CHAPTER 7
SINGLE AND TWO-PHASE FLOW MODELLING II: AN INTRODUCTION TO LOCA SYSTEM CODE DEVELOPMENT AND TWO-FLUID MODELLING

B.H. McDonald
Thermalhydraulics Research Branch
Whiteshell Nuclear Research Establishment
Atomic Energy of Canada Limited
Pinawa, Manitoba, ROE 1LO

ABSTRACT

An introduction to the methodology of LOCA system code development is presented, with emphasis on the evolutionary and iterative process involved. It is frequently necessary to improve and extend component models to provide better representation of physical effects, and the two-fluid model is described for improved treatment of flow-boiling during certain postulated accident scenarios.

7.1 Introduction

Analysis of reactor loss-of-coolant accidents (LOCA) is important to both design and safety engineers. Beginning with a normally operating reactor, one must follow, step-by-step, the sequence of events resulting from a postulated leak or rupture of the heat transport system (HTS). The size and location of the break both affect the sequence. Since it is simply not feasible to subject real reactors to the kind of experiments that would be needed to answer all the questions, the engineer must rely on analytical tools, tested against simpler experiments, to assist him in constructing realistic scenarios following postulated accidents.

One of the more powerful tools available to the engineer is the LOCA system code. Such codes contain mathematical models of all the physical process anticipated during a LOCA, written in algebraic form for digital computation. The process of developing a code involves compiling all the mathematical models, with whatever empirical data is required, reducing them to algebra using numerical methods, and testing them against whatever experimental evidence exists for validation. The process is iterative. Final testing may reveal a flaw in a basic model - which can send the code designer back to the drawing board. Codes available today, such as FIREBIRD, SOPHT, RELAP, TRAC, RETRAN, RAMA, and so on, are all at some stage in this development. They have been validated for some scenarios, not for others. The engineer is advised to use the tools carefully.

In this session we will look generally at what LOCA system codes do, how they are constructed, and what we can expect of them. The first reference, on RAMA, by Barclay, Bean and Nieman describes some of the aspects of code development and validation. It was mentioned earlier that flaws in basic models can send the designers back to the drawing board; we shall look at an example of this which involves the need to look at liquid and vapour flow separately in the same pipe - the two-fluid model. The second reference, on this development for RAMA by Hanna et al, describes some of the agony involved.

7.2 How LOCA Codes Work

In the most general sense, the basic operation of a typical code may be described as follows: given the status of the reactor now, with boundary conditions, produce the status of the reactor at some future time. For example, if we have the status at time t_0, the code will produce the status at time t_1, (where the timestep is t_1-t_0). If we then repeat the process starting from the status at t_1 the code will produce the status at t_2, and so on. Thus the typical code can trace the evolution of a transient event, like a LOCA. If the reactor is operating steadily, or nearly steadily, its status is changing slowly if at all, and we would like to use large timesteps. However, during rapid transients - triggered by a changing boundary condition like a guillotine pipe break at a specified time - the reactor status can change very quickly and small timesteps are required. Automatic timestep control is a feature of most reactor LOCA codes. Output from a system code is like a sequence of snapshots (status reports) or like frames of a motion picture, with time-lapse photographic techniques for slow transients, and slow-motion techniques for fast transients.

The photographer has to decide what lens to use, depending on the resolution he wants in each frame. Similarly, the information described as "reactor status" depends on the resolution desired when one is dealing with a LOCA system code. Clearly, any item of information which can change during a postulated LOCA must be included - for example flow conditions throughout the circuit, piping network temperatures, fuel bundle temperatures and power, pump conditions, boiler conditions, break flow conditions, cold water injection flow conditions, and so on. However, the code user may select the number of flow condition stations in the circuit, for example, at the outset, thus defining ahead of time the resolution. Although some work is being done in automatically changing resolution during code operation, for most codes the user specifies it, and may have to respecify it, if the original estimate was incorrect. Just as there is an art to photography, there is an art to getting the most out of a LOCA system code.

We now turn our attention to what is inside the code: the rules for transforming the reactor status from time t_1 to t_2. These rules end up being in the form of algebraic relationships whatever their sources. Some come directly - steam-water property routines, friction factors, heat transfer coefficients are for example often already in algebraic form. Some, however, come from the application of numerical methods to partial differential equations: for example the flow-boiling equations, and the conductive heat transfer equation, and these may be subject to timestep and spatial resolution criteria for accuracy and stability. The quality of the model as well as the quality of its algebraic representation are both important to code performance. During any step, one model will be more dominant than the others in prescribing the change in reactor status. If that model is mathematically capable of describing the physics, and its numerics are of good quality, the change in reactor status will describe reality quite well, and with reasonable efficiency. However, if the models or the numerics are incompetent, either quality of prediction or efficiency (or both) will suffer. Experience with RAMA suggests that sooner or later ..... more developmental work will be required to improve the operation of any code.

7.3 Code Validation

From the preceeding discussion it should be clear that if all physical effects which can dominate in a postulated LOCA are modelled well, both mathematically and algebraically within the code, and the code user is adept at using the tool, the predicted sequence of events should be realistic. The question is proof.

The pressurized water loops at Whiteshell are providing data from LOCA-type experiments at various scales by which our codes may be validated. The RD-12 rig is a medium scale replica of a CANDU PHTS circuit, complete with electric heaters (fuel simulators), headers, pumps and boilers. The RD-14 loop, presently under construction, will be a nearly full scale replica. Our LOCA codes should be able to produce reasonable facsimiles of the measured loop transients when subjected to the same loop specifications and boundary conditions.

We have observed good agreement in some cases and poor agreement in others. One model, the homogeneous equilibrium model (HEM) for flow-boiling, has come under suspicion for its inability to cope with separated vapour and liquid flows. While in many experiments the driving forces remain large and the assumption of well mixed flows leads to good code representation, in other experiments the vapour rises and the liquid falls. Global stratification can affect the whole loop, and stratification within the test channels can lead to upper pins getting hotter than lower pins. The HEM simply cannot cope.

7.4 Two-Fluid Modelling

We leave the generalities of LOCA system codes now and focus our attention on a more sophisticated basic model for two-phase flow. To begin with we will consider the implications to the system code itself of a two-fluid model, wherein at any location in a duct, the vapour and liquid streams may have different velocities and temperatures.

Firstly consider heat transfer. With the HEM, heat from the piping network and from the fuel bundle (or fuel simulator bundle) is simply dumped into the steam-water mixture using well-known, but approximate, heat transfer correlations. Now we must split heat transfer between the two separate phases. Models for doing that, especially during transients, need some development. As well, the bundle itself may have some pins exposed to vapour with others submerged in liquid, prescribing the need for a more sophisticated bundle model. Code decisions must be made in light of the current flow pattern, or flow regime at the channel cross-section. Flow regime data, and flow regime transition points during transients are hard to come by, and fundamental experiments and analysis are needed.

Now consider where separated flow can occur. Certainly in the fuel channels we can expect to see stratification, and in fact, this is currently the main thrust of our work. However, during global stratification events we will have to model separated flows through the boilers, pumps and headers. We may have to delay code validation in global stratification cases for some time.

Finally consider the effect of incorporating a two-fluid model on the numerics. The methods for generating algebraic equations for the HEM may have to be modified to cope with the more sophisticated two-fluid model. Automatic timestep control strategies may need revision, and the guidelines for spatial resolution may need modification which could involve retraining code users.

The purpose of this discussion has been to show that the ramifications of a change in a basic model can be extensive. It is one reason why code development can be a slow process and why code users are often asked to be patient.

7.5 The Two-Fluid Model

In this section we will examine the two-fluid model itself. There are many ways to derive the appropriate equations, and the reader is referred to Mathers et al [3] for one such derivation. Here we look at a simpler way of approaching the two-fluid model, beginning with the mass conservation equation for one-dimensional flow in a duct:

$$A \frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x} A\rho u = o, \qquad 7.1$$

where A is the duct cross-sectional area, ρ is mixture density, and u is mixture velocity. We now define

$$\rho = \sum_k \alpha_k \rho_k \qquad 7.2$$

$$A\rho u = A \sum_k \alpha_k \rho_k u_k \qquad 7.3$$

where α_k is the fraction of the duct cross-section occupied by phase k. The summation runs from k=1 (vapour) to k=2 (liquid). We now substitute these definitions into equation (7.1):

$$A \frac{\partial}{\partial t} \sum_k \alpha_k \rho_k + \frac{\partial}{\partial x} A \sum_k \alpha_k \rho_k u_k = o \qquad 7.4$$

or, interchanging the order of summation and differentiation:

$$\sum_k \left\{ A \frac{\partial}{\partial t} \alpha_k \rho_k + \frac{\partial}{\partial x} A \alpha_k \rho_k u_k \right\} = o \quad . \qquad 7.5$$

We may now identify the phase k mass conservation equation.

$$A \frac{\partial}{\partial t} \alpha_k \rho_k + \frac{\partial}{\partial x} A \alpha_k \rho_k u_k = \Gamma_{mk} \quad ; \quad k=1,2 \qquad 7.6$$

where Γ_{mk} is the phase generation term (e.g. vapour generation rate for the

vapour phase). Mixture mass conservation, equation (7.5), is satisfied by enforcing the mass interphase jump condition

$$\sum_k \Gamma_{mk} = 0 . \qquad 7.7$$

We have replaced the single mass conservation equation (7.1) by two phase conservation equations (7.6) plus a jump condition (7.7). Note that the jump condition says nothing about the nature of the phase generation terms; this information would have to be supplied by reference to some interphase transfer law determined for example from fundamental experiments: a constitutive equation.

If we repeat this process for the mixture momentum and energy equations we should end up with six partial differential equations (mass, momentum and energy conservation for each phase) and three algabraic jump conditions. These, together with the interphase transfer laws, define a two-fluid model. It is well beyond the scope of this introduction to complete the derivation, but some interesting observations can be made about the process by looking at the mixture momentum conservation equation.

$$A \frac{\partial}{\partial t} \rho u + \frac{\partial}{\partial x} A\rho u^2 + A \frac{\partial p}{\partial x} = f \qquad 7.8$$

where p is the mixture pressure and f is the momentum source term. If we substitute the same definitions as before, equations (7.2), (7.3), we get

$$A \frac{\partial}{\partial t} \sum_k \alpha_k \rho_k u_k + \frac{\partial}{\partial x} A\, u \sum_k \alpha_k \rho_k u_k + A \frac{\partial p}{\partial x} = f \qquad 7.9$$

which is the same as

$$A \frac{\partial}{\partial t} \sum_k \alpha_k \rho_k u_k + \frac{\partial}{\partial x} \sum_k A\, \alpha_k \rho_k u_k^2 + \frac{\partial}{\partial x} \sum_k A\, \alpha_k \rho_k u_k (u-u_k)$$

$$+ A \frac{\partial p}{\partial x} \sum_k \alpha_k = f \sum_k \alpha_k . \qquad 7.10$$

Interchanging summation and differentiation, we can write

$$\sum_k \left\{ A \frac{\partial}{\partial t} \alpha_k \rho_k u_k + \frac{\partial}{\partial x} A\, \alpha_k \rho_k u_k^2 + \alpha_k A \frac{\partial p}{\partial x} - \alpha_k f \right\} = - \frac{\partial}{\partial x} \sum_k A\, \alpha_k \rho_k u_k (u-u_k) . \qquad 7.11$$

This permits us to identify a phase k momentum equation in the same spirit as with mass:

$$A \frac{\partial}{\partial t} \alpha_k \rho_k u_k + \frac{\partial}{\partial x} A \alpha_k \rho_k u_k^2 + \alpha_k A \frac{\partial p}{\partial x} - \alpha_k f = \Gamma_{fk}; \; k=1,2 \qquad 7.12$$

where Γ_{fk} is the interphase momentum source for phase k. Having separate phase momentum equations permits each phase to move and accelerate separately, which is necessary for realistic treatment of stratified flows. Mixture momentum conservation is produced by enforcing the jump condition:

$$\sum_k \Gamma_{fk} = - \frac{\partial}{\partial x} \sum_k A \alpha_k \rho_k u_k (u-u_k) \qquad 7.13$$

We note that the jump condition is not algebraic; it contains a spatial derivative. More work is needed. We note as well an arbitrary split of the source term f between the phases, and that pressure has been assumed to be the same in each phase. One must really resort to a formal derivation, as in reference [3], to resolve some of these questions.

Suffice it to say here that a well-defined, if complex, two-fluid model exists and has been implemented in the RAMA code [2].

Work is currently focussed on defining the interphase transfer laws for each flow regime or flow pattern expected, from well-mixed flows (bubbly or droplet) to well-separated flows (stratified or annular), and to produce a realistic treatment for flow-regime transitions.

7.6 Closure

The objective of this session has been to introduce the process of code development and two-fluid modelling. The intent has been to provide an understanding of the principles involved, rather than to provide all the details. The two-fluid model in particular is not trivial, but it does appear to be necessary for faithful treatment of the physics observed in flow stratification cases.

7.7 References

[1] F.W. Barclay, D. Bean, R.E. Nieman, "RAMA: A Computer Code for Analysis of Transient Two-Phase Flow and Heat Transfer in Pipe Networks", Symposium on Simulation Modelling and Decision in Energy Systems, Montreal, 1978.

[2] B.N. Hanna, B.H. McDonald, D.A. Scarth, V.S. Krishnan, "Development and Application of a Two-Fluid Model for RAMA", Simulation Symposium on Reactor Dynamics and Plant Control, Sheridan Park, 1982.

[3] W.G. Mathers, R.L. Ferch, W.T. Hancox and B.H. McDonald, "Equations for Transient Flow-Boiling in a Duct", OECD/CSNI Specialists' Meeting on Transient Two-Phase Flow, Paris, 1978.