wisconsin-Madison, 1996
2. M.L. Corradini, J. Murphy, S. Nilsuwankosit, D. Pineau, B. Shamoun, J. Tang, and S. Wang, **Fuel-Coolant Interaction Analyses with TEXAS-V Vapor Explosion Model**, Nuclear Safety Research Center, Department of Engineering Physics, University of Wisconsin, USA, 1998.
3. William H. Press, Saul A. Teukolsky, William T. Vetterling, Brian P. Flannery, **Numerical Recipes in Fortran**, Cambridge University Press, 1992

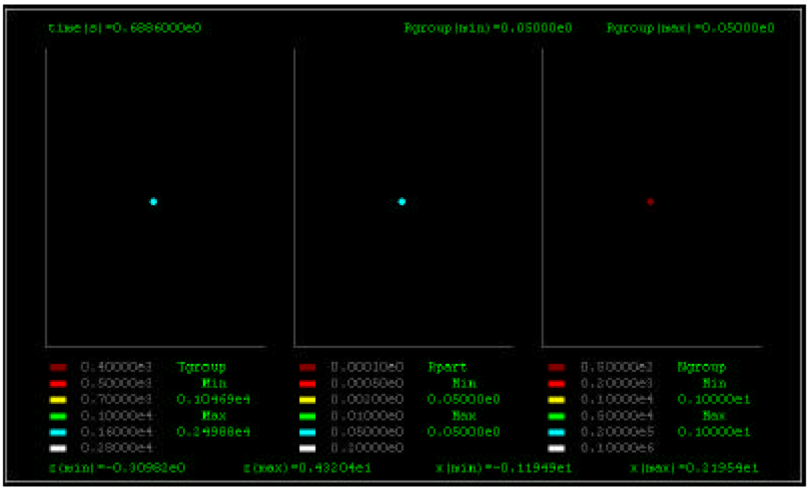

Figure 1 LeSiM calculation with only one particle dropped

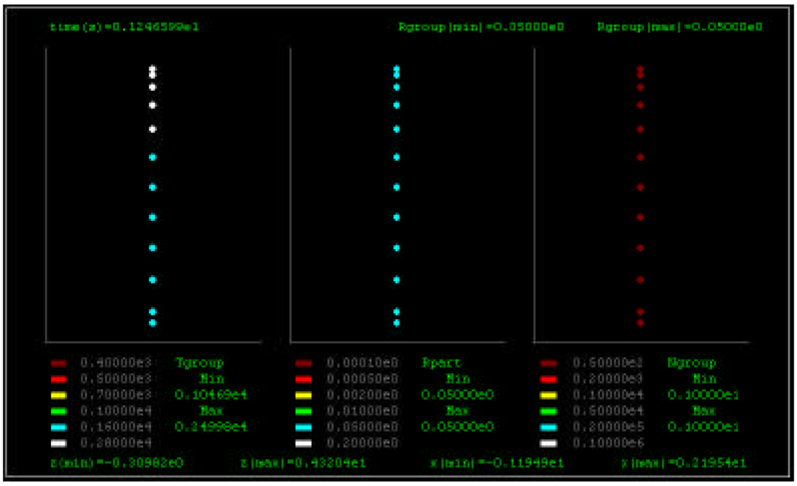

Figure 2 LeSiM calculation with only 20 particle, uniformly dropped

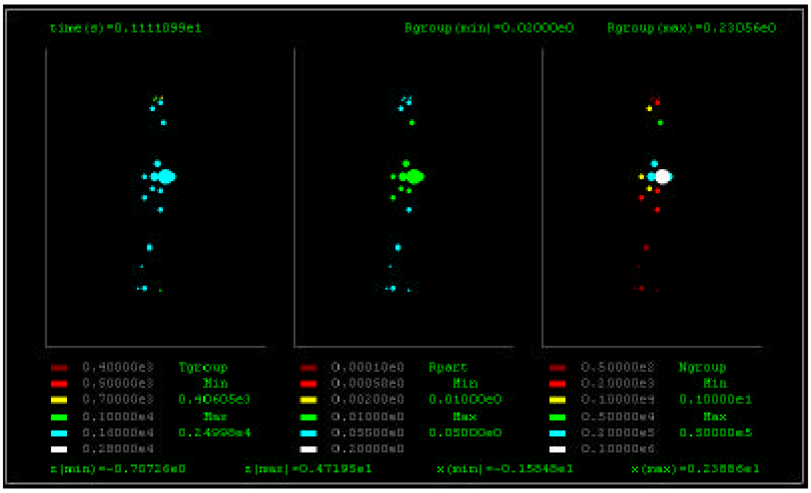

Figure 3 LeSiM calculation with only 20 particle, randomly dropped

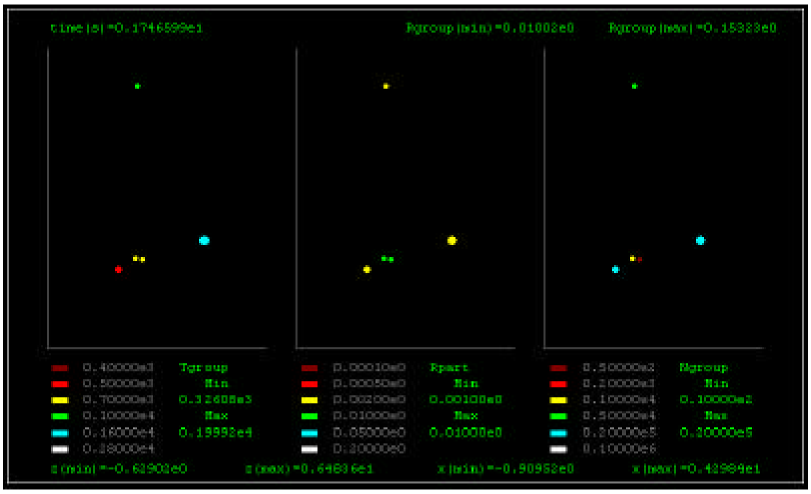

Figure 4 LeSiM calculation with only 5 particle, randomly injected