Journal of the Korean Nuclear Society
Volume 31, Number 3, pp.344~363, June 1999

A Multi-Dimensional Thermal-Hydraulic System Analysis Code, MARS 1.3.1

Jae-Jun Jeong, Kwi Seok Ha, Bub Dong Chung, and Won Jae Lee
Korea Atomic Energy Research Institute
150 Dukjin-dong, Yusong-gu, Taejon 305-353, Korea
wjlee@nanum.kaeri.re.kr

(Received December 31, 1998)

A multi-dimensional thermal-hydraulic system analysis code, MARS 1.3.1, has been developed in order to have the realistic analysis capability of two-phase thermal-hydraulic transients for pressurized water reactor (PWR) plants. As the backbones for the MARS code, the RELAP5/MOD3.2.1.2 and COBRA-TF codes were adopted in order to take advantages of the very general, versatile features of RELAP5 and the realistic three-dimensional hydrodynamic module of COBRA-TF. In the MARS code, all the functional modules of the two codes were unified into a single code first. Then, the source codes were converted into the standard Fortran 90, and then they were restructured using a modular data structure based on "derived type variables" and a new "dynamic memory allocation" scheme. In addition, the Windows features were implemented to improve user friendliness. This paper presents the developmental works of the MARS version 1.3.1 including the hydrodynamic model unification, the heat structure coupling, the code restructuring and modernization, and their verifications.

Key Words : multi-dimensional system code, MARS, RELAP5, COBRA-TF, restructuring, modernization.

1. Introduction

The RELAP5 [1] and COBRA-TF [2] codes had been introduced to KAERI through the international cooperation programs [3] in 1983. The RELAP5 code is a versatile and robust code based on a one-dimensional two-fluid model for two-phase flows. The code includes many generic component models and special process models, which enable to simulate most of transients and postulated accidents. The COBRA-TF code, developed to predict nuclear fuel channel behavior during a loss-of-coolant accident (LOCA), adopts a three-dimensional two-fluid, three-field model for two-phase flow. The three-dimensional feature permits extremely flexible noding of the reactor vessel and, thus, more realistic simulations. As an effort to develop a best-estimate system analysis code, Lee et al. coupled the COBRA-TF and RELAP5/MOD3 5m5 codes in 1992 [4]. The resulting code was named COBRA/RELAP5. This effort to link the existing codes was very reasonable and economical to take the advantage of the existing capabilities of the verified codes, yielding a

new code with an extended scope of applications.

A more systematic and extensive code development program was commenced in 1996 at KAERI. The objective of the program is to develop a best-estimate multi-dimensional system analysis code that has a coupled capability of three-dimensional neutron kinetics, containment analysis, and critical heat flux calculations. As a backbone for code development, the COBRA/ RELAP5 code was selected in order to fully utilize the very general, versatile features of RELAP5, domestic user's accumulated experience in RELAP5, and the realistic three-dimensional reactor vessel module of COBRA-TF. Then, RELAP5/MOD3 5m5 of the COBRA/RELAP5 code was replaced with RELAP5/MOD3.2.1.2 (hereinafter RELAP5). In order to enhance the code portability, the programming language of COBRA/RELAP5 was converted into the standard Fortran 90. Also, for better user access, the hydrodynamic calculation sequence was modified from a process-level parallel-computation mode [4] to a serial-computation mode [5]. The resulting unified code was renamed as MARS (Multi-dimensional Analysis of Reactor Safety). Further unification of code models and the restructuring and modernization have been carried out in order to have enhanced code readability, maintainability, flexibility, portability, and user friendliness [6,7]. The MARS code has been evolved to the current version, MARS 1.3.1, in which RELAP5 and COBRA-TF are consolidated as the 1D module and the 3D module, respectively.

This paper provides a summary and review of the developmental activities up to the MARS version 1.3.1 [6,7]. The code development requirement is introduced in Sec. 2. Because MARS is a unified code based on the RELAP5 and COBRA-TF code, descriptions in Sec. 2 through 5 are mainly focused on the unification requirements, methods and verifications. These include the unification of hydrodynamic models, heat structure models, and point kinetics model, etc. Section 6 presents the restructuring and modernization process, which is one of the distinguished developmental activities and permits a unique feature of the MARS code. In Sec. 7, the verifications of the developmental works are described. The last section gives the summary and conclusions. The results of overall code assessment are provided in Ref. 6.

2. Code Development Strategy and Progress

The first step of the MARS code development is to unify the COBRA-TF and RELAP5 codes, while maintaining the excellent features of the two codes. The basic guideline of the code unification is that the inherent features of each code should be maintained and improved.

Both codes have their own integral structure, that is, each code consists of the following functional routines: (i) input processing routines, (ii) initialization routines, (iii) transient calculation routines for reactor kinetics, hydrodynamics, heat structure, (iv) output processing routines including restart and strip functions, and (v) basic routines for thermodynamic and transport properties. Thus, the scopes of the code unification are open to all the functional routines of the two codes. In terms of "code integrity" and "user friendliness", the following requirements were observed during the code unification:

(i) The MARS code should work as a "single code" consisting of RELAP5 and COBRA-TF modules (hereinafter called 1D and 3D modules, respectively), running on a computer with a single CPU.

(ii) The conservation laws should not be violated through the unification process, i.e., the mass, energy, and momentum should be conserved

across the 1D/3D interfaces.

(iii) The numerical stability should be guaranteed within the Courant limit.

(iv) The 3D module can be optionally used. Multiple use of the 3D module is permitted.

(v) The unified MARS code uses a single input file. The input forms of the 1D and 3D modules should be consistent.

In recognition of the above requirements, all the functional routines of the two codes are compared and classified into four groups:

(i) Routines that have identical functions are replaced with better ones. For instance, the input structure of RELAP5 is more convenient than that of COBRA-TF. Thus, the input processing routines of COBRA-TF are modified.

(ii) Routines that have their own advantage are maintained/linked so that the users can utilize both modules. For example, RELAP5 has various useful hydrodynamic component models, whereas COBRA-TF has a realistic 3D hydrodynamic model. Then, these are integrated so that the users can use both of them.

(iii) Routines that are provided only in a code are modified for shared use in the other code. An example is that, in COBRA-TF, the reactor kinetics model is not provided. So, the RELAP5 kinetics model is modified/implemented into COBRA-TF.

(iv) Some modules should be unified for mathematical consistency. These include the thermodynamic and transport properties of water. In this case, the COBRA-TF routines were replaced with those of RELAP5.

The second step is the restructuring of the unified MARS code. The basic structures of RELAP5 and COBRA-TF were developed in the 1970s and therefore they did not take advantage of today's abundant supply of inexpensive, fast memory [8]. In addition, older programming language did not provide a means for simple dynamic memory allocation. As a result, elegant programming styles, such as "bit packing" and "big container arrays", were invoked to overcome these limitations. Unfortunately, these techniques produced cryptic coding and compromised readability, maintainability and portability. To resolve these problems, all the routines were converted into the standard Fortran 90 and, then, the restructuring was conducted, using a modular data structure based on "derived type variables" and a new dynamic memory allocation scheme [9]. The resulting source programs are easy to understand and modify. The first two steps were completed in 1998.

The third step, a major part of the MARS code development program under progress, aims at both the improvement of the existing capabilities through a systematic code assessment program and the code couplings for three-dimensional neutron kinetics, containment analysis, and critical heat flux calculations. During this step, the code modernization is also to be performed. To facilitate the use of the MARS code, a graphical user interface (GUI) is to be established. Currently a simple Windows GUI function is implemented so that x-y plots of major variables are displayed on-line during the calculation.

3. Hydrodynamic Models

The MARS hydrodynamic module consists of the RELAP5 and COBRA-TF hydrodynamic models. The flow system simulated by MARS is divided into three-dimensional COBRA-TF regions (3D regions) and one-dimensional RELAP5 regions (1D regions). For instance, the reactor vessel of a PWR can be a 3D region, and the primary loops and the secondary systems become 1D regions. This is a form of the "domain

decomposition" method. The hydrodynamic behavior of each subdomain is modeled by either the 1D or 3D module, however, the system pressure equations of the hydrodynamic models are implicitly coupled and solved simultaneously (see Sec. 3.3).

3.1. Conservation Equations and Constitutive Relationships

The 1D hydrodynamic module employs a one-dimensional, transient, two-fluid model for two-phase flows [1]. The two-fluid equations consist of the following equations:

- Continuity equation of vapor phase:

$$\frac{\partial}{\partial t}(\alpha_g\rho_g)+\frac{1}{A}\frac{\partial}{\partial z}(\alpha_g\rho_g V_g A)=\Gamma_g, \tag{1}$$

- Continuity equation of liquid phase:

$$\frac{\partial}{\partial t}(\alpha_f\rho_f)+\frac{1}{A}\frac{\partial}{\partial z}(\alpha_f\rho_f V_f A)=\Gamma_f. \tag{2}$$

where $\alpha_f = 1 - \alpha_g$.

- Momentum equation of vapor phase:

$$\begin{aligned}\alpha_g\rho_g A\frac{\partial V_g}{\partial t}+\frac{1}{2}\alpha_g\rho_g A\frac{\partial V_g^2}{\partial z}=&-\alpha_g A\frac{\partial P}{\partial z}+\alpha_g\rho_g B_x A-\alpha_g\rho_g A F_{w,g}V_g\\&+\Gamma_g A(V_{ig}-V_g)-\alpha_g\rho_g A F_{i,g}(V_g-V_f)\\&-C\alpha_g\alpha_f\rho_m A\left[\frac{\partial(V_g-V_f)}{\partial t}+V_f\frac{\partial V_g}{\partial z}-V_g\frac{\partial V_f}{\partial z}\right]\end{aligned} \tag{3}$$

- Momentum equation of liquid phase:

$$\begin{aligned}\alpha_f\rho_f\frac{\partial V_f}{\partial t}+\frac{1}{2}\alpha_f\rho_f A\frac{\partial V_f^2}{\partial z}=&-\alpha_f A\frac{\partial P}{\partial z}+\alpha_f\rho_f B_x A-\alpha_f\rho_f A F_{w,f}V_f\\&-\Gamma_g A(V_{if}-V_f)-\alpha_f\rho_f A F_{i,f}(V_f-V_g)\\&-C\alpha_g\alpha_f\rho_m A\left[\frac{\partial(V_f-V_g)}{\partial t}+V_g\frac{\partial V_f}{\partial z}-V_f\frac{\partial V_g}{\partial z}\right]\end{aligned} \tag{4}$$

- Energy equation of vapor phase:

$$\begin{aligned}\frac{\partial}{\partial t}(\alpha_g\rho_g U_g)+\frac{1}{A}\frac{\partial}{\partial z}(\alpha_g\rho_g U_g V_g A)=&-P\frac{\partial\alpha_g}{\partial t}-\frac{P}{A}\frac{\partial}{\partial z}(\alpha_g V_g A)+Q_{wg}+Q_{ig}\\&+\Gamma_g h_g^*+\Gamma_w h_g'+Q_{DISS,g}\end{aligned} \tag{5}$$

where $h_g^*=\begin{cases}h_g^s,\Gamma_g\geq 0\\ h_g,\Gamma_g<0\end{cases}$, and $h_g'=\begin{cases}h_g^s,\Gamma_w\geq 0\\ h_g,\Gamma_w<0\end{cases}$.

- Energy equation of vapor phase:

$$\begin{aligned}\frac{\partial}{\partial t}(\alpha_f\rho_f U_f)+\frac{1}{A}\frac{\partial}{\partial z}(\alpha_f\rho_f U_f V_f A)=&-P\frac{\partial\alpha_f}{\partial t}-\frac{P}{A}\frac{\partial}{\partial z}(\alpha_f V_f A)+Q_{wf}+Q_{if}\\&-\Gamma_g h_f^*-\Gamma_w h_f'+Q_{DISS,f}\end{aligned} \tag{6}$$

where $h_f^*=\begin{cases}h_f,\Gamma_g\geq 0\\ h_f^s,\Gamma_g<0\end{cases}$, and $h_f'=\begin{cases}h_f,\Gamma_w\geq 0\\ h_f^s,\Gamma_w<0\end{cases}$

- Continuity Equation of noncondensible gases.

$$\frac{\partial}{\partial t}(\alpha_g\rho_g X_n)+\frac{1}{A}\frac{\partial}{\partial z}(\alpha_g\rho_g V_g X_n A)=0. \tag{7}$$

The 3D module adopts a three-dimensional, two-fluid, three-field model on rectangular Cartesian or sub-channel coordinates [2].

- Four continuity equations for vapor, continuous liquid, entrained liquid droplets, and non-condensible gases:

$$\frac{\partial}{\partial t}(\alpha_k\rho_k)+\nabla\cdot(\alpha_k\rho_k\vec{V}_k)=\Gamma_k \tag{8}$$

where k=l, v, e, or n for continuous liquid, vapor, entrained liquid, and noncondensible gases, respectively. $\Gamma_k = 0$ for noncondensible gases.

- Three momentum equations for continuous liquid, entrained liquid, and the mixture of vapor and noncondensible gases (it is assumed that vapor and noncondensible gases are in mechanical equilibrium).

$$\frac{\partial}{\partial t}(\alpha_k\rho_k\vec{V}_k)+\nabla\cdot(\alpha_k\rho_k\vec{V}_k\vec{V}_k)=\alpha_k\rho_k\vec{g}-\alpha_k\nabla P+\nabla\cdot[\alpha_k(\vec{\tau}_k+\vec{T}_k)]+\vec{M}_k^{\Gamma}+\vec{M}_k^{d} \tag{9}$$

where k=l, v+n, or e for continuous liquid, gaseous phase, entrained liquid, respectively.

- Two energy equations for the mixture of vapor and noncondensible gases (it is assumed that vapor and noncondensible gases are in thermal equilibrium), and the mixture of continuous and entrained liquid (it is assumed that continuous and entrained liquid are in thermal equilibrium).

$$\frac{\partial}{\partial t}(\alpha_k\rho_k h_k)+\nabla\cdot(\alpha_k\rho_k h_k\vec{V}_k)=-\nabla\cdot[\alpha_k(\vec{Q}_k+\vec{q}_k^{T})]+\Gamma_k h_k'+q_{ik}^{-}+\alpha_k\frac{\partial P}{\partial t} \tag{10}$$

where $k=l+e$, or $v+n$ for liquid and gaseous phase, respectively.

For closure of the system of equations in the two modules, constitutive relations for various flow regimes are incorporated in each module. These include the state-of-the-art physical models for the interfacial mass transfer, the interfacial forces, the wall drag, and the wall and interfacial heat transfer. The thermodynamic and transport properties of water are also included. In the 3D module, the rate of entrainment/deposition and a vapor/droplet interfacial area transport equation are additionally included. Detailed description of the constitutive relations are provided in Refs. 1 and 2.

Comparing the 1D and the 3D hydrodynamic modules, the 1D module does not separately model the entrained liquid droplet field, but models the nonvolatile component such as boron. Thus, the following assumptions apply for the unification of the hydrodynamic models:

- Nonvolatile components do not exist in the flow system.
- The entrained liquid and continuous liquid flowing from the 3D to the 1D region are agglomerated into the liquid phase in the 1D region. The liquid phase flowing from the 1D to the 3D region is divided into continuous liquid phase and droplets in the 3D region according a film/droplets partition rule.
- The vapor phase and noncondensible gases of the two regions are treated as continuum across the regions.

3.2. Numerical Solution Scheme

The numerical solution schemes for the 1D and 3D hydrodynamic models are basically the same although the calculational sequence is slightly different. Both of them use a semi-implicit, finite-difference method based on a staggered-grid mesh and donor cell scheme.

In both the 1D and 3D modules, the phasic momentum equations are solved first to represent the phasic velocity (phasic mass flow rate in the 3D module) at junction j in terms of pressures of the adjoining cells K and L:

$$V_{k,j}^{n+1} = \alpha_{k,j} + \beta_{k,j}(\delta P_K - \delta P_L), \tag{11}$$

where $\delta P = P^{n+1} - P^n$,

α, β : Coefficients calculated from old time step variables.

$$k = \begin{cases} f \text{ or } g \text{ in RELAP 5} \\ l, v \text{ or } e \text{ in COBRA-TF} \end{cases}$$

Next, the finite difference equations (FDEs) of the mass and energy equations for a hydrodynamic cell are linearized with respect to the independent scalar variables. The FDEs are ordered in the sequence shown in Table 1 and are rearranged as follows:

The 1D module (Ref. 1, vol. 1, p. 3.32)

$$\begin{bmatrix} r_{11} & r_{12} \cdots r_{15} \\ r_{21} & \cdots r_{25} \\ r_{31} & \cdots r_{35} \\ r_{41} & \cdots r_{45} \\ r_{51} & \cdots r_{55} \end{bmatrix} \begin{bmatrix} \delta\tilde{X}_n \\ \delta\tilde{U}_g \\ \delta\tilde{U}_f \\ \delta\tilde{\alpha}_g \\ \delta P \end{bmatrix} = \underline{s} + \underline{g}^1 V_{g,j+1}^{n+1} + \underline{f}^1 V_{f,j+1}^{n+1} + \underline{g}^2 V_{g,j}^{n+1} + \underline{f}^2 V_{f,j}^{n+1}, \tag{12a}$$

where $\underline{s}$, $\underline{g}^1$, $\underline{f}^1$, $\underline{g}^2$, and $\underline{f}^2$ are coefficient vectors.

The 3D module (Ref. 2, vol. 2, p. 4.4)

$$\begin{bmatrix} c_{11} & c_{21} & \cdots c_{16} \cdots c_{16+NC} \\ c_{21} & & \cdots c_{26} \cdots c_{26+NC} \\ c_{31} & & \cdots c_{36} \cdots c_{36+NC} \\ c_{41} & & \cdots c_{46} \cdots c_{46+NC} \\ c_{51} & & \cdots c_{56} \cdots c_{56+NC} \\ c_{61} & & \cdots c_{66} \cdots c_{66+NC} \end{bmatrix} \begin{bmatrix} \delta(\alpha_v P_g) \\ \delta\alpha_v \\ \delta(\alpha_v h_v) \\ \delta\{(1-\alpha_v)h_l\} \\ \delta\alpha_e \\ \delta P_J \\ \delta P_{J,1} \\ \cdots \\ \delta P_{J,NC} \end{bmatrix} = \begin{bmatrix} E_{CN} \\ E_{CL} \\ E_{EV} \\ E_{EL} \\ E_{CE} \\ E_{CV} \end{bmatrix}, \tag{12b}$$

where subscript NC is the number of adjoining

cells to cell *J*. In Eq. (12b), the unknown phasic mass flow rates are already substituted with Eq. (11).

Multiplying Eq. (12a) by the inverse of the cell Jacobian matrix, the bottom row results in a single equation involving just pressures, where the unknown velocities in the right-hand side (RHS) are replaced with Eq. (11). Eq. (12b) also can be reduced using the Gaussian elimination, of which the bottom row results in an equation involving just pressures. This is done for each cell and, at last, one can set up the so-called system pressure matrix equation:

$$\underline{\underline{A}}\,\delta\underline{P} = \underline{b}, \tag{13}$$

where $\underline{\underline{A}}$ is an N ×N matrix (*N* is the number of computational cells).

The pressure variations are obtained by solving Eq. (13), and substituting them into Eq. (11) yields the new time-step velocities (or mass flow rates). Further back-substitutions are done to obtain the variations of other independent variables. After then, the remaining numerical procedures are performed until the completion of a time-step calculation.

3.3. Integration of the System Pressure Matrices

Consider a flow system that is divided into a 3D region and several 1D regions with *NR* interface junctions. For convenience, let us define *Ci* and *Ri* as the index numbers of *i-th* interfacing cells in the 3D and 1D regions, respectively (see Fig. 1). Basic concepts and assumptions for the integration are

- The momentum equations at the interface junction are solved by the 1D module.
- It is assumed that continuous liquid and entrained droplets at the interface are in mechanical equilibrium, .

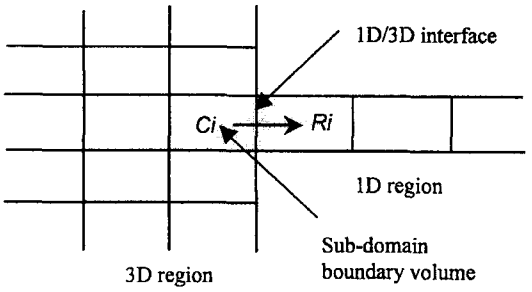

Fig. 1. Typical 1D/3D Connection in the MARS Code.

- In the 1D module, cell *Ci* is treated as a **"*sub-domain boundary volume (sdbvol)*"**, whose scalar variables except for pressure are updated every time-step by the 3D module (Because the cell *Ci* is not a real boundary volume, but a pseudo boundary volume, this component was named as a **"*sub-domain boundary volume.*"** The **"sdbvol"** is a new component of the 1D module which is used to represent cell *Ci* [6]). When the flow enters cell *Ri* from cell *Ci*, continuous liquid and droplets are agglomerated into the liquid phase at the 1D/3D interface.
- In the 3D module, cell Ri is regarded as a sink that is implicitly coupled. When liquid enters cell Ci from cell Ri, the liquid phase is divided into continuous liquid and droplets at the 1D/3D interface according to a partition rule[6]. The rule directly uses some correlations for **"**entrainment fraction**"** calculation in annular-mist flow regime.
- The system pressure matrices, which are set up in each module, are coupled via the momentum modeling at the interfaces and solved simultaneously.

The momentum balance at the interface junction from cell *Ci* to *Ri* is modeled by the 1D module, where the old time-step variables of cell *Ci* are transferred from the 3D module. Then, the phasic velocity at the *i-th* interface junction is given by

$$V_{k,i}^{n+1} = \alpha_{k,i} + \beta_{k,i}(\delta P_{Ci} - \delta P_{Ri}) \quad (14)$$

where $k=f$ or g

Effects of the connections should be taken into account for the conservation of momentum in the 3D region. For simplicity, it is assumed that cell Ri is normal to the interface surface and that the connection is either vertical or transverse. When cell Ci is a fully three-dimensional cell, the five momentum cells in the 3D module adjacent to the interface are influenced by the connection of cell Ri.

Because of the 1D/3D connections, the system pressure matrix equation of the 1D module involves additional unknown terms that include the unknown velocities at the 1D/3D interfaces [see Eq. (13) and Fig. 2(b)] :

$$\underline{\underline{A}}_R \delta \underline{P}_R = \underline{b}_R + \sum_{i=1}^{NR} (\underline{\gamma}_{f,i} V_{f,i}^{n+1} + \underline{\gamma}_{g,i} V_{g,i}^{n+1}) \quad (15a)$$

where $\underline{\gamma_f}$ and $\underline{\gamma_g}$ are coefficient vectors, all of whose elements except one are zeros. Inserting Eq. (14) into Eq. (15a) and rearranging yields

$$\underline{\underline{A}}'_R \delta \underline{P}_R = \underline{b}'_R + \sum_{i=1}^{NR} \underline{\gamma}'_i \delta P_{Ci}, \quad (15b)$$

Equation (15b) can be written as follows:

$$\delta \underline{P}_R = \underline{\underline{A}}'^{-1}_R \underline{b}'_R + \sum_{i=1}^{NR} (\underline{\underline{A}}'^{-1}_R \underline{\gamma}'_i) \delta P_{Ci} \quad (16)$$

Since is also an element of , can be written as

$$\delta P_{Ri} = \xi_i + \sum_{j=1}^{NR} \eta_{i,j} \delta P_{Cj} \quad (17)$$

Then, equation (14) is represented in terms of ;

$$V_{k,i}^{n+1} = \alpha_{k,i} - \beta_{k,i} \xi_i + \beta_{k,i} (\delta P_{Ci} - \sum_{j=1}^{NR} \eta_{i,j} \delta P_{Cj}). \quad (18)$$

Likewise, the system pressure equation of the 3D region should be also changed due to the 1D/3D

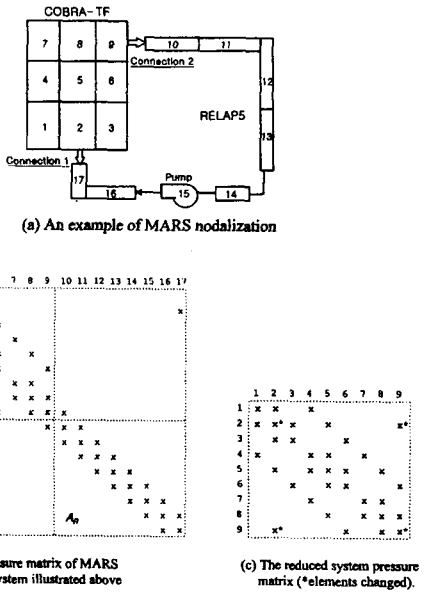

Fig. 2. An Example of MARS Nodalization and the Corresponding System Pressure Matrix.

connection [see Eqs. (13) and (15a)]:

$$\underline{\underline{A}}_C \delta \underline{P}_C = \underline{b}_C + \sum_{i=1}^{NR} (\underline{\kappa}_{f,i} V_{f,i}^{n+1} + \underline{\kappa}_{g,i} V_{g,i}^{n+1}) \quad (19)$$

Inserting Eq. (18) into Eq. (19) and rearranging results in a reduced, integrated system pressure matrix equation. The new time-step pressures in the 3D region are obtained by solving the reduced system pressure matrix equation. Fig. 2(c) shows an example of the reduced matrix.

The phasic velocities at the interface junctions are then obtained by back-substitutions of Eq. (18). The new time-step pressures in the 1D regions are obtained from Eq. (16). The remaining numerical sequences are performed in each module.

3.4. Thermodynamic and Transport Properties of Water

In the earlier versions of MARS, the 1D and 3D

modules calculated the light water properties separately. These needed to be unified for mathematical consistency. The existing property routines of the 3D module were completely modified to use the 1D module routines. Also, the routines for transport properties in the 1D and 3D modules, such as routines for surface tension, thermal conductivity, and viscosity, were replaced with those from the '92 ASME steam tables [10].

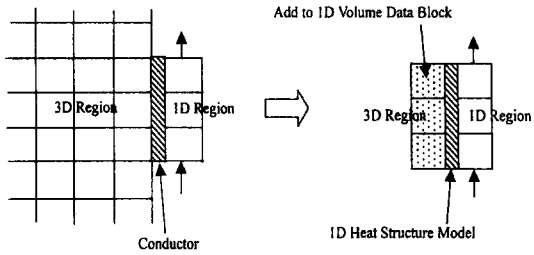

Fig. 3. The Concept of the 1D/3D Heat Structure Coupling.

4. Heat Structure Models

Both the 1D and 3D modules provide the heat structure model. This enables the calculation of the heat transferred across solid boundaries of hydrodynamic volumes. Modeling capabilities are general and include fuel pin or electrically heated rod, heat transfer across steam generator tubes, and heat transfer from pipe and vessel walls. In the two modules, heat structures are assumed to be represented by one-dimensional heat conduction equation. The equation is solved by a first-order finite-difference method. A difference in the heat structure models is that, in the 3D module, heat structures are divided into "rod" and "unheated conductor" depending on whether there is a heat source within the heat structure.

For nuclear fuel rods, additional models are involved in the two modules, such as models for gap conductance, fuel deformation, metal water reaction, and reflood heat transfer. In the reflood model, a two-dimensional (r-& z-direction) conduction equation is used and a fine mesh-rezoning scheme is implemented to efficiently use the two-dimensional conduction solution [1, 2]. The scheme is intended to resolve the large axial variation of wall temperatures and heat fluxes. The number of axial nodes in the heat structures is varied in such a way that the fine nodes exist only in the nucleate boiling and transition boiling regions. Currently the reflood model of the 1D module is not operational, but that of the 3D module works well.

As boundary conditions applied to the exterior surface, the followings can be used; (i) insulated condition, (ii) convective boundary condition, (iii) heat flux condition, and (iv) surface temperature condition. In the case of the convective boundary condition, when the 1D module heat structure is used, only the hydrodynamic volume of the 1D module can be specified. Thus, it is not possible to simulate a heat structure, of which one side (or the left boundary surface) contacts the 1D module hydrodynamic volumes and the other side (or the right boundary surface) contacts the 3D module hydrodynamic volumes. Figure 3 shows an example. To resolve this problem, we modified the heat structure model of the 1D module so that it can simulate the heat structures contacting with either the 1D or 3D module hydrodynamic volumes. This is called "heat structure coupling."

The "heat structure coupling" permits a thermal coupling between the 1D and the 3D hydrodynamic volumes and, in addition, the 1D heat structure model can simulate a heat structure, whose left and right boundaries contact the 3D hydrodynamic volumes. Therefore, the "unheated conductor" model of the 3D module can be

replaced or eliminated.

5. Reactor Kinetics Model

In the 1D module of MARS, the reactor power is computed using the point kinetics model with reactivity feedback, which assumes that power can be separated into space and time functions. This model is adequate for cases in which space distribution remains nearly constant. The point kinetics model computes both the immediate fission power and the power from the decay of fission products.

In the case of the 3D module, there are no reactor kinetics models, so direct simulation of nuclear reactor kinetics is not possible. Instead, users should provide the power shape and its evolution in a tabular form. However, using the "heat structure coupling" feature, the point kinetics model and the heat structure model of the 1D module can be used in the 3D module. In this case, fuel rod behavior during reflood period cannot be simulated because of the limitation of the 1D heat structure model. To resolve the problem, the point kinetics model in the 1D module was modified so that the feedback data, such as fuel and coolant temperatures can be taken from the 3D module, and the resulting core power can be deposited in the 3D heat structure. That is, the point kinetics model of the 1D module was implemented into the heat structure model of the 3D module.

6. Restructuring and Modernization of the MARS Code

As mentioned previously, the basic structures of RELAP5 and COBRA-TF were developed in the 1970s and therefore they do not take advantage of recent advances in computer technology and new programming languages.

6.1. Restructuring and New Dynamic Memory Management

The basic frames of the MARS code, i.e., the RELAP5 and COBRA-TF codes, are written in Fortran 77 and Fortran IV, respectively. Because these old programming languages did not provide a means for simple dynamic memory allocation, a very complicated programming technique (e.g. the FTB package) was invoked to overcome the limitation. Unfortunately, the use of the FTB package [8] produced cryptic coding and compromised readability, maintainability and portability. To resolve these problems, all the MARS subroutines were converted into the standard Fortran 90 [9] using an automatic conversion program [6]. The Fortran 90 was selected among various new programming languages, because it can facilitate the linkage of the existing programs written in old languages and the recent new programming environment.

For the MARS code restructuring, it is necessary first to understand the memory allocation schemes used in the original subroutines. For instance, consider that a hydrodynamic volume needs three real variables, *p*, *t*, and v. Let's assume that total number of volumes is 100. In this case, two types of memory allocations are possible to treat *n* volumes' data:

```
Type 1: paramater n=100
        real(8) p,t,v
        common/voldat/p(n),t(n),v(n)
Type 2: real(8) fa(300)     ! a big array
        common/fast/fa
        real(8) p(1),t(1),v(1)
        equivalence (p(1),fa(1)),(t(1),fa(2)),(v(1),fa(3))
```

COBRA-TF uses Type 1. In this case, the index is very clear, but it is difficult to treat removal/addition of data set for *i-th* volume because a data set [p(i),t(i),v(i)] is not stored

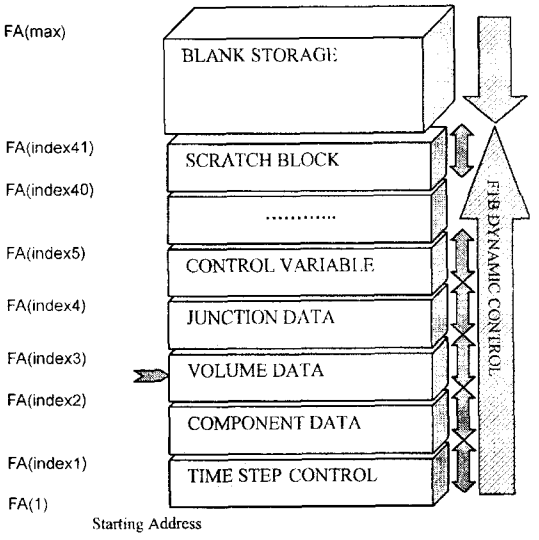

Fig. 4. Dynamic Memory Allocation of the Original RELAP5 Code.

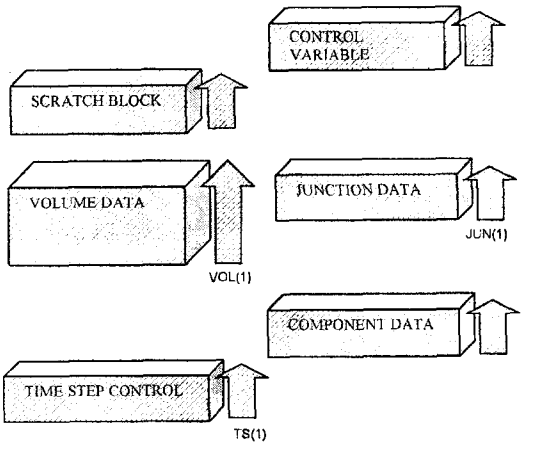

Fig. 5. New Dynamic Memory Allocation of the MARS Code.

contiguously. Meanwhile RELAP5 uses Type 2, where p(1), v(1), and t(1) are sequentially equivalenced to a big one-dimensional array. Type 2 is easy to treat the volume data removal/addition because the data set for a volume are stored contiguously, but difficult to understand the array index because of the "equivalence" statement. However, using the derived type variables (or structured variables) of the standard Fortran 90, the advantages of the two types can be taken at the same time. The above arrays can be written in Fortran 90 as follows:

```
Type 3:  TYPE VOL_DATA; SEQUENCE
           REAL(8) p      ! pressure
           REAL(8) t      ! temperature
           REAL(8) v      ! velocity
         END TYPE
         TYPE (VOL_DATA), ALLOCATABLE:: vd(:)
```

Using the above statements, p(i), t(i) & v(i) get to equivalent with vd(i)%p, vd(i)%t & vd(i)%v, respectively. When the above statements are included in a "MODULE··· END MODULE" statement, the data can be shared between subroutines by using the "USE" statement. Therefore, the "COMMON, EQUIVALENCE, and INCLUDE" statements in the original MARS code can be replaced with "MODULE and USE" statements. By using the derived type variables, the code maintenance and readability are greatly enhanced. At the same time, it gives a basis for a new dynamic memory management (DMM) scheme.

The FTB package has been used for the RELAP5 DMM. The scheme is very flexible and successful in memory usage, but it is difficult to understand and change. The scheme also requires a significant effort to find any index error. Figure 4 illustrates the schematic of the old RELAP5 DMM. The RELAP5 code uses 40 dynamic common data blocks (such as data blocks for components, volumes, junction, heat structures, control variables) at maximum. All the data blocks are equivalenced to a big one-dimensional array, FA. The size of each data block is dependent on the user's input, i.e. the problem size. This means each of 40 variable-sized data blocks is stacked

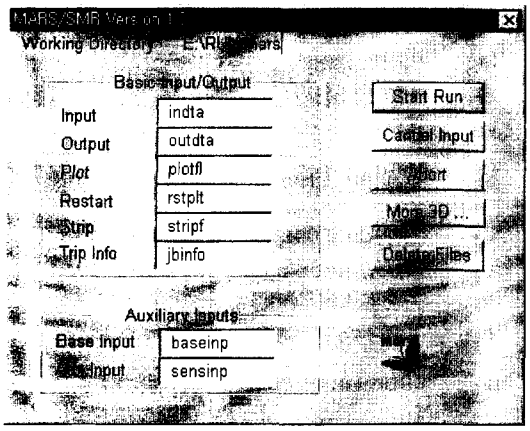

Fig. 6. Dialog Box of the MARS Run Problem.

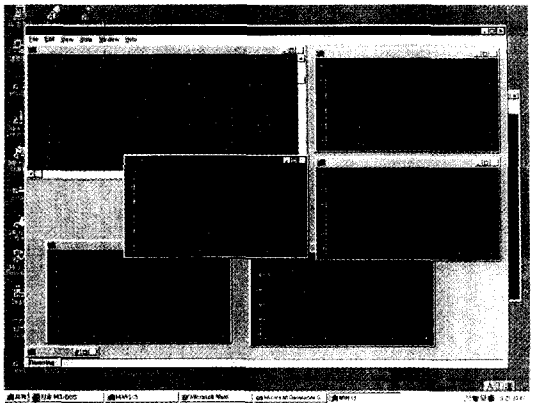

Fig. 7. Windows Version of MARS 1.3.1 During Execution.

one-by-one and dynamically managed as shown in Fig. 4. Thus, to find a variable, we have to know both the starting address of the block storing the variable and the offset in the block.

With the new features mentioned above, it is possible to implement a new DMM scheme in a more clear and safe way. Each of the FTB data blocks is replaced with a separate data block, which uses suitably organized derived type variables like Type 3 (see Fig. 5). The programmer can directly control creation and deletion of these blocks using ALLOCATE and DEALLOCATE statement. The block size is specified when the derived type variable is allocated by ALLOCATE statement.

Comparing the old and new DMM schemes, the new scheme provides much better readability. For example, to manipulate the pressure of i-th volume, the following Fortran statements are used:

```
(a) in RELAP5          (b) In the 1D module of MARS
  i=filndx(4)
  DO k=1,nvols(2,i)      DO i=1,v_hd%nvols(2)
    pix=p(i)               pix=vd(i)%p
    ..                     ..
    i=i+ivskp
  ENDDO                  ENDDO
```

Furthermore, using the new DMM scheme, another advantage of index-error-free feature is taken, that is, most of index errors can be checked during either compilation or execution.

In order to realize the new DMM feature, approximately 40 percents of the original source programs of the MARS 1D module were rewritten. In the 3D module, a memory allocation of Type 1 is used and the DMM scheme is absent. Instead, array size should be big enough to accommodate any problems. This feature invoked unnecessary memory size and bigger restart files. Thus, the new DMM was implemented for some major variables.

6.2. Windows Programming

Previously, the user interface of the MARS code consisted of a set of input file, output files, and restart/plot files. To analyze the transient behavior of variables, users had to extract the data and, then, make x-y plots using another software. It can be said that the code was not user-friendly. This was mainly caused by the text-oriented command-line programming technique. Now, using the

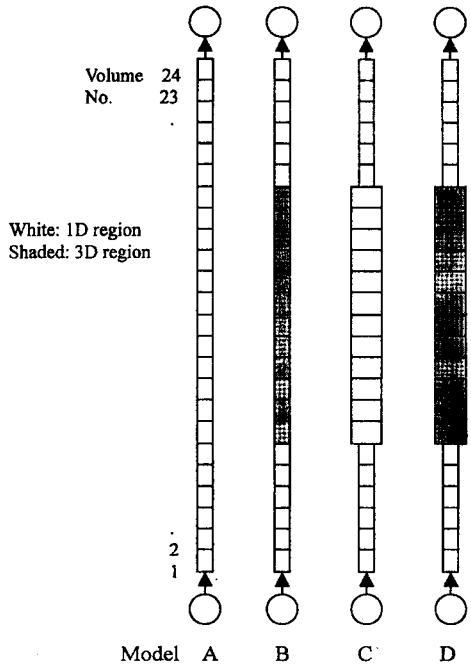

Fig. 8. MARS Nodalization for the Single Channel Tests.

Windows programming technique, the user-friendly features can be easily implemented with minimal changes of the source programs.

Thus, we adopted the QuickWin run-time library of Digital Company Visual Fortran [9] as a tool for user interface enhancement. The library supports pixel-based graphics, real-coordinate graphics, text windows, character fonts, user-defined menus, mouse events, and editing of text and graphics. The MARS code was re-compiled with the QuickWin feature in Visual Fortran, and the mask of user's dialog and on-line graphical x-y plot were established. This feature is available for PC Windows users and provides simple GUI features as shown in Figs. 6 and 7. Now we can start the MARS code by a click of Mouse and can analyze the transients using on-line x-y plots. The feature is evaluated to be more convenient and efficient than expected.

7. Verification of the Developmental Works

Because the MARS code was developed from the unification of the RELAP5 and COBRA-TF codes, it is necessary to verify the added new features as well as the unification works. Each of the developmental works has been verified step by step through appropriate procedure. These can be summarized as follows:

- Hydrodynamic model unification, and modification of the thermodynamic and transport property routines
- Heat structure coupling
- Implementation of the point kinetics model into the 3D module
- Input processing unification
- Code restructuring and modernization.

7.1. Verification of the Hydrodynamic Model Unification

The conservation of mass, energy, and momentum across the interface between the 1D and 3D regions should be verified first of all. An efficient way to verify these is to perform single channel tests; steady-state flow in a single channel is modeled as shown in Fig. 8, where the inlet flow conditions and the exit pressure are given as boundary conditions. The conservation of mass can be confirmed by comparing mass flow rates at the inlet and the exit. Likewise, the conservation of energy can be seen. The thermodynamic property routine modifications can be also assessed through these tests. Finally, the momentum conservation at the 1D/3D interface is verified by comparing the local pressure drops.

Figure 8 shows the MARS nodalization for the tests. Models A and C use the 1D module only, whereas Models B and D uses both the 1D and 3D modules. It is noted the 3D module can be

Table 1. The FDEs and Independent Scalar Variables in the 1D and 3D Modules

Code	COBRA-TF	RELAP5/MOD3.2
FDEs	• Continuity eq. of noncondensible gases (CN) • Continuity eq. of continuous liquid (CL) • Energy eq. of vapor (EV) • Energy eq. of continuous and entrained liquid (EL) • Continuity eq. of entrained liquid (CE) • Continuity eq. of vapor (CV)	• Density eq. of non-condensible gases • Energy eq. of vapor • Energy eq. of liquid • Difference density eq. • Sum density eq.
Independent variables	$\alpha_v P_g$, α_v, $\alpha_v h_v$, $(1-\alpha_v)h_l$, α_e, *and* P $(\alpha_l + \alpha_v + \alpha_e = 1)$	X_n, U_g, U_f, α_g, *and* P $(\alpha_f + \alpha_g = 1)$

Table 2. Boundary Conditions for the Single Channel Tests Inlet Flow Condition Exit Pressure (MPa)

Cases	Inlet flow condition	Exit pressure (MPa)
1	Subcooled water; 600 K, 10 kg/s	15
2	Two-phase mix.; sat. liq., 4.4 kg/s, sat. vap., 0.6 kg/s	12
3	Two-phase mix.; sat. liq., 4.7 kg/s, sat. vap., 0.3 kg/s	8
4	Two-phase mix.; sat. liq., 4.7 kg/s, sat. vap., 0.3 kg/s, X_n=0.3 (noncondensible gas quality)	5
5	Superheated steam; 400 K, 0.3 kg/s	0.1013

reduced one-dimensionally. Models A and B represent a vertical pipe of 0.1 m inner diameter and 4.8 m long, which is divided into 24 equal-length volumes. In the case of Models C and D, the flow areas of the central 12 volumes are increased twice. These models are selected to check the momentum balance at the 1D/3D interface with sudden area changes. The boundary conditions at the inlet and exit are listed in Table 2. To cover a wide range of flow conditions, five different boundary conditions named Case 1 through 5 are given. These include single-phase liquid, two-phase mixture, two-phase mixture with noncondensible gas, and single-phase steam flows, with pressure ranging from atmospheric pressure to 15 MPa.

The steady-state calculation results are summarized in Table 3. The mass conservation is easily confirmed from the mass flow rates at the inlet and the exit. Table 3(i) shows that the mass errors across the 1D/3D interfaces of Models B and D are zero except for Case 5 (single-phase steam flow). This is due to the difference of single-phase flow modeling in the 1D and the 3D modules. In the 3D module, even in the case of single-phase steam flow, a small amount of continuous liquid and droplets (minimum volume fraction = 10^{-8}) is assumed to exist. As a result, the results of Models B and D have small mass errors, but this is negligible in practice. The same kind of mass error occurs in Case 1 (single-phase liquid flow). However, in this case, the mass error is evoked by steam phase with minimum volume fraction, thus the resulting mass error is more negligible because of large density difference.

Table 2. The Results of Single Channel Tests

(i) Mass error: $(W_{exit}-W_{inlet})/W_{inlet}$

Cases	Model A	Model B	Model C	Model D
Case 1	0.0	0.0	0.0	0.0
Case 2	0.0	0.0	0.0	0.0
Case 3	0.0	0.0	0.0	0.0
Case 4	0.0	0.0	0.0	0.0
Case 5	0.0	0.000033	0.0	0.000033

(ii) Total flow enthalpy difference: $(H_{exit}-H_{inlet})/H_{inlet}$

Cases	Model A	Model B	Model C	Model D
Case 1	-1.95370E-05	-4.01116E-06	-2.00844E-05	-5.32035E-06
Case 2	-3.64642E-05	-2.06707E-05	-3.70136E-05	-2.14365E-05
Case 3	-3.75362E-05	-1.98849E-05	-3.80158E-05	-1.92586E-05
Case 4	-8.98223E-05	-5.42773E-05	-8.80871E-05	-4.99015E-05
Case 5	-6.61338E-05	1.72326E-04	-4.51442E-04	-1.01775E-03

(iii) Pressure drops (Pa) : $dP_{3,4}$, $dP_{6,7}$, $dP_{18,19}$, $dP_{1,24}$, $(dP_{i,j}=P_i - P_j)$

Case 1	1346.7 1346.7 1346.6 30973.	1346.7 1346.7 1346.6 30807.	1346.7 401.32 2247.0 30432.	1346.7 401.32 2246.9 30415.
Case 2	802.79 802.49 801.89 18404.	802.77 761.49 801.64 17665.	802.81 422.61 1191.8 18854.	802.77 373.08 1185.8 17665.
Case 3	935.82 935.78 934.15 21342.	935.70 878.23 933.73 20110.	935.89 627.33 1268.7 22038.	935.74 561.48 1262.1 20474.
Case 4	837.05 841.81 848.12 19425.	836.81 714.13 877.18 17548.	837.05 352.54 1286.4 18863.	837.03 297.87 1241.3 18876.
Case 5	61.804 61.917 62.373 1429.3	61.663 61.370 61.213 1729.7	62.091 -957.11 1019.7 826.30	62.055 -954.98 1017.6 898.02

The energy conservation is shown in Table 3(ii). Due to the flow work, the total flow enthalpy slightly decreases along the channel. The results of Models A and C show this effect. It is found that Models B and D yield relative small enthalpy decreases. This is because the 3D module neglects the flow work term, while the 1D module takes it into account. Considering this difference and the mass error described above, the difference of the flow enthalpy decreases can be explained.

The momentum conservation is assured from single- and two-phase pressure drops across the 1D/3D interface. Comparing the results of Models A and B, it can be seen that the local pressure drops agree well with each other. Some deviations in Cases 2 through 4 are due to the different interfacial friction models in the 1D and 3D modules. In general, the interfacial friction of the 3D module was greater than that of the 1D module. This feature results in the higher void fraction and, in turn, the smaller body forces in the 3D region. The results of Models C and D show that the momentum is conserved even in case of sudden area change at the 1D/3D interfaces. The pressure drop differences in case of two-phase flows are due to the same reason described above.

In summary, it is confirmed that the mass, energy, and momentum are conserved well across the 1D/3D interface. Thus it can be concluded that both the hydrodynamic model unification and the modifications of the thermodynamic property routines were successfully performed.

7.2. Verification of the Heat Structure Coupling

The code modifications relevant to the "heat structure coupling"are (i) data mapping from the 3D hydrodynamic volume data into the 1D hydrodynamic volume data form, and (ii) energy partition, that is, wall heat flux partition and near-wall vapor generation/condensation, from the 1D heat structure to the 3D hydrodynamic volume. To fully test the effect of the relevant code modifications, a systematic assessment was needed so that 21 heat transfer regimes (including the effect of noncondensible gases) of the 1D heat structure model [1] can be covered.

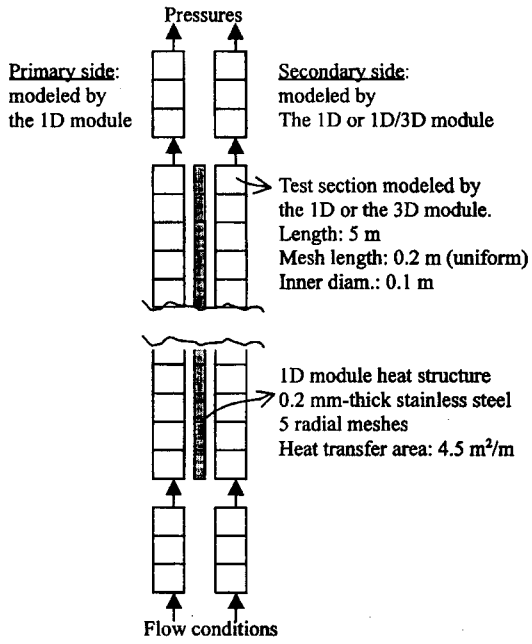

Fig. 9. An Example of "Heat Structure Coupling".

Figure 9 shows an example of the input model for testing. Two separated pipes thermally connected by a heat structure were modeled. The primary pipe is modeled using the 1D module and the secondary pipe is modeled using either the 1D module or the 1D/3D modules, where the heat structure is modeled using the 1D heat structure model. Boundary conditions are listed below:

- Primary side: exit pressure 15 MPa, inlet water temp. 610 K, flow rate 20 kg/s,

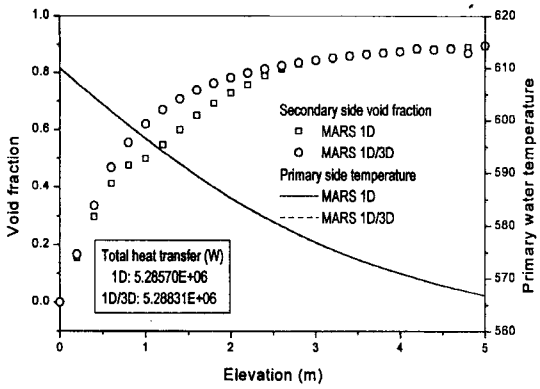

Fig. 10. Results of the "Heat Structure Coupling" Test.

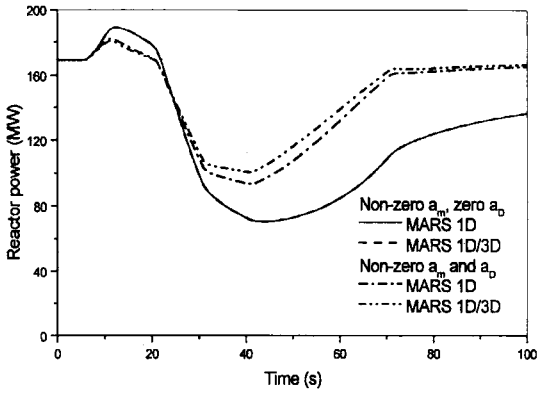

Fig. 11. Results of the Point Kinetics Model Test : Total Reactivity vs. Time.

- Secondary side: exit pressure 7 MPa, inlet water temp. 550 K, flow rate 10 kg/s.

The steady-state calculation results are compared in Fig. 10. The primary coolant temperature profiles agreed exactly, however the secondary void profiles showed some discrepancy. This is due to higher interfacial drag force in the 3D module and, in turn, higher void fraction. Difference in the total primary-to-secondary heat transfer rate was 0.0495 %. But, it was confirmed that the differences are not

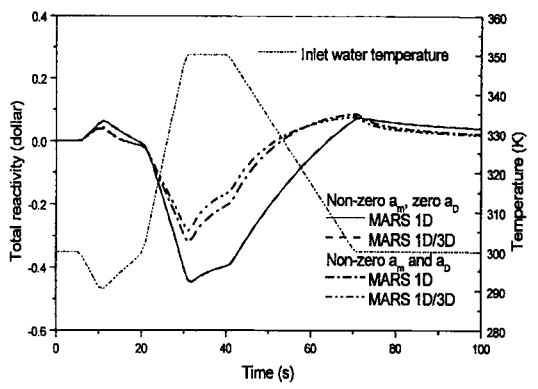

Fig. 12. Results of the Point Kinetics Model Test : Reactor Power vs. Time.

related to the "heat structure coupling", but due to the difference of the hydrodynamic models of the 1D and 3D modules. In this test, two heat transfer regimes of subcooled nucleate boiling and saturated nucleate boiling appeared. Detailed assessment procedures and results for other regimes, such as single phase liquid convection, subcooled transition film boiling, saturated transition film boiling, subcooled film boiling, saturated film boiling, single phase vapor convection, condensation including the effect of noncondensible gases, are documented in Ref. 7. The results showed that the implementation of "heat structure coupling" algorithm is complete.

7.3. Verification of the Implementation of Point Kinetics Model into the 3D Module

To verify the implementation of the point kinetics model into the 3D module, a conceptual problem was established:

- Core power: 168.9 MW
- Rod length: 5 m (active rod length 4 m)
- Total number of fuel rods: 2640

Two input models, similar to Model A and B in Fig.8, were made for the above conceptual problem. The central pipe contains the reactor core. The MARS 1D input model with the reactor core was made using the 1D heat structure model and the point kinetics model. For comparison, in the 1D/3D input model, the core region was replaced with the 3D module; that is, a hydrodynamic channel of the 3D module and a "rod" component that uses the point kinetics model.

In each calculation, an initial steady state was reached by null transient calculation with fixed boundary conditions and zero feedback coefficients. The steady-state boundary conditions are given below:

- Inlet water temperature: 300 K
- Mass flow rate: 998 kg/s
- Exit pressure: 0.4 MPa.

Then, transient calculations are performed with changes of inlet coolant temperature as shown in Fig. 11. To see the effects of moderator temperature and Doppler feedback coefficient (a_m and a_D) separately, Doppler feedback coefficients were set to zero in the first transient calculation and, then, another transient was calculated with nonzero Doppler feedback coefficient.

The results of 1D and 1D/3D calculations are compared in Figs. 11 and 12. As can be seen, when the Doppler feedback coefficient was set to zero, the two results are exactly the same. But, with non-zero Doppler feedback coefficient, the two results show some deviations. The difference between the 1D and 3D heat structure modelings results in differences in the average fuel rod tempratures, which again results in different kinetics behaviors because of Doppler feedback. Nevertheless, overall transient behaviors agree well with each other. These calculations were repeated with a reduced Doppler feedback and, then, the differences in the reactor kinetic behavior also decreased. Thus, it can be said that the point kinetics model was successfully implemented into the 3D module.

7.4. Verification of the Input Process Unification

The input data of the original MARS code consists of the 3D module's input with fixed format and the 1D module's input with free format. These data files should be separately maintained. For better user friendliness and input maintenance, the two input files were integrated into a single input file and, at the same time, the input structure of the 3D module was changed into a free format like the 1D module input. To accomplish this unification, the INP package used in the 1D module [11] was adapted in the 3D module's input processing routines. So, input card identification numbers were given first to each of the 3D module input data. Then, the original 3D module's input processing routines were appropriately modified.

Verification of the input unification is straightforward because the associated code modifications do not affect the calculation results. To verify the modification, the existing 3D input data files (that we have used) were changed into the new input format and, then, merged with the corresponding 1D input. The test input data include those for the single channel tests, FLECHT SEASET tests, THTF test 105, LOFT L2-3 and L2-5 tests, and other 1D/3D test problems [6]. These input data cover all aspects of the 3D input form. Some of these are steady-state tests and others are transient tests. Then, after running, the output files of MARS were compared with those of a

MARS version without the input unification. For the comparison of output files, the utility software that can compare each line of two files was used. The results showed that the output files were exactly the same, confirming that the input unification was successfully completed.

7.5. Verification of the Code Restructuring and Modernization

As described in Sec. 6.1, approximately 40 % of the MARS 1D module (i.e., RELAP5) source programs were changed during the code restructuring, and new features were added during the code modernization. However, like the input processing unification, these modifications should not affect the calculation results. Therefore, two versions of MARS with and without the code modifications were executed with a variety of input data, which cover nearly all aspects of the RELAP5 input feature. These include the input data for three different power plants, five integral effect tests, and a number of separate effect tests [6]. Then, the output files were compared with each other. From the results of the comparisons, it was confirmed that the code modification did not affect the calculation results at all. Instead, the computational speed was found to be slower by 5 to 10 percents after the modifications.

8. Conclusions

A best-estimate multi-dimensional system analysis code, MARS 1.3.1, has been developed. In the MARS code, all the functional modules of the COBRA-TF and RELAP5 codes were unified. Each of the unification works was verified step by step and, finally, the RELAP5 and COBRA-TF codes were successfully consolidated as the 1D and 3D module in the MARS code, respectively. Then, to establish a basic code frame for further development, the source programs of MARS were converted into standard Fortran 90, and they were restructured using a modular data structure based on the "derived type variables" and a new dynamic memory allocation scheme. This eliminated the cryptic coding of the original RELAP5 code and greatly enhanced code readability, flexibility, maintainability, and portability. In addition, to facilitate the use of MARS, the Window features were developed as a part of the GUI system.

In conclusion, a firm basis for the development of a best-estimate multi-dimensional system analysis code has been successfully established. For further improvement, the coupled capabilities of three-dimensional reactor kinetics and containment analysis are currently being developed[12,13]. These improvements will extend the MARS applicability not only to the LOCAs but also to the non-LOCAs and the beyond-design-basis accidents.

Nomenclature

A : Flow area
$\underline{\underline{A}}$: matrix
B_x : Body force
b : vector
C : coefficient for the virtual mass term
F : volumetric force
h : enthalpy
$\vec{M}_d^{\Gamma}$: average supply of momentum to phase k due to mass transfer to phase k
$\vec{M}_k^{d}$: average drag force on phase k by other phases
NR : number of COBRA-TF to RELAP5 junctions
P : pressure
Q : volumetric heat source
$\vec{Q}_k$: average k-phase conduction vector

q_{ik}'' : Interfacial heat flux
$\overrightarrow{q}_k''$: k-phase turbulent heat flux
$\overrightarrow{T}_k$: average phasic turbulent stress tensor
U : internal energy
V : velocity
W : mass flow rate
X_n : ratio of the noncondensible gas mass to the total gaseous phase mass
z : spatial coordinate

Greek

α : volumetric fraction or coefficient
$\delta\phi$: $\phi^{n+1} - \phi^n$
ρ : density
$\overrightarrow{\tau}_k$: average k-phase viscous stress tensor
Γ : volumetric vapor generation rate

Subscripts

DISS : wall friction dissipation
e : entrained liquid droplet phase
f : liquid phase
g, v : vapor or gas phase
gas : noncondensible gas phase
i : interface
ik : interface to k-phase
l : continuous liquid phase
m : two-phase mixture
w : wall

Superscripts

n : old time-step
$n+1$: new time-step
s : saturated
T : transpose

Acknowledgment

This project has been carried out under the Nuclear R&D Program by Ministry of Science and Technology(MOST). The authors wish to express their appreciation to those who were involved in the development of RELAP5 and COBRA-TF. The authors also wish the MARS code be used for enhancing nuclear reactor safety.

References

1. RELAP5 Code Manual, NUREG/CR-5535, USNRC (1995).
2. Thurgood, M. J. et al., NUREG/CR-3046, USNRC (1983).
3. Driskell, W. E. and Hanson, R. G., "Summary of ICAP Assessments of RELAP5/MOD2," Nuclear Safety, 30, 2, 343 (1989).
4. Lee, S. Y., Jeong, J.-J., Kim, S. H., and Chang, S. H., "COBRA/RELAP5: A Merged Version of the COBRA-TF and RELAP5/MOD3 Codes," Nuclear Technology, 99, 177 (1992).
5. Jeong, J.-J. et al., "Development of the Unified Version of COBRA/RELAP5," Proc. Korean Nuclear Society '97 Autumn Meeting, Korean Nuclear Society (1997).
6. Lee, W. J. et al., Development of A Multi-Dimensional Realistic Thermal-Hydraulic System Analysis Code, MARS 1.3 and Its Verification, KAERI/TR-1108/98, KAERI (1998).
7. Lee, W. J. et al., Improvement of Multi-Dimensional Realistic Thermal-Hydraulic System Analysis Code, MARS 1.3, KAERI/TR-1141/98, KAERI (1998).
8. Kelly, J. M., OECD/CSNI Workshop on Transient Thermal-Hydraulic & Neutronic Codes Requirements, Nov. 5-8, (1996).
9. Digital Fortran Language Reference Manual, Digital Equipment Corporation (1997).
10. ASME steam tables, Fifth edition, ASME (1992).

11. Kim, J. S. et al., Reactor Accident Analysis and Evaluation: KAERI Improved Nuclear Environmental Package (KINEP), KAERI/RR-384/82, KAERI (1982).
12. Jeong, J.-J. et al., "Development of a Draft Version of MARS/MASTER; A Coupled Code of MARS 1.3 and MASTER 2.0", Proc. Korean Nuclear Society '98 Autumn Meeting, Korean Nuclear Society (1998).
13. Chung, B. D. et al., "MARS 1.3 System Analysis Code Coupling with CONTEMPT4/MOD5/PCCS Containment Analysis Code using Dynamic Link Library," Proc. Korean Nuclear Society '98 Autumn Meeting, Korean Nuclear Society (1998).