CSNI Report No 116 Addendum

Restricted



OECD

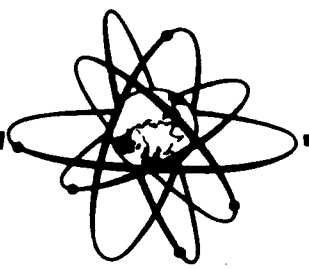
NEA

# COMPARISON OF THE PHYSICAL MODELS AND NUMERICAL METHODS IN THE CODES USED FOR THE GREST LWR CODE COMPARISON EXERCISE

B.H. McDonald

AECL/WNRE, Canada

1988



COMMITTEE ON THE SAFETY OF NUCLEAR INSTALLATIONS

OECD NUCLEAR ENERGY AGENCY

38, boulevard Suchet, 75016 Paris, France

# COMPARISON OF THE PHYSICAL MODELS AND NUMERICAL METHODS IN THE CODES USED FOR THE GREST LWR CODE COMPARISON EXERCISE

B.H. McDonald

AECL/WNRE, Canada



NUCLEAR ENERGY AGENCY
ORGANISATION FOR ECONOMIC CO-OPERATION AND DEVELOPMENT

## TABLE OF CONTENTS

**Figures**

# COMPARISON OF THE PHYSICAL MODELS AND NUMERICAL METHODS IN THE CODES USED FOR THE GREST LWR CODE COMPARISON EXERCISE

B.H. McDonald

## SUMMARY

In the 1985 GREST LWR Containment Aerosol Code Comparison Exercise, eleven different codes were run on combinations of three different postulated accident scenarios. The results submitted by the code exercise participants have been published. In this report, the physical models and numerical methods used in the various codes are examined. Physical models for aerosol particle agglomeration, particle removal by deposition on building structures, and steam condensation on aerosol particles, are reviewed and documented. There is considerable variation amongst these models within the various codes. The numerical methods used in the codes for discretizing and integrating the non-linear equations describing the physics, and including the physical models, are examined as special cases of a general solution procedure, the Method of Weighted Residuals. It would appear that all the numerical methods incorporated in the codes are able to generate reasonable solutions to the mathematical problem when used with care and skill. Also major differences in results produced by the various codes are thought to be due to the differences in the physical models which remain unresolved.

Atomic Energy of Canada Limited
Whiteshell Nuclear Research Establishment
Pinawa, Manitoba, Canada ROE 1LO
1988

# 1. INTRODUCTION

In 1985 a code comparison exercise was set up within the framework of CSNI's principal Working Group 4 by the Group of Experts on the Source Term (GREST) [1]. Three postulated accident scenarios relating to loss-of-coolant accidents in light water reactors formed the test cases for the exercise. The results submitted by the various participants have been documented [2]. The objective of the current exercise was to examine and document the physical models and numerical methods used in the codes. Similar comparisons have been performed before, between codes which describe the dry sodium combustion aerosols produced during hypothetical fast reactor accidents [3-9]. The results from those exercises are not completely applicable to LWR accident scenarios because of the presence of steam and water droplets in the containment atmosphere following a hypothetical LWR accident. In fact, the aerosols in such cases may consist primarily of water droplets due to steam condensation on aerosol particles. On the one hand the presence of water in the containment atmosphere complicates the analysis because of the need to consider steam condensation and Stephan flow, while on the other hand it does permit the aerosol particles to be treated as spheres (that is, all shape factors may be set equal to one).

Eleven different codes were run by participants in eight different countries on three hypothetical accident sequences. Table 1.1 lists the codes and the participants. Table 1.2 itemizes the three different accident scenarios modelled, and Table 1.3 gives the test case input parameters. Table 1.4 lists the result packages actually submitted by the participants.

The objective of this report, as a companion to the data report [2], is to describe and compare the physical models used in the codes in order to improve understanding of how the codes work. As well, the numerical methods employed in converting the physical models into computer code algorithms are reviewed and compared. It should be remembered that the comparisons are made on the basis of documentation submitted by the code runners [10-25] and current versions of these codes could be different and improved. The codes that were actually used to produce the results submitted are under review here, and in the absence of analytical, or exact, answers or hard experimental data for comparison it is not possible to say which codes produce the more accurate answers.

This report begins with a summary of the fundamental equations which the codes endeavor to solve, in Section 2. Basic physical properties, of both the carrier gas atmosphere and the aerosol particles are described in Section 3. Use of these properties in specifying total aerosol particle agglomeration rates is presented in Section 4, and in specifying particle removal rates in Section 5. Attention is given to the special models required to handle steam condensation in Section 6. Numerical methods used to recase the fundamental integro-differential equation as a system of ordinary differential equations are discussed in Section 7. Sections 8 and 9 represents some new work. In a continuing effort to understand more fully the models and numerics employed in containment aerosol codes, this author constructed a fairly simple "black-

box" aerosol model containing most of the physical models and both discrete bin and continuous log-normal representation for the particle mass dimension. By suitable selection of switches, this "black-box" model should be able to emulate any one of the codes under review. There was no real attempt to reproduce the results submitted; rather the goal was to assess qualitatively the performance of the various models within a controlled environment: differences observed in the performance of the "black-box" model might be useful in explaining differences reported by the code runners. This "black-box" model is described briefly in Section 8, and its performance in emulating the various codes under review on the SURRY AB test case is described in Section 9. Closing comments are recorded in Section 10.

The documentation available for this review was variable: from a code listing in one case to a full set of manuals in another. It has been a struggle to determine what exactly is in each code. Code users and/or developers might wish to amend some of the inevitable mistakes, oversights, and omissions that have occurred, and are to be noted in the following text and tables. The material included here has also been reported by the author at the OECD/CEC Workshop on Water-Cooled Reactor Aerosol Code Evaluation and uncertainty Assessment, Brussels, Belgium, 1987 Sept. 9-11 [31,32].

## 2. THE FUNDAMENTAL AEROSOL EQUATION

For a single aerosol species, a single equation of integro-differential form is used to describe the particle size distribution. If C(m,t) is the number of particles of mass m at time t, the rate at which this value changes with time is given by [3,8]

$$\frac{\partial}{\partial t} C(m,t) = \frac{1}{2} \int_0^m d\mu \; \phi(\mu, m-\mu) C(\mu,t) C(m-\mu,t) - C(m,t) \int_0^\infty d\mu \; \phi(\mu,m) C(\mu,t) - \frac{\partial}{\partial m} [\xi(m,t) C(m,t)] - R(m,t) C(m,t) + S(m,t) \qquad (2.1)$$

The first two terms on the right, the integrals, describe the process whereby small particles agglomerate to form larger ones. In the first term, all particles that combine to form new particles of mass m are added. The agglomeration kernel, $\phi(\mu, m-\mu)$, prescribes the rate at which particles of masses u and m - u combine to produce particles of mass m. The coefficient 1/2 is needed because the integral counts each agglomeration twice. In the second term, all particles of mass m that combine with all other particles through agglomeration are subtracted. In the third term, the growth of particles due to condensation of steam is prescribed. The condensation rate is $\xi(m,t)$. In the fourth term, the removal rate R(m,t) describes the rate at which particles of mass m are removed from the containment atmosphere by deposition on containment structure surfaces or by leakage to the environment. In the last term, new particles of mass m are injected into the containment atmosphere with the source rate S(m,t).

It is instructive to examine this equation in a little more detail. If only agglomeration is present (i.e., $\xi$, R and S vanish), the total airborne mass must be invariant:

$$\int_0^\infty dm\ m\frac{dC(m,t)}{dt} = \frac{d}{dt}\int_0^\infty dm\ mC(m,t) = 0. \tag{2.2}$$

This implies that the agglomeration process must be mass conservative:

$$\int_0^\infty dm\ m\{\frac{1}{2}\int_0^m d\mu\ \phi(\mu,m-\mu)C(\mu,t)C(m,t) - C(m,t)\int_0^\infty d\mu\ \phi(\mu,m)C(\mu,t)\} = 0. \tag{2.3}$$

Aerosol code developers must ensure that Equation (2.3) is satisfied, no matter what numerical methods they employ. Another observation may be made: with a non-zero agglomeration kernel $\phi$, in the limit as t approaches $\infty$, all particles will have agglomerated to form a single mass. Code developers who choose to impose an upper limit on particle mass should ensure that the largest expected mass can be accommodated.

Note that, in the absence of agglomeration, new particles or condensation (i.e., $\phi$, S, $\xi$ all vanish), the fundamental equation, (2.1), takes the familiar form of simple exponential decay:

$$\frac{d}{dt}C(m,t) = -R(m,t)C(m,t) \tag{2.4}$$

and with only new particles ($\phi$, R, $\xi$, all vanish), it takes the familiar form

$$\frac{d}{dt}C(m,t) = S(m,t). \tag{2.5}$$

When only condensation (or equivalently, evaporation) is present (i.e., $\phi$, R, S all vanish), the fundamental equation takes the form

$$\frac{\partial}{\partial t}C(m,t) = -\frac{\partial}{\partial m}[\xi(m,t)C(m,t)]. \tag{2.6}$$

This is very similar to the continuity, or mass balance equation, written in point form for one-dimension fluid flow in a uniform duct

$$\frac{\partial}{\partial t}\rho(x,t) = -\frac{\partial}{\partial x}[u(x,t)\rho(x,t)] \tag{2.7}$$

where $\rho$ is fluid density and u is fluid velocity; and the time rate of mass accumulation, $\partial\rho/\partial t$, is given by the negative of the spatial gradient of the mass flux, $-\partial(\rho u)/\partial t$. By analogy, in Equation (2.6), one may say that accumulation of particles of mass m through condensation is given by the negative of the mass gradient of a condensation flux defined by the condensation coefficient $\xi$. Alternatively, one may integrate Equation (2.6) over a small mass range:

$$\frac{d}{dt}\int_m^{m+\Delta m} C(m,t)dm = \xi(m,t)C(m,t) - \xi(m+\Delta m,t)C(m+\Delta m,t). \tag{2.8}$$

This equation shows that the time rate of change in the number of particles between masses m and $m + \Delta m$ is given by the number of particles that enter the region by growth by condensation at mass m less the number that leave the region by growth by condensation at mass $m + \Delta m$.

A further observation with regard to the time integration of the fundamental equation may be made. Depending upon the instantaneous values of the agglomeration rate, $\phi$, the condensation rate, $\xi$, the removal rate, R, and the source term, S, the equation to be integrated may be essentially of simple form, Equation (2.5), exponential decay form, Equation (2.4), or of convective form, Equation (2.6), or integro-differential form. A viable integration procedure should be able to handle each of these forms. The convective form is particularly difficult and often requires the use of one-sided, or upstream, discrete difference operators.

For those codes that have true multicomponent capability, e.g., CONTAIN and AEROSIM-M, it is necessary to specify an equation like (2.1) for each component. If $q_k(m,t)$ is the mass of component k on particles of mass m at time t, the multicomponent aerosol population balance equation, analogous to Equation (2.1) is

$$\frac{\partial}{\partial t} q_k(m,t) = {}_0\int^m d\mu\ \phi(m-\mu,\mu)q_k(\mu,t)C(m-\mu,t) - q_k(m,t)\ {}_0\int^\infty d\mu\ \phi(m,\mu)C(\mu,t) + \delta_{ik}\xi(m,t)C(m,t) - \frac{\partial}{\partial m}[\xi(m,t)q_k(m,t)] - R(m,t)q_k(m,t) + S_k(m,t). \tag{2.9}$$

Here, C(m,t) remains the total number of aerosol particles of mass m. If there are L species,

$$\sum_{k=1}^{L} q_k(m,t) = mC(m,t). \tag{2.10}$$

All terms are equivalent to those preceding except for the term $\delta_{ik}\ \xi(m,t)\ C(m,t)$. The Kronecker delta function $\delta_{ik}$ is unity when k = i, defined as the condensing component, water, and is zero for other components. This term corresponds to the change in the airborne mass of this component. In other words, it defines the condensation of steam on aerosol particles of mass m and is separate from the familiar differential term that corresponds to the shifting of mass up (or down) the mass range due to condensation (or evaporation) [8].

For the most part, multicomponent numerical methods are similar to single-component methods, and without loss in generality, it is sufficient to examine single-component models. As well, some of the codes, for example CONTAIN, have multi-compartment capabilities, and this requires an extension of the single-volume equations described here to permit the inter-compartment flow of carrier gas and aerosol materials. Once again, the single-volume representation may be used as the model equation without loss in generality.

In AEROSOL/B1, the concept of two distinct sources and their mixing is introduced to provide a means of averaging particle densities.

AEROSIM-M is a fully multicomponent code, but does not include the effects of condensation of steam on the aerosol particles. On the other hand, AEROSIM-W does include steam condensation on aerosol particles, but does not have multicomponent capability.

These codes calculate physical aerosol processes, and only these processes are examined. Chemical effects, such as iodine chemistry, are not included.

## 3. BASIC PHYSICAL PROPERTIES

The characteristics of the carrier gas, normally air, or a mixture of air and steam, within the containment building are needed for specification of aerosol agglomeration and removal rates. The carrier gas temperature (T) and pressure (P) are usually assumed to be input data for the aerosol codes, except in CONTAIN where they result from a thermalhydraulic calculation. The gas density ($\rho_g$) is usually calculated using the ideal gas law. The gas mean free path length ($\lambda$) may be calculated from the temperature, density and molecular weight of the gas. The gas viscosity ($\eta$) may be calculated from the gas temperature. The carrier gas turbulent energy dissipation rate ($\varepsilon_T$) is usually given as code input data. Tables 3.1 through 3.3 summarize code specific formulas for the determination of these properties.

Aerosol particles also possess properties required for specification of the agglomeration and removal processes. As wet aerosol particles are assumed to be spherical (all shape factors equal to one), the particle radius (r), mass (m) and density ($\rho$) are very simply related by $m = (4\pi\rho r^3)/3$. The particle Knudsen number (Kn) is the gas mean free path length divided by the particle radius. The Cunningham slip correlation (Cu) is then determined by the various codes as a function of the Knudsen number (Table 3.4). The particle mobility (B) is then obtained as

$$B = \frac{Cu}{6\ \pi r \eta \chi} \cdot \tag{3.1}$$

The dynamic shape factor $\chi$ is set equal to unity for wet aerosols. The particle mobility is simply Stoke's Law amended by the Cunningham correction for small particles.

The particle terminal settling velocity ($V_T$) is calculated from the particle density, radius, and gas viscosity using

$$V_T = \frac{2\rho r^2 g}{9\eta\chi} Cu \tag{3.2}$$

where g is the gravitational acceleration.

Note the use of the Cunningham slip correlation in Equation (3.2) to correct for small particles. A correction for large particles due to Klyachko is available in RETAIN [16]. It involves identifying the drag coefficient ($C_D$) in (3.2) through the particle Reynolds number (Re) and replacing $C_D = 24/Re$ with the expression $C_D = 24/Re + 6/Re^2$.

The particle diffusivity, or diffusion constant, (D) is obtained as

$$D = kTB \tag{3.3}$$

where k is Boltzmann's constant, and the temperature (T) is in K.

NAUA makes use of an effective density for calculating aerosol behaviour in steam. All particles are treated as porous spheres with this effective density; therefore, all shape factors can be set to unity. This effective density is the same as the actual density, because the wet aerosol particles may be treated as being spherical.

## 4. AEROSOL PARTICLE AGGLOMERATION

The agglomeration kernel $\phi(m,n)$, introduced in Section 2, prescribes the rate at which particles of masses m and n will agglomerate to form new particles of mass m + n. A recent paper by Simons et al. [26] suggests that $\phi$ should be calculated as the consequence of the net forces acting on the aerosol particles. All codes under review here take a different approach, which is simpler. Separate agglomeration rates are calculated in the codes for gravitational agglomeration, Brownian (or diffusive) agglomeration, and turbulent agglomeration (both shear and inertial), and then these separate agglomeration rates are combined to produce the total agglomeration rate. The code-specific methods for combination are summarized in Table 4.1. When any one of these separate effects completely dominates the others, there is no doubt the procedures used in the various codes are satisfactory. It is when more than one mechanism contributes substantially to the total agglomeration rate that the methods for combining the separate effects come into doubt.

The gravitational agglomeration rate, $\phi_G(m,n)$, is proportional to the difference in particle settling velocities ($V_{Tm}$, $V_{Tn}$):

$$\phi_G(m,n) = \pi\varepsilon(m,n)(r_m + r_n)^2\gamma^2|V_{Tm} - V_{Tn}|S_T. \qquad (4.1)$$

The sticking parameter ($S_T$) and the agglomeration shape factor ($\gamma$) are both set to unity for wet aerosols. The radii of the two particles are $r_m$ and $r_n$, and the collision efficiency ($\varepsilon(m,n)$) is given by

$$\varepsilon(m,n) = C_\varepsilon\left[\frac{r_m}{r_m + r_n}\right]^2 \quad r_m < r_n \qquad (4.2)$$

where the coefficient $C_\varepsilon$ depends upon the collision model used:

$$C_\varepsilon = \begin{cases} 0.5 & ; \quad \text{Pruppacher-Klett} \\ 1.5 & ; \quad \text{Fuchs.} \end{cases} \qquad (4.3)$$

The basis for each of these models is examined in detail by Dunbar and Fermandjian [4], and they also review more recent attempts than those of Pruppacher-Klett and Fuchs to specify the collision efficiency. They conclude: "Gravitational agglomeration rates are sensitive to the choice of collision efficiency, so this question is one of the most important areas of uncertainty in current aerosol physics."

The AEROSOLS/B1 model considers the possibility that particles of different sizes will have different densities and includes both particle densities explicitly, although they are implicitly included in calculation of the terminal settling velocities here.

In AEROSIM [11], one has several options for selecting the collision efficiency. In addition to the two models here, one may input his own value or one may select a numerical fit due to Loyalka [30] or a truncated form of the Pruppacher-Klett formula [28], or a variation known as Klett-Davis. This rather wide set of possibilities suggests a degree of uncertainty.

The standard formula in all the codes for the Brownian (or diffusional) agglomeration rate ($\phi_B(m,n)$) is determined from the particle diffusivities ($D_m$, $D_n$) and the particle radii:

$$\phi_B(m,n) = 4\pi(D_m + D_n)(r_m + r_n)\,\gamma S_T. \tag{4.4}$$

In CONTAIN alone [24], the Brownian agglomeration rate is multiplied by a non-isotropic-gas correction factor $F_F$, originally formulated by Fuchs:

$$F_F(m,n) = \left[1 + \frac{4(D_m + D_n)}{\bar{V}(r_m + r_n)}\right]^{-1} \tag{4.5}$$

with

$$\bar{V} = \left[\frac{8kT}{\pi}\left(\frac{1}{m} + \frac{1}{n}\right)\right]^{1/2} \tag{4.6}$$

Turbulent agglomeration is described in two parts. Shear agglomeration ($\phi_{TS}(m,n)$) is modelled after the formula of Saffman and Turner [4]:

$$\phi_{TS}(m,n) = \varepsilon(m,n)(r_m + r_n)^3\gamma^3\left[\frac{8\pi\rho_g\varepsilon_T}{15\,\eta}\right]^{1/2} S_T \quad . \tag{4.7}$$

As indicated earlier, the turbulent energy dissipation rate is usually specified by the code user as input data. Inertial turbulent agglomeration is modelled using

$$\phi_{TI}(m,n) = 0.148\varepsilon(m,n)|V_{Tm} - V_{Tn}|\gamma^2(r_m+r_n)^2\left[\frac{\rho_g\varepsilon_T^3}{\eta}\right]^{1/4} S_T \; . \tag{4.8}$$

It has been noticed that in documentation available for MAEROS [24], the aerosol behaviour model in CONTAIN, the collision efficiency, $\varepsilon(m,n)$, is not included in either Equations (4.7) or (4.8), i.e., it is taken to be unity. Also, in HAA-4 [15], it may be set to unity by the user for turbulent shear agglomeration and, in fact, must be constant for use of the Klyachko correction to the terminal settling velocity for large particles. As noted earlier, the agglomeration shape factor $\gamma$ and the sticking parameter are usually specified to be unity for wet aerosols.

The codes under review generally treat agglomeration in a similar fashion, except for some variation in the methods of combining the individual rates to produce the overall agglomeration rate. It would be useful to pursue the work reported by Simons [26] and attempt to define a method for combining the physical effects to produce an overall, integrated agglomeration rate.

There is still some question about shape factors, and many various forms have been studied for use with dry aerosols [3,4]. Wet aerosols permit the use of shape factors equal to unity, so that there may not be any problem with application for water-cooled reactors.

## 5. AEROSOL PARTICLE REMOVAL

The aerosol particle removal rate, R(m,t), introduced in Section 2, prescribes the rate at which particles are removed from the containment atmosphere by deposition on structural surfaces or by leakage through the containment building to the environment. The total removal rate is usually constructed as the sum of removal rates calculated from each removal mechanism:

$$R(m,t) = R_G(m,t) + R_B(m,t) + R_T(m,t) + R_D(m,t) + R_L(m,t) \qquad (5.1)$$

where

$R_G(m,t)$ is gravitational deposition,
$R_B(m,t)$ is Brownian or diffusional deposition,
$R_T(m,t)$ is thermophoretic deposition,
$R_D(m,t)$ is diffusiophoretic deposition, and
$R_L(m,t)$ is the containment building leak rate.

Although it has been customary to combine the removal effects in this fashion, it may be more valid to consider the net result of the effects acting together, as suggested by Simons [26] for agglomeration.

Gravitational deposition on upward facing surfaces is expressed in terms of the particle settling velocity $V_{Tm}$

$$R_G(m,t) = \frac{A}{V} V_{Tm} \qquad (5.2)$$

where the ratio A/V is the total upward-facing surface area divided by the containment volume V. Note that this neglects bulk turbulence.

Diffusional, or Brownian, deposition on all internal surfaces is expressed in terms of the particle diffusivity $D_m$

$$R_B(m,t) = \frac{A}{V} \frac{D_m}{\delta_B} \quad . \qquad (5.3)$$

Here A is the total surface area available for Brownian deposition and $\delta_B$ is the diffusive boundary layer thickness. For many calculations, this thickness is simply specified as

$$\delta_B = 0.1 \text{ mm}. \qquad (5.4)$$

In AEROSOLS/B1 [20], diffusive deposition through turbulence and impaction through turbulence are modelled primarily for the "circuit" option, where the code is used to model the primary heat transport system.

Thermophoretic deposition arises from the movement of particles in a warm carrier gas to a cooler wall, the movement induced by the temperature gradient at the wall. In applying the well-mixed aerosol assumption, all gradient information was lost. The temperature gradient at the wall, $\nabla T$, is usually reconstructed by an empirical calculation

$$\nabla T = \frac{T - T_w}{\delta_T} \tag{5.5}$$

where $T_w$ is the wall temperature, T is the carrier gas bulk temperature, and $\delta_T$ is a thermophoretic boundary layer thickness, often prescribed to be

$$\delta_T = 2.0 \text{ mm}. \tag{5.6}$$

The thermophoretic removal rate is then usually calculated using Brock's equation [4]

$$R_T(m,t) = \frac{A}{V} \nabla T \frac{3\eta}{2\chi} Cu(m) F_B(m) \tag{5.7}$$

where $F_B$, the Brock factor, is given by

$$F_B(m) = \frac{K_B}{1 + 3C_M Kn(M)} \cdot \frac{R_T + C_t Kn(m)}{1 + 2\{R_T + C_t Kn(m)\}} \tag{5.8}$$

In this equation Kn(m) is the Knudsen number, $R_T$ is the ratio of carrier gas to aerosol material thermal conductivities and $K_B$, $C_M$, and $C_t$ are constants, summarized for various code applications in Table 5.1. In Equation (5.7), A is the surface area available for thermophoretic deposition, Cu is the Cunningham slip relation, $\eta$ is the carrier gas viscosity and $\chi$ is the dynamic shape factor, here prescribed to be unity because of the wet aerosol assumption. Note that if there are several surfaces, each at different temperatures ($T_w$), the total thermophoretic removal rate may be obtained by summing the contributions made by calculating Equation (5.7) for each surface.

Diffusiophoretic deposition is modelled to account for those aerosol particles that are dragged to a wall by the net gas flow, and towards the wall when steam is condensing upon it. The rate is independent of the radius of the aerosol. The usual approach uses the equation

$$R_D(m,t) = \{\frac{A}{V}\} \frac{(m_v)^{1/2}}{\gamma_v(m_v)^{1/2} + \Sigma \gamma_j(m_j)^{1/2}} \{\frac{D_v}{P_a}\} \nabla P_v \tag{5.9}$$

where $m_j$ and $\gamma_j$ are the molar mass and fraction of component j (v for steam), $P_a$ is the containment pressure, $D_v$ is the steam diffusion coefficient in the mixture of the other components, and $\nabla P_v$ is the steam pressure gradient in the gas, calculated as

$$\frac{dP_v}{dx} = \frac{P_v - P_s(T_w)}{D_v(C_\infty - C_w)J_v^{-1}} \cdot \tag{5.10}$$

In this relation, $P_v$ is the containment steam pressure, $P_s(T_w)$ is the saturation pressure at the wall temperature, $C_\infty$ and $C_w$ are the steam concentrations in the containment and at the wall, and $J_v$ is the condensation flux. This is the formula used in AEROSOLS/B1. RETAIN uses a form based on the original TRAP-MELT model, formulated by Gormley and

Kennedy [21], and appears to include effects due to resuspension and vapour pressure lowering, due in turn to formation of aqueous solutions during steam condensation.

An instructive paper by Dunbar [8] leads to a somewhat simpler result

$$R_D(m,t) = \frac{P_s}{P} \cdot \frac{W}{m_s} \qquad (5.11)$$

where $P_s$ is the partial pressure of steam, P is the total pressure, W is the total mass rate of steam condensation and $m_s$ is the total mass of steam in the containment atmosphere. Assumptions inherent in Equation (5.9) are the perfect gas law for both steam and the carrier gas (not true for steam near saturation) and the dependence of temperature and pressure on distance from the wall. RETAIN-S uses the same formula. Note that the total mass rate of steam condensation needs to be obtained, usually from a separate thermalhydraulic calculation. It is to be noted [22] that this approach neglects the "diffusiophoretic" retardation of the transport of the particles to the surface resulting from the buildup of non-condensible gases near the surface.

The leak rate $R_L(m,t)$ will usually be prescribed as a boundary condition. For most cases, it is specified as a certain volume percent per day.

Removal mechanisms are modelled in much the same fashion by all the codes, except for thermophoresis, where different coefficients are used in the calculation of the Brock factor and for diffusiophoresis. Selection of appropriate boundary layer thicknesses could continue to be a point of some uncertainty. In the recent code comparison exercise it was specified that all code runners use the same boundary layer specifications. However, there remains much room for discussion, and perhaps some consensus, as to what values, or formula to generate values, should be used in reconstructing gradients near walls.

## 6. STEAM CONDENSATION MODELLING

The presence of steam and water droplets in the containment after a LOCA in a water-cooled reactor changes the physics considerably. Rather than having to consider only the dynamics of dry, fluffy aerosol particles of the condensed fission product and structural materials, one has to deal with the physics of water droplet growth (or shrinkage) due to steam condensation (or evaporation). The problem is essentially a thermalhydraulics one, because bulk condensation or evaporation rates will be determined by steam saturation or supersaturation conditions in the containment atmosphere, as will steam condensation on the cooler containment building walls and internal structures. In fact, one could address the problem almost without regard for the fission product and structural aerosol materials, as they could be considered to be entrained in the water droplets and would only marginally change the effective density of the droplets from the nominal "water-only" value. In any case, the aerosol particles will be wet.

A view of the problem as just stated leads naturally to an approach like that used in CONTAIN. One first considers that the essential physics is thermalhydraulics and treats primarily the growth, agglomeration and deposition of water droplets in a dynamic steam-air atmosphere. Second, these droplets contain various fission product and structural materials, which are "along for the ride", so a code has to account for, or keep track of, the amounts of these materials in the water droplets during the calculation. This would seem to be the proper approach to the problem: a fully coupled, thermalhydraulics-aerosol, multicomponent, dynamic model. At the time of this review, only the CONTAIN code had taken that approach.

The other codes appear to have been written originally to analyze fast reactor accidents: dry aerosol particles in a dry, air-filled containment building. To handle water-cooled reactor cases, these codes have been amended to include some of the additional physical effects associated with the presence of steam in containment. There appears to be considerable variation amongst these codes as to how these physical effects have been included. In one case, the Stone and Webster version of NAUA-4, diffusiophoretic particle removal appears to be included by simply increasing the containment leakage rate [25]. In most cases, the time-dependent bulk-steam condensation rate is read as input data, and the codes split this condensation between the airborne particles and the building walls. However, this is done differently, even in different versions of the same code, notably the EPRI and Stone and Webster versions of NAUA-4. The GREST exercise data report [2] states that, "SWEC partitions the total condensation rate between the wall and the particles according to the supplied input data. This is consistent with the way NAUA, in general, was conceived (input thermalhydraulic data). EPRI attempts to calculate mechanistically the condensation rates on the wall and the particles, keeping the total equal to the input. The effect of this difference, so far, has been that SWEC calculates considerably more condensation on the particles than does EPRI."

For the purpose of defining an appropriate procedure, the objective is straightforward: if a steam condensation rate is known, however it is obtained, the code is to calculate the condensation coefficient $\xi(m,t)$ required for aerosol particle growth in Equations (2.1) or (2.5), and produce the data required for the diffusiophoretic particle removal rate, Equation (5.9). Evaporation may also be included, as simply negative condensation.

An equation for the condensation rate coefficient has been derived by Clement, as reported by Dunbar [8]. For a spherical particle of radius r, the Mason equation for the condensation rate may be written as

$$\xi(r) = 4\pi D \rho_{s\xi}(T) \frac{C_n}{C_n + S_o(r)} [S - S_o(r)]. \qquad (6.1)$$

For a spherical drop of water, density $\rho_w(T)$, the equation can be expressed in terms of rate of change of radius

$$\frac{dr}{dt} = A \frac{S - S_o(r)}{r} \tag{6.2}$$

where $$A = D \frac{\rho_{sE}(T)}{\rho_w(T)} \frac{C_n}{C_n + S_o(r)} \tag{6.3}$$

In these equations, the following definitions apply:

$\rho_s(T)$ = density of steam in the bulk gas,
$\rho_{sr}$ = density of steam at the droplet surface,
$\rho_{sE}(T)$ = density of steam in equilibrium with a plane surface of pure water at temperature T,
T is the temperature of the bulk gas,
$T_r$ is the temperature at the droplet surface,
D is the diffusivity of steam through air,

$S = \frac{\rho_s(T)}{\rho_{sE}(T)}$ bulk saturation ratio, and

$S_o(r) = \frac{\rho_{sr}}{\rho_{sE}(T_r)}$ saturation ratio at the droplet surface.

A dimensionless "condensation number" was defined by Clement (and reported by Dunbar [8]) to be

$$C_n = \frac{K}{LD\rho'_{sE}(T)} \tag{6.4}$$

where L is the latent heat of evaporation of water, K is the gas thermal conductivity, and $\rho'_{sE}(T)$ is the temperature derivative of the equilibrium vapour density.

As reported by Dunbar [8], the rate at which droplets grow is impressive: "a supersaturation (relative to the drop itself) of only 1% causes a 1 μm radius drop to grow at an instantaneous rate of 4.65 μm per second." It is apparent that proper thermalhydraulics analysis needs to be in place. Dunbar produces a strong and convincing argument for integrated thermalhydraulics-aerosol behaviour analysis. The essence is this: thermalhydraulics codes, which are not coupled to an aerosol calculation, assume implicitly that there always are sufficient nucleation sites to ensure instantaneous equilibration. In that case, condensation always occurs, preventing the atmosphere from becoming supersaturated, and that subsaturation can happen only when there is no water left to evaporate; if water is present in the atmosphere, S = 1. Dunbar states [8]: "Condensation and evaporation are too tightly coupled to the rest of the thermalhydraulics to be separated out into a separate calculation."

In the AEROSIM-W code, the total condensation rate is supplied as input, and the code distributes this condensation over the particle size distribution and then follows the subsequent aerosol behaviour. If the total condensation rate is $Y_T(t)$, it may be expressed in terms of the rate on a single particle as

$$Y_T(t) = V \int_0^\infty dm\ \xi(m,t)C(m,t). \tag{6.5}$$

Separation of variables is assumed, giving

$$\xi(m,t) = \xi(0,t)r(m) \tag{6.6}$$

where the particle radius is considered to be only a function of mass. The distribution of condensation over the range of particle sizes then becomes

$$Y(m,t) = \frac{Y_T(t)r(m)C(m,t)}{\int_0^\infty dm\ r(m)C(m,t)}. \tag{6.7}$$

Note that Y(m,t)dm is the condensation rate on all particles sized between m and m + dm.

NAUA uses a more complex form of the Mason equation, including the effects of curvature, with a condensation form factor. In the NAUA code series, particle volume, rather than particle mass is used as the basic measure of particle size. In the NAUA strategy, two time steps are used. An outer step is used for removal and agglomeration processes, and an inner step for changes due to condensation; in fact, the outer step is just divided into finer inner steps. At each step NAUA requires the steam input rate per unit volume, which in practice [8] is the total condensation rate per unit volume ($Y_T(t)/V$). NAUA calculates the saturation ratio from the input, which also includes the temperature of the atmosphere, and then the change in particle volume due to condensation. NAUA also keeps track of the mass of airborne steam.

AEROSOLS/B1, unlike the other codes, is able to handle condensation on soluble aerosol particles. This code, like NAUA, uses particle volume, rather than mass, as a measure of size. Input consists of a factor related to the saturation ratio S (in fact, the term is 1 - S). When S is less than one, the solute effect may play an important role in determining the amount of water on aerosol particles. In this aspect, AEROSOLS/B1 is more advanced than the other codes. AEROSOLS/B1 is often run iteratively with, the thermalhydraulics code, JERICHO [8], in an attempt to couple thermalhydraulics and aerosol behaviour.

The CONTAIN algorithm is quite complex [8], because it is both a thermalhydraulic model and an aerosol model combined in an integrated fashion. The condensation rate $\xi(m,t)$ is obtained in a form similar to (6.1).

The method of operation of the separate aerosol codes is to rely on input data for the bulk condensation rate, or the saturation ratio, in the containment atmosphere, and then to distribute this condensation over the airborne aerosol particles, thus increasing their sizes. As well, they rely on input data for the condensation rates of steam on internal containment building structures, from which data the diffusiophoretic modelling is carried out. This input data is usually obtained as the output from a thermalhydraulics code. Because water droplet growth rates from condensation can be large, analysis of how the codes treat wet aerosol particle behaviour cannot be decoupled from the thermalhydraulics calculation, which produces the condensation rates.

In a recent comparison of European aerosol codes, which contain condensation physics [9], large discrepancies were observed to exist in prediction of mass concentration and size of suspended particles resulting from differences of steam condensation modelling. The uncertainties remain.

## 7. NUMERICAL METHODS

The approach taken in this section will be somewhat abstract. Although each code contains very specific calculations for generating algebraic expressions to approximate the integro-differential equations, and these could be simply recorded, it could be more helpful to take a longer view and look more generally at what the codes are trying to do. It is possible to look at all the methods employed as variations of a common theme, and this is what this section will try to do.

The objective is to obtain a solution to the particle size distribution, either Equation (2.1) or (2.9). At the risk of being unfair to those codes which have multicomponent and/or multicompartment capabilities, only the single-component, single-compartment representation, Equation (2.1) is considered in detail here. In only very special cases can analytical solutions to these equations be obtained. In practice it is necessary to resort to numerical methods and to use a computer. The process involves defining a sequence of algebraic operations that produce numbers approximating the true solutions to the equations.

The process can be viewed as a two-part procedure. The first step is to define an algorithm to convert the integro-differential equation in mass (m) and time (t) to a system of coupled ordinary differential equations in time. The second step is to define an algorithm to integrate this system of ordinary differential equations.

Although many of the codes base numerical representation on particle mass, some, for example, NAUA, REMOVAL, and AEROSOLS/B1, choose particle volume instead of mass. This requires some rewriting of the basic equations, but there is no significant structural change. Because it is an objective of this review to compare the numerical strategies used in the codes, this section will proceed as though all codes chose to base the numerical methods on the mass variable. Although this might render invalid certain specific terms in the text that follows, the general relationships should be valid, and the sense of the various numerical approaches should be preserved.

To facilitate comparison of the methods used in the various codes, the first step in the process will be described within the general method of weighted residuals [27]. Briefly, given an integro-differential equation of the form of (2.1), which may be expressed as

$$\frac{\partial}{\partial t} C(m,t) = F[C(m,t),m,t] \qquad (7.1)$$

where F is the right side of Equation (2.1). One defines a residual $\Lambda(m,t)$ as

$$\Lambda(m,t) = F[C(m,t),m,t] - \frac{\partial}{\partial t} C(m,t) \tag{7.2}$$

and then integrates the residual, weighted by a function W(m), over the m domain

$$I(t) = \int_0^\infty dm\ W(m)\Lambda(m,t) = \int_0^\infty dm\ W(m)\ F[C(m,t),m,t]\ \frac{\partial}{\partial t}C(m,t)\ . \tag{7.3}$$

The argument is that, with suitable W(m), if I(t) vanishes for all t or is forced to vanish for all t, then Equation (7.1) is solved.

In practice, a form is assumed for the unknown C(m,t) that involves a function of N time-only dependent variables $Z_i(t)$:

$$C(m,t) = f[m,\underline{Z}(t)] \tag{7.4}$$

where $\underline{Z}(t)$ is the vector of the variables $Z_i(t)$, expressed when transposed as

$$\underline{Z}(t) = \{Z_1(t),\ Z_2(t),\ \ldots,\ Z_N(t)\}. \tag{7.5}$$

If the $Z_i(t)$ variables are known, then C(m,t) can be constructed from (7.4). The N required ordinary differential equations are obtained by using N different weighting functions $W_i(m)$ in the equation

$$\int_0^\infty dm\ W_i(m)f(m,\frac{dZ}{dt}) = \int_0^\infty dm\ W_i(m)F[f(m,\underline{Z}),m,t]\ ;\quad i = 1,\ldots N. \tag{7.6}$$

Note that this assumes the integrated residual, I(t) in Equation (7.3), vanishes throughout. This can be recast, with linear functions f, in an N x N system, represented as the matrix-vector equation:

$$[A]\frac{d\underline{Z}}{dt} = \underline{b}(\underline{Z},t) \tag{7.7}$$

which can then be integrated, for example, by the forward Euler method:

$$\underline{Z}(t + \Delta t) = \underline{Z}(t) + \Delta t[A]^{-1}\underline{b}[\underline{Z}(t),t]. \tag{7.8}$$

Alternatively, one can define $\underline{Z}^*$ by

$$\frac{d\underline{Z}^*}{dt} = [A]\frac{d\underline{Z}}{dt} = \underline{b}(\underline{Z},t) \tag{7.9}$$

whence

$$\underline{Z}^*(t + \Delta t) = \underline{Z}^*(t) + \Delta t\underline{b}[\underline{Z}(t),t]. \tag{7.10}$$

All of the methods used in the codes to produce their systems of ordinary differential equations can be regarded as methods of weighted residuals with suitable choices of the variables $\underline{Z}(t)$ (or $\underline{Z}(t)$ and $\underline{Z}^*(t)$), the functions f and the weighting functions $W_i(m)$.

In the following part of this section, the various choices used in the codes under review are described. To be specific, the terms to be examined from Equations (7.6) and (2.1) are:

$$\int_0^\infty dm\ W_i(m) f(m, \frac{d\underline{Z}}{dt}) = \sum_{j=1}^{5} I_j \quad ; \quad i = 1,..N \tag{7.11a}$$

where

$$I_1 = \int_0^\infty dm\ W_i(m)\ \frac{1}{2} \int_0^m d\mu\ \phi(\mu, m-\mu) f(\mu, \underline{Z}) f(m-\mu, \underline{Z}) \tag{7.11b}$$

$$I_2 = \int_0^\infty dm\ W_i(m)\ -f(m, \underline{Z}) \int_0^\infty d\mu\ \phi(\mu, m) f(\mu, \underline{Z}) \tag{7.11c}$$

$$I_3 = \int_0^\infty dm\ W_i(m)[S(m,t)] \tag{7.11d}$$

$$I_4 = \int_0^\infty dm\ W_i(m)[-R(m,t) f(m, \underline{Z})] \tag{7.11e}$$

$$I_5 = \int_0^\infty dm\ W_i(m)\ \{-\frac{\partial}{\partial m}[\xi(m,t) f(m, \underline{Z})]\} \tag{7.11f}$$

## 7.1 METHODS OF MOMENTS CODES

In the methods of moments codes--HAA-4, HAARM, RETAIN-S, RETAIN-2C --a lognormal distribution is assumed:

$$C_L(m,t) = \frac{C_T(t)}{[2\pi]^{1/2} 3\alpha_g(t)\ m} \exp\{-\frac{1}{2}\ [\frac{\ln \frac{m}{m_g(t)}}{3\alpha_g(t)}]^2\} \tag{7.1.1}$$

There are 3 time-only dependent variables:

$Z_1 = C_T(t)$, the total number density;

$Z_2 = m_g(t)$, the geometric mean mass; and

$Z_3 = \alpha_g(t)$, the log of the geometric standard deviation.

The three equations to be constructed use the first three moments, which means $W_i = m^{i-1}$, with i = 1,2,3. The vector of unknowns, $\underline{Z}^*$, becomes simply the vector of the moments. From Equation (7.6),

$$\frac{d\underline{Z}_i^*}{dt} = \int_0^\infty dm\ m^{i-1} F[C_L(m, \underline{Z}), m, t] \quad ; \quad i = 1,2,3. \tag{7.1.2}$$

After obtaining $\underline{Z}^*(t+\Delta t)$, one may recover the new distribution. Defining a vector $\underline{X}$ to be the moments and following convention,

$$\underline{X} = \underline{Z}^*(t + \Delta t) \tag{7.1.3}$$

then

$$C_T(t + \Delta t) = X_1 \tag{7.1.4a}$$

$$m_g(t + \Delta t) = \frac{X_2^2}{[X_1^3 X_3]^{1/2}} \tag{7.1.4b}$$

$$\alpha_g(t + \Delta t) \frac{1}{3} [\ln(\frac{X_1 X_3}{X_2^2})]^{1/2} \tag{7.1.4c}$$

Agglomeration, source, and removal integrals are handled by expanding the agglomeration kernel, $\phi$, and the removal rate, R, as power series in the size coordinates (m and $\mu$). This effectively recasts all terms as higher order moments and permits the analytical results to be simply reapplied. A major difficulty with the method is the need to represent agglomeration and removal rates as finite series expansions: this introduces errors (of truncation) and makes it difficult to change or modify the rates.

## 7.2 DISCRETE FINITE-DIFFERENCE METHODS

In these methods, the mass coordinate range between the smallest expected mass, $m_o$, and the largest expected mass, $m_N$, is divided into N discrete subdivisions. The mean mass is

$$M_i = \frac{m_{i-1} + m_i}{2} \quad ; \quad m_{i-1} < m_i \quad ; \quad i = 1,..N \tag{7.2.1}$$

with $$h_i = m_i - m_{i-1} \quad ; \quad i = 1...N \tag{7.2.2}$$

and the ith interval $A_i$ (or "bin") is defined by

$$A_i = [m_{i-1}, m_i) \; ; \; i = 1...N. \tag{7.2.3}$$

The particle densities at the bin midpoints become the time-only dependent variables

$$C(m,t) = C(M_i,t) = Z_i(t) \quad ; \quad i = 1,...N. \tag{7.2.4}$$

To represent the methods used in CONTAIN, AEROSIM and REMOVAL, assuming that the mass variable is retained, one may define the weighting functions as

$$W_j(m) = \begin{cases} m & ; \; m_{j-1} \le m < m_j \\ 0 & ; \; \text{otherwise} \end{cases} \tag{7.2.5}$$

and for the NAUA family of codes

$$W_j(m) = \begin{cases} 1 & ; \; m_{j-1} \le m < m_j \\ 0 & ; \; \text{otherwise.} \end{cases} \tag{7.2.6}$$

For the NAUA versions, one may define the actual variables as

$$Z_i^* = h_i Z_i \tag{7.2.7}$$

and for AEROSIM

$$Z_i^* = Z_i/M_i. \tag{7.2.8}$$

Application of the method of weighted residuals to the basic equation--leaving out for the moment the condensation term (7.11f)--results in the system of equations where $A_i$ is the range of integration defined by Equation (7.2.3),

$$\int_{Ai} dm\ m^L \frac{dZ_i(t)}{dt} = \int_{Ai} dm\ m^L F[Z_i(t),m,t]\ ;\quad i = 1\ldots N \tag{7.2.9a}$$

where

$$L = \begin{cases} 0 & \text{for NAUA versions} \\ 1 & \text{for all others.} \end{cases} \tag{7.2.9b}$$

A result proved in [4], which states that

$$\int_{A_i} dm\ m^L \int_0^m d\mu\ F(\mu, m-u) = \int_0^{m_i} d\mu \int_0^{m_i} d\nu\ (\mu+\nu)^L \theta_i(\mu+\nu) F(\mu,\nu) \tag{7.2.10a}$$

where

$$\theta_i(\mu+\nu) = \begin{cases} 1 & \text{for } m_{i-1} \le \mu+\nu < m_i \\ 0 & \text{otherwise} \end{cases} \tag{7.2.10b}$$

allows Equation (7.2.9a) to be expanded as follows:

$$\begin{aligned} \frac{d}{dt} \int_{Ai} dm\ m^L Z_i(t) = {} & \tfrac{1}{2} \sum_{j=1}^{i} \sum_{k=1}^{i} \int_{A_j} d\mu \int_{A_k} d\nu\ (\mu+\nu)^L\ \theta_i(\mu+\nu)\phi(\mu,\nu) Z_j Z_k \\ & - \sum_{j=1}^{N} \int_{A_i} dm\ m^L Z_i \int_{A_j} d\mu\ \phi(\mu,m) Z_j \\ & + \int_{Ai} dm\ m^L S(m,t) - \int_{Ai} dm\ m^L R(m,t) Z_i. \end{aligned} \tag{7.2.11}$$

Condensation is most readily included by using an upwind differencing scheme [8], resulting, for example, in the addition to the right side of (7.2.11) of $I_5$, where

$$I_5 = -\int_{A_i} dm\ m^L \nabla_\xi z \tag{7.2.12a}$$

$$\nabla_{\xi z} = \begin{cases} \dfrac{\xi(M_i)Z_j - \xi(M_{i-j})Z_{i-1}}{m_i - m_{i-1}} ; & \xi(M_i) > 0 \quad (7.2.12b) \\ \dfrac{\xi(M_{i+1})Z_{i+1} - \xi(M_j)Z_i}{m_{i+1} - m_i} ; & \xi(M_i) < 0 \quad (7.2.12c) \end{cases}$$

with suitable convections for mass range end-points. The particular integrals that remain to be discretized, from (7.2.11), are:

$$\psi_i^A = \int_{Ai} dm\ m^L \quad (7.2.13a)$$

$$\psi_{ij}^B = \int_{Ai} dm\ m^L \int_{A_j} d\mu\ \phi(\mu,m) \quad (7.2.13b)$$

and

$$\psi_{jk}^C = \int_{A_j} d\mu \int_{A_k} d\nu(\mu + \nu)^L \theta_i(\mu + \nu)\phi(\mu,\nu). \quad (7.2.13c)$$

Unfortunately, the particular choices for all the various finite difference models are not given in the available code documentation. However, in an earlier review of several codes used for sodium-cooled reactors [4], Dunbar and Fermandjian described some of the choices made for AEROSIM, MAEROS, the aerosol behaviour model in CONTAIN, and PARDISEKO, a forerunner of NAUA. Of particular interest is that the evaluation of the integrals must be prescribed in such a way as to ensure that new particles formed by agglomeration are correctly assigned to existing, discrete "bins". For example, if the bin sizes are "one", "two", "four", "eight", and so on, a particle of size three formed by agglomeration of a "one" and a "two" must be put partly in the "two" bin and partly in the "four" bin in order to conserve mass; there are no other options. As a result, some particles will grow artificially too quickly, and others, too slowly: a phenomenon often called numerical diffusion. This is just a fact of life when using such discrete formulations.

After evaluating the integrals, the resulting system of ordinary differential equations for the unknown $Z_i$ may be written as

$$\psi_i^A \frac{dZ_i}{dt} Z_i = \frac{1}{2} \sum_{j=1}^{i} \sum_{k=1}^{i} \psi_{jk}^C Z_j Z_k - Z_i \sum_{j=1}^{N} \psi_{ij}^B Z_j + S\,\psi_i^A - R\,\psi_i^A Z_i$$

$$- \psi_i^A \nabla_{\xi z} \quad ; \quad i = 1....N. \quad (7.2.14)$$

This set of ordinary differential equations is integrated with respect to time by numerical methods; in the process the solution is advanced from time t to time t + Δt. The general approaches used in the codes are given in Table 7.1.

In AEROSIM no functional relationship is assumed for the particle size distribution within a "bin", but the "bin", or interval edges, are related by a fixed ratio ($m_{i+1}/m_i = m_i/m_{i-1}$), which can lead to economies of scale when both very small and very large particles are expected.

It is of interest to note that, in the CONTAIN algorithm, particle mass is assumed to vary logarithmically in each "bin". This affects the way in which the numerical integrals are evaluated. It also suggests that the CONTAIN algorithm could be viewed as a finite element method instead of as a finite difference method, at least theoretically.

## 7.3 FINITE ELEMENT METHODS

The Finite Element Method described for the code AEROSOLS/B1 is a discrete, or "bin", method, not unlike the finite difference methods described previously. The most significant difference is the choice of approximating function. Here, the values of the time-only dependent parameters are defined at the interval end-points. Assuming the use of the mass variable, one may write

$$Z_i = Z(m_i) \; ; \quad i = 0.....N \tag{7.3.1}$$

and the distribution is assumed to be linear over the intervals:

$$Z(m) = \frac{(m_i - m)\, Z_{i-1} + (m - m_{i-1})Z_i}{m_i - m_{i-1}} \; ; \quad m_{i-1} \leq m < m_i. \tag{7.3.2}$$

It is convenient to introduce the concept of an interpolating polynomial $g_i(m)$, defined as taking the following values:

$$g_i(m) = 1 \; ; \quad m = m_i \tag{7.3.3a}$$

$$g_i(m) = 0 \; ; \quad m = m_j, \text{ with } i \neq j \tag{7.3.3b}$$

and

$$g_i(m) = \frac{m_{i+1} - m}{m_{i+1} - m_i} \; ; \quad m_i \leq m < m_{i+1} \tag{7.3.3c}$$

$$g_i(m) = \frac{m - m_{i-1}}{m_i - m_{i-1}} \; ; \quad m_{i-1} \leq m < m_i. \tag{7.3.3d}$$

With these definitions we have

$$Z(m) = \sum_{i=0}^{N} g_i(m) Z_i. \tag{7.3.4}$$

Note that there are N + 1 variables for N intervals.

In application in AEROSOLS/B1, the mass scale m is replaced by log(m). The distribution is thus linear on a log scale. The actual algorithm used in AEROSOLS/B1 can be constructed by using a Dirac delta weighting function

$$W_j = \delta(m - m_j). \tag{7.3.5}$$

Thus

$$\int_0^\infty W_j f(m) dm = f(m_j). \tag{7.3.6}$$

Application of this to the basic equation (7.11), leaving out the condensation term, results in

$$\frac{d}{dt} Z_i = \tfrac{1}{2} \int_0^{m_i} d\mu\ \phi(\mu, m_i - \mu)\ Z_i \sum_{j=0}^{N} g_j(\mu) Z_j$$

$$- Z_i \int_0^{\infty} d\mu\ \phi(m_i, \mu) \sum_{j=0}^{N} g_j(\mu) Z_j$$

$$+ S(m_i, t) - R(m_i, t) Z_i + I_5. \qquad (7.3.7)$$

The condensation term is established using methods similar to those used in finite difference method, (7.3.12), and $I_5 = \nabla_{\xi z}$.

Integrals in (7.3.7) that need to be evaluated are

$$\int_0^{m_i} d\mu\ \phi(\mu, m_i - \mu) g_j(\mu) \text{ and } \int_0^{\infty} d\mu\ \phi(m_i, \mu) g_j(\mu).$$

Numerical diffusion occurs in finite element methods, just as it does in finite difference methods. Details of these integral evaluations has been given by Dunbar and Fermandjian [4].

## 7.4 DISCUSSION

Each step in a computer code algorithm involves only linear operations. Solution of the non-linear integro-differential equation by numerical methods involves reducing it to a repeated set (or iterated subsets) of linear operations. Each of the algorithms described--the moment method, the various finite difference methods, and the finite element method--are simply ways of linearizing the system. In the process, information is lost. In particular cases of complex agglomeration kernels and removal rates, it is not clear which algorithm would be superior.

Two essentially different aspects must be considered. First of all, one must ask whether or not the assumed form of the particle size distribution is adequate to model the actual physical phenomenon. Secondly, assuming the adequacy of this representation, does the computer algorithm do a reasonably accurate job of calculating agglomeration and removal rates and integrating these to produce the time-dependent behaviour.

With respect to the adequacy of the representation, the moment method codes with their assumed lognormal distribution obviously cannot cope with situations like the first test problem, where there are two "humps" in the distribution after the corium/concrete interaction. The discrete, or "bin", methods assume either a histogram representation on a mass (or volume or log-mass) scale in finite difference applications, or a linear representation on a log-mass scale. Resolution depends upon the number of "bins". Users of "bin" codes are under some obligation to demonstrate convergence of their own code versions to demonstrate that the resolution is "good enough" to be believable. The maximum bin size used is also of importance in determining action to be taken if particles generated by agglomeration exceed the maximum expected size.

An important consideration is the ability of the numerical model to conserve mass. The agglomeration process itself must be conservative, for there should be no noticeable mass gained or lost when two particles combine to form a larger particle. As well, the inventories of mass added to the system from sources, mass removed from the system through the various deposition mechanisms, and the suspended mass should form an arithmetically closed set; all the mass should be accounted for. Each code under review apparently ensures that mass is conserved, but the specifics as to how this is accomplished are not given.

In so far as accuracy of the algorithms is concerned, some basic tests could be performed by code users for characterization. For example, in the absence of removal, source and condensation terms, particles can only agglomerate to produce particles of larger mass. When agglomeration rates are constant, some analytical solutions exist [27]. The airborne mass must remain constant. Such tests are often done by code developers to ensure that the coding is correct; they could also be done in the field to ensure that the number of "bins" used in the discrete codes and time steps are reasonable.

Other tests, in the same vein, come to mind. With only removal mechanisms acting, the solution should approximate exponential decay, and with only condensation operating, the solution should approximate a simple wave propagation equation. Tests such as these would be very useful and helpful in characterizing algorithm performance and significant errors made in using the codes could be minimized.

## 8. A TEST CODE FOR AEROSOL MODEL ASSESSMENT

To facilitate comparing the effects of the various model assumptions, a simple finite-difference algorithm has been built that permits, by switch selection, any of the physical models described in this report to be run. Although all integrations are done numerically, an equivalent lognormal distribution at each time step is selectable. Thus, virtually all of the major possibilities are available. By selecting models appropriately, users of this test code should be able to emulate the performance of any of the codes under review, provided, of course, that the models have been faithfully transported. The multicomponent model, equation (7.8), was chosen for the model so that capabilities reported for CONTAIN and AEROSIM-M could be examined.

The numerical method chosen was the method of weighted residuals, in a finite difference formulation, and the equation modelled was Equation (7.2.14) with parameters chosen to match AEROSIM [4,8].

Semi-implicit backward Euler integration was used to advance the solution in time, with automatic time-step control as used in the thermalhydraulics code CATHENA [28]. The simple tests, as outlined in Section 7.4, were performed to ensure that numerical errors were "reasonable". Because the objective was to produce a fast-running code to compare assumptions, some error in numerics is involved. But the numerical bias introduced should be consistent, and differences in results produced by modifying assumptions should be qualitatively, if not quantitatively, correct.

The question of mass conservation was addressed as follows. At the beginning of each increment in time, the agglomeration process is calculated first and is checked to see if the mass resulting at the end of the increment from the agglomeration processes will match that at the beginning. If not, and this is usually the case because of roundoff errors, the particle concentration in the "bin" with the most mass is arbitrarily adjusted at the end of the time increment. Typically, the fractional adjustment is of the order of $10^{-14}$, reflecting the errors accumulated as a result of roundoff in double precision. After the time increment, the inventories of suspended and deposited mass are compared to the cumulative source mass, to check on the overall conservation of mass for the time step. Any discrepancy found is arbitrarily assigned to the cumulative settled mass. The operation is subtle but necessary to maintain any confidence in the values calculated for suspended mass at times late in the event, when the suspended mass is but a tiny fraction of the total mass injected (or removed by deposition) to that time.

Although this test code is a "bin", or discrete, finite difference model, it was found possible and relatively easy to force the distribution to be lognormal, upon request, as follows: at the end of a time-step, the parameters characterizing a lognormal distribution (Equation 7.10) are calculated for the actual discrete distribution produced, and are used to modify the actual discrete distribution to one lognormal in shape. When working in this mode, the test-code can emulate the performance of HAA-4, HAARM-S, RETAIN-S and RETAIN-2C, albeit with numerical rather than analytical integration of the agglomeration and removal models.

All variations for agglomeration, removal and condensation were modelled following the documentation available for the various codes. Thus, by specifying the code-to-be-emulated as input data, the appropriate models are automatically selected. Over-ride selection of collision efficiency model, wet or dry aerosols, and Stephan flow model permits emulation of many variations. In cases where the available documentation is unclear about model selection, CONTAIN parameters were used.

The test code is strictly an aerosol behaviour model and does not incorporate any aerosol-thermalhydraulics interaction. Thus it cannot really emulate the performance of CONTAIN, which does include a full thermalhydraulics package. The test code simply distributes the condensing steam over the aerosol particles, and the method used is that reported for AEROSIM [8].

## 9. RESULTS FOR THE SURRY-AB CASE

The test code was run in emulation of the actual codes used for the cases run for the SURRY-AB hypothetical accident case in a recent code comparison exercise sponsored by the OECD group of experts on the source term. This sequence is a loss-of-coolant accident resulting from a large break in the primary hot-leg, combined with failure of AC power to the engineered safety systems. Molten corium/concrete interactions following reactor vessel melt-through are included, but ex-vessel steam explosions are not.

Selected comparisons between the actual code results and their test-code emulations on the SURRY-AB case are presented in Tables 1.2 through 1.3. The data shown are cumulative settled mass (Table 9.1), plated mass (Table 9.2), and leaked mass (Table 9.3) at the end of the run at 172 800 seconds.

The emulation of the lognormal codes, without steam condensation or Stephan flow, using the finite difference numerics and a forced lognormal distribution, is quite close. The shape of the distribution is the dominant effect, and replacing the analytical integrations based on power series approximations of the agglomeration kernels and removal rates with simple numerical integration appears to have little effect.

The test code appears to over estimate settled mass and under estimate plated mass for the "wet" cases, where steam is allowed to condense on the aerosol particles. Leaked mass is only seriously underestimated for the CONTAIN emulation, and the reason is not yet clear. For the cases where comparison is possible, HAA-4 and SWNAUA, it appears as though changing the collision efficiency from Fuchs to Pruppacher-Klett has a greater effect on cumulative leaked aerosol mass than allowing steam condensation on the particles. The effect is noticeable in both the actual and the emulated results, indicating that the effect is indeed physical and not numerical.

It is of some interest to note from the ratios of the emulated results to the actual results that, in the mean from all the runs, the emulated leaked mass is within 10% of the actual code-calculated leaked mass. This suggests that the numerical modelling, in general, is quite sound, and that discrepancies should be resolved by improving the physical models. Although it is true that slight variations in selecting weighting functions and numerical integration procedures will have some effect on the detail of the code predictions, these are probably of secondary importance compared to the variations in the physical models. It remains true, however, that code users must take care in using the code packages to ensure that they don't artificially create errors by misuse, for example, by not having a sufficient number of bins in a discrete representation.

## 10. CLOSURE

The documentation made available for the codes used in the 1985 GREST Containment Code Comparison Exercise has been used to compile the physical models used in the codes for the basic aerosol physical phenomena. The numerical methods have been described as variations of the general Method of Weighted Residuals, and appear all to be effective when used with care and skill. Moment methods are presently restricted by the log-normal particle size distribution assumption, and discrete methods can lose accuracy if an insufficient number of bins is used. At present, there does no seem to be a single numerical method, that is, a numerical agglomeration for approximating a solution to the mathematical problem, which is demonstrably superior to all the others.

There is a considerable amount of variation in the physical models used in the codes. Some consensus is required, or some definitive experiments are required, to remove the uncertainties that presently exist. In a parameter as fundamental as particle collision efficiency, for example, there is no agreement as to the best model. From the results received, it would appear that variation in the collision efficiency can lead to variations in leaked mass predictions as large as those observed from the inclusion of steam condensation phenomena.

Condensation phenomena are not well documented at this time. There can be little doubt that in the humid containment building environment following a loss-of-coolant accident in a water-cooled reactor, condensation and steam thermalhydraulics behaviour will play a very important role. In fact, it is possible that during some stages of an accident scenario, water aerosol physics will be completely dominant, and the fission product materials will just "be along for the ride". Those codes which have true multi-component capabilities will be more successful in accounting for the airborne components than will those codes which lump the components together, particularly during accidents where different components may be released at different times.

It would appear that the ideal code for predicting nuclear aerosol transport behaviour in containment following a severe core damage accident would have state-of-the-art, acknowledged best models for basic physical phenomena, would be able to keep track separately of each material species, would have a state-of-the-art atmospheric model for integrated analysis of thermalhydraulic and aerosol interaction, and would have a fast-running, highly accurate numerical modelling package. Such a code may never exist, although at present, the CONTAIN code perhaps comes closest.

## 11. SYMBOLS

| | |
|---|---|
| $A$ | Surface area |
| $A_i$ | Range of discrete bin i |
| $[A]$ | Matrix of coefficients |
| $\underline{b}$ | Vector of source terms |
| $\overline{B}$ | Particle mobility |
| $C(m,t)$ | Number of particles of mass m at time t |
| Cu | Cunningham slip correlation |
| $C_\varepsilon$ | Particle collision efficiency coefficient |
| $C_T$ | Particle total number density - log-normal distribution |
| $C_L(m,t)$ | Number of particles of mass m at time t (log-normal distribution) |
| $C_M, C_t$ | Brock factor coefficients |
| $C_n$ | Condensation number |
| $C_\infty, C_w$ | Steam concentrations, in containment, at wall |
| $D$ | Particle diffusivity |
| $D_v$ | Steam diffusion coefficient in containment gas mixture |
| $E_i$ | Coefficients for Cunningham slip correlation |
| $F(C(m,t),m,t)$ | Function describing particle size time rate of change |
| $F_B$ | Brock factor |
| $F_F$ | Non-isotropic correction for Brownian agglomeration |
| $g$ | Gravitational acceleration |
| $g_i(m)$ | Interpolating polynomial function i |
| $h_i$ | length of discrete mass range i |
| $I_j$ | Particular integral j |
| $J_v$ | Wall steam condensation flux |
| $K$ | gas thermal conductivity |
| $k$ | Boltzmann constant |
| Kn | Knudsen number |
| $K_B$ | Brock factor coefficient |
| $L$ | Latent heat of evaporation of water |
| Mw | Molecular weight of gas |
| $m_s$ | Mass of steam in containment |
| $m_g$ | Geometric mean mass - log-normal distribution |
| $M_i$ | Mean mass in discrete range i |
| $m_i$ | Particle mass at lower end of discrete range i |
| $m,n$ | Particle mass |
| $m_v, m_j$ | Molar masses in containment mixture: steam, component j |
| $n_k$ | Relative number of moles of constituent gas k |
| $P$ | Pressure |
| $P_i$ | Partial pressure of constituent i |
| $q_k(m,t)$ | Mass of component k on particles of mass m at time t |
| $R(m,t)$ | Removal rate of particles of mass m at time t |
| $R_G(m,t)$ | Gravitational removal rate |
| $R_B(m,t)$ | Brownian diffusion removal rate |
| $R_T(m,t)$ | Thermophoretic removal rate |
| $R_D(m,t)$ | Diffusiophoretic removal rate |
| $R_L(m,t)$ | Particle leakage rate |
| $r$ | Particle radius |
| $R$ | Universal gas constant |
| Re | Reynolds number |
| $S$ | Saturation ratio |
| $S_o$ | Saturation ratio at droplet surface |
| $S(m,t)$ | Source rate of particles of mass m at time t |
| $S_T$ | Particle agglomeration sticking parameter |

| | |
|---|---|
| $T$ | Temperature |
| $t$ | Time |
| $T_w$ | Temperature of wall surface |
| $V$ | Containment volume |
| $V_T$ | Particle terminal settling velocity |
| $W_i$ | Method of residuals weighting function i |
| $Y_T$ | Total condensation rate of steam on particles |
| $Z_i$ | Method of residuals time-only dependent variable i |
| $\alpha_g$ | Logarithm of geometric standard deviation - log-normal distrib. |
| $\chi$ | Dynamic shape factor |
| $\delta_B$ | Diffusive boundary layer thickness |
| $\delta_T$ | Thermophoretic boundary layer thickness |
| $\delta_{ik}$ | Kronecker delta function |
| $\Delta m$ | Increment in mass |
| $\Delta t$ | Increment in time |
| $\varepsilon(m,n)$ | Collision efficiency between particles of masses m and n |
| $\varepsilon_T$ | Gas turbulent energy dissipation rate |
| $\nabla T$ | Temperature gradient at wall surface |
| $\nabla_{\xi z}$ | Numerical condensation gradient function |
| $\phi(m,n)$ | Agglomeration kernel: collision rate; particles of masses m and n |
| $\phi_G(m,n)$ | Gravitational agglomeration rate |
| $\phi_B(m,n)$ | Diffusional, of Brownian, agglomeration rate |
| $\phi_{TS}(m,n)$ | Turbulent shear agglomeration rate |
| $\phi_{TI}(m,n)$ | Turbulent inertial agglomeration rate |
| $\gamma$ | Agglomeration shape factor |
| $\gamma_v, \gamma_j$ | Molar fractions in containment gas mixture: steam, component j |
| $\eta$ | Gas viscosity |
| $\theta(m,n)$ | Discrete mass range selector function |
| $\lambda$ | Gas mean free path |
| $\Lambda(m,t)$ | Residual in Method of Weighted Residuals |
| $\mu, \nu$ | Particle mass |
| $\psi$ | Particular numerical integration term |
| $\rho_g$ | Gas density |
| $\rho_s$ | Density of steam in bulk carrier gas |
| $\rho_w$ | Density of water droplet |
| $\rho_{s\varepsilon}$ | Steam density, equilibrium with plane surface of pure water |
| $\rho'_{s\varepsilon}(T)$ | Temperature derivative of equilibrium vapour density |
| $\xi(m,t)$ | Condensation rate of steam on particles of mass m at time t |

## 12. REFERENCES


1. J. Royen, "Summary Record of the Meeting of an Ad Hoc Group on the CSNI LWR Containment Aerosol Code Comparison Exercise," 1984 October 18," OECD-NEA Draft Report No. SEN/SIN(84)52. 1984 October.

2. A.A. Morewitz, F.J. Rahn, R. Sher, "Results of the GREST Code Comparison Exercise," CSNI Report 116 OECD/NEA, 1986.

3. J. Fermandjian, "Comparison of Computer Codes Related to the Sodium Oxide Aerosol Behaviour in a Containment Building," Symposium on Reactor Safety, Karlsruhe, FRG, 1984 September 4-6.

4. I.H. Dunbar and J. Fermandjian, "Comparison of Sodium Aerosol Codes," CEC Report EUR 9172, 1984 July.

5. F.J. Rahn, "GREST Code Comparison Exercise," Electric Power Research Institute Letter Report, 1984 November.

6. H. Haggblom, "GREST Code Comparison Exercise," Studsvik Energiteknik AB, Nykoping Technical Note NR-85/4, 1985 January 8.

7. K.K. Murata et al., "CONTAIN. Recent Highlights in Code Testing and Validation," Proc. International Meeting on LWR Severe Accident Evaluation, Cambridge, MA. 1983 Aug. 28 - Sept. 1.

8. I.H. Dunbar, "Steam Condensation Modelling in LWR Aerosol Codes," CEC Report on Contract ECI-1232B 7210-84-UK, 1984.

9. J.Fermandjian, H. Bunz, I.H.Dunbar, G Lhiaubet, and F. Beonio-Brocchieri, "Comparison of European Computer Codes Relative to the Aerosol Behaviour in PWR Containment Buildings During Severe Core Damage Accidents", AEROSOLS: Formation and Reactivity, p.p. 1087-1091,, Second Int. Aerosol Conf. Berlin (West), 1986 September 22-26, Pergamon Press, 1986.

10. B.C. Walker, R. Williams, and C.R. Kirby, "Discretization and Integration of the Equation Governing Aerosol Behaviour," UKAEA Report: SRD R98, 1978 July.

11. S.A. Ramsdale, "AEROSIM Input and Output," UKAEA Report: NST/84/245, 1984.

12. A. L'Homme et al., "Presentation du code AEROSOLS/B1," CEA Report, 1983 November.

13. H. Bunz et al., "NAUA Mod 4, A Code for Calculating Aerosol Behaviour in LWR Core Melt Accidents, Code Description and Users Manual," Kernforschungszentrum Karlsruhe, Report No. KFK-353, 1983 August.

14. H. Bunz et al., "NAUA Mod 5, A Code for Calculating Aerosol Behaviour in LWR Core Melt Accidents," Kernforschungszentrum Karlsruhe, Unpublished Computer Listing, 1985 January.

15. J.M. Otter and E.U. Vaughan, "HAA-4 Code Description and Users Manual," Rockwell International Report: AI-DOE-13528 (1985 September); also U.S. Dept. of Energy, Fast Reactor Safety Program, Argonne National Lab. Report Nos. ANL/TMC 82-1, ANL/TMC 82-2, 1982.

16. R. Burns et al., "RETAIN Users Manual," EDS Nuclear Inc. Draft Report, 1982 December.

17. REMOVAL (unpublished JAERI aerosol transport code) (1985).

18. A. Gieseke, K.W. Lee and L.D. Reed, "HAARM-3 Users Manual, "Battle Columbus Report: BMI-NUREG-1991, 1978 January. (HAARM-S is a version of HAARM-3 developed by Studavik Energiteknik AB and is obtainable from the NEA Data Bank).

19. I.H. Dunbar, "The Role of Diffusiophoresis in LWR Accidents," Unpublished UKAEA Report, SRD, Culcheth, 1986.

20. J. Gauvain and G. Carvallo, "Description of the AEROSOLS/B1 Code", Technical Rport DEMT 84-269 SYST/LECS 84-035, CEA (France), 1984 October.

21. P. Nakayama and M. Ringham, "Review of the RETAIN Code", Prepared for the IDCOR Program by JAYCOR, 1983 December.

22. M.L. Tobias and R.E. Adams, "Status of Validation of the NAUA Computer Code Used for the Accident Source Term Reassessment Program", in "Review of Validation of the Computer Codes Used in the Severe Accident Source Term Reassessment Study (BMI-2104)", ORNL/TM-8842, 1985 April.

23. K.D. Bergeron et al, "User's Manual for CONTAIN 1.0", Sandia National Laboratories, Albuquerque NM, NUREG/CR-4085, SAND84-1204, 1985 May.

24. F. Gelbard, "MAEROS User Manual", Sandia National Laboratories, Albuquerque NM, NUREG/CR-1391, SAND80-0822, 1982 December.

25. A. Drozd, Stone and Webster Engineering Corporation, Boston MA, Letter to T.S. Kress, Dated 1985 January 10.

26. S. Simons, M.M.R. Williams, J.S. Cassell, "A Kernel for Combined Brownian and Gravitational Coagulation" J. Aerosol Sci., Vol. 17, No. 5, pp. 789-793, 1986.

27. B.A. Finlayson, "The Method of Weighted Residuals and Variational Principles with Application in Fluid Mechanics, Heat and Mass Transfer", Academic Press, New York, 1972.

28. I.H.Dunbar and S.A. Ramsdale, "The Development and Testing of the Aerosol Behaviour Code, AEROSIM", AEROSOLS: Formation and Reactivity, p.p. 1083-1086,, Second Int. Aerosol Conf. Berlin (West), 1986 September 22-26, Pergamon Press, 1986.

29. B.H. McDonald and D.J. Richards, "Application of Dynamic Grid Control in the RAMA code", 9th Simulation Symposium on Reactor Dynamics and Plant Control, Mississauga,Ontario, 1982 April 19-20.

30. S.R. Loyalka and G.A. Pertmer, "Gravitational Collision Efficiency of Post Hypothetical Core Disruptive Accident Liquid-Metal Fast Breeder Reactor Aerosols: Spherical Particles," Nuclear Technology, pp. 70-80, Vol. 47, 1980 January.

31. B.H. McDonald, "Assessing Physical Models Used in Nuclear Aerosol Transport Models", proceedings of the OECD/CEC Workshop on Water-Cooled Reactor Aerosol Code Evaluation and Uncertainty Assessment, Brussels, Belgium, 1987 Sept. 9-11, (Also available as AECL-9098).

32. B.H. McDonald, "Assessing Numerical Methods Used in Nuclear Aerosol Transport Models", proceedings of the OECD/CEC Workshop on Water-Cooled Reactor Aerosol Code Evaluation and Uncertainty Assessment, Brussels, Belgium, 1987 Sept. 9-11, (Also available as AECL-9099).

TABLE 1.1 OECD/GREST COMPARISON CODES AND PARTICIPANTS

| MOM - MOMENT METHOD CODES: NAMES; SUBMITTING LABORATORIES | | |
|---|---|---|
| HAA-4 | Rockwell | Rocketdyne Division, Rockwell International, Canoga Park, CA, U.S.A. |
| HAARM-S | UPM | Catedra de Tecnologia Nuclear, Universidad Politecnica, Madrid, Spain |
| RETAIN-2C | VTT | Valtion Teknillinen Tutkimuskeskus, Helsinki, Finland |
| RETAIN-S | Studsvik | Studsvik Energiteknik AB, Nykoping, Sweden |
| **FDM - FINITE DIFFERENCE CODES: NAMES; SUBMITTING LABORATORIES** | | |
| AEROSIM-M | UKAEA | United Kingdom Atomic Energy Authority, Safety and Reliability Directorate, Culcheth U.K. |
| CONTAIN | SNL | Sandia National Laboratory, Division 6449, Albuquerque, NM, U.S.A. |
| NAUA-4 | EPRI | Electric Power Research Institute, Palo Alto, CA, U.S.A. |
| NAUA-4 | SWEC | Stone and Webster Engineering Corporation, Boston, MA, U.S.A. |
| HAUA-5 | KfK | Kernforschungszentrum Karlsruhe GmbH, Karlsruhe, F.R.G. |
| REMOVAL | JAERI | Japan Atomic Energy Research Institute, Tokai-Mura, Japan. |
| SWNAUA | SWEC | Stone and Webster Engineering Corporation, Boston, MA, U.S.A. |
| **FEM - FINITE ELEMENT CODE: NAME; SUBMITTING LABORATORY** | | |
| AEROSOLS/B1 | CEA | Commissariat a l'Energie Atomique Saclay, France. |

TABLE 1.2

HYPOTHETICAL TEST CASE ACCIDENTS

1. The AB hot-leg sequence in the U.S. Surry 1 PWR. This sequence was a loss-of-coolant accident (LOCA) resulting from a large break in the primary system hot-leg, combined with failure of AC power to the engineered safety features. Molten corium/concrete interactions following reactor vessel melt-through were included, but ex-vessel steam explosions were excluded.

2. The $_{S}$2CD cold-leg sequence in the U.S. Surry 1 PWR. This accident was a LOCA due to a small break in the primary system cold-leg, combined with failure of the emergency core cooling injection system and containment spray system, but with the containment safety systems in operation. Extensive corium/concrete reactions were precluded due to assumed formation of a coolable debris bed. Ex-vessel steam explosions were not considered in this scenario.

3. A large hot-leg break LOCA for the German Biblis-B PWR. This scenario assumes that: (1) the reactor cavity was initially dry and corium/concrete interaction follows immediately upon pressure vessel melt-through; (2) there are no containment sprays; and that (3) the Emergency Core Cooling System operates, but that conversion to recirculation core cooling fails, leading to core melting.

Note: Two different sets of initial conditions were used for Problem #3 by various participants resulting in the submission of results to two different problems:

Problem 3-A: source release time at 4080 seconds after accident initiation used as starting time, with total source mass of approximately 4200 kg; and

Problem 3-B: accident initiation used as starting time, with a total source mass of 5800 kg.

## TABLE 1.3

## TEST CASE INPUT PARAMETERS

All shape factors = 1 (i.e. Mobility, Coagulation, Condensation)

Density Correction Factor = 1

Collision efficiency model: Fuchs for problems #1 & #3;
Pruppacher-Klett for problem #2

| Size Distribution of Aerosol Source: | Mass Median Radius | Sigma |
|---|---|---|
| Problem #1 (Surry AB) | 0.5 micrometer | 2.0 |
| Problem #2 (Surry $S_2CD$) | 1.5 micrometer | 2.0 |
| Problem #3 (Biblis LOCA) | 0.16 micrometer | 1.5 |

Ratio of gas to particle thermal conductivity = 0.0042

| | |
|---|---|
| Diffusiophoresis boundary layer | 2.5 mm |
| Thermophoresis boundary layer | 2.0 mm |
| Brownian diffusion boundary layer | 0.1 mm |

Van't Hoff factor = 1.913

Turbulent energy dissipation rate = 0.1 $m^2/s^3$

Geometry:

SURRY (Problems 1 & 2):

Containment:

Volume = 50 970 cubic meters
Wall area = 21 900 sq. meters
Floor area = 1 277 sq. meters
Leak rate = 1.0 volume %/day

Spray Parameters

Flow rate = 0.4 cubic meters/s
Mass Median Drop Size = 1000 μM
Sigma = 1.5
Entering liquid temperature = 280 K

BIBLIS (Problem 3):

Containment:

Volume = 72 000 cubic meters
Wall area = 50 000 cubic meters
Floor area = 4500 cubic meters
Leak rate = 2.5 volume %/day

Ringraum (annulus)

Volume = 30 000 cubic meters
Wall area = 34 000 sq. meters
Floor area = 4100 sq. meters
Leak Rate = 400 vol %/day
(no credit for filter)

TABLE 1.4 CODE PARTICIPANT RESULTS SUBMITTED

| MOM CODES | LAB | SURRY1-AB DRY | SURRY1-AB WET | SURRY-$S_2$CD DRY | SURRY-$S_2$CD WET | BIBLIS-A DRY | BIBLIS-A WET | BIBLIS-B DRY | BIBLIS-B WET |
|---|---|---|---|---|---|---|---|---|---|
| HAA-4 | Rockwell | F[1] | F,P | P | P | F | F | F | F |
| HAARM-S | UPM | F | – | F | – | – | – | P[1] | – |
| RETAIN-2C | VTT | F,P[2] | F | P | P | F | F | – | – |
| RETAIN-S | Studsvik | P | – | P | – | P | – | – | – |

| FDM CODES | LAB | SURRY1-AB DRY | SURRY1-AB WET | SURRY-$S_2$CD DRY | SURRY-$S_2$CD WET | BIBLIS-A DRY | BIBLIS-A WET | BIBLIS-B DRY | BIBLIS-B WET |
|---|---|---|---|---|---|---|---|---|---|
| AEROSIM-M | UKAEA | F | F | P | P | F | F | – | – |
| CONTAIN | SNL | F | – | – | – | – | – | – | – |
| NAUA-4 | EPRI | F[3] | F[3] | P | P | F[3] | F | – | – |
| NAUA-4 | SWEC | F,P | F,P | – | – | – | – | P | P |
| HAUA-5 | KfK | P | P | – | P | – | P | – | – |
| REMOVAL | JAERI | F[4] | – | P[4] | – | – | – | – | – |
| SWNAUA | SWEC | F,P | P | P[5] | P[5] | – | – | P | P |

| FEM CODE | LAB | SURRY1-AB DRY | SURRY1-AB WET | SURRY-$S_2$CD DRY | SURRY-$S_2$CD WET | BIBLIS-A DRY | BIBLIS-A WET | BIBLIS-B DRY | BIBLIS-B WET |
|---|---|---|---|---|---|---|---|---|---|
| AEROSOLS/B1 | CEA | F[5] | – | – | – | – | – | – | – |

DRY = Results submitted without steam condensation on particles
WET = Results submitted with steam condensation on particles
F = Results submitted with Fuchs collision efficiency
P = Results submitted with Pruppacher-Klett collision efficiency
– = No results submitted

Superscripts: 1 = without Stephan flow
2 = with water mass
3 = with and without Stephan flow
4 = dry/wet modelling unclear
5 = collision efficiency unclear

TABLE 3.1

CODE CALCULATED GAS DENSITIES

In cases where the codes calculate carrier gas densities (air or a mixture of air and steam) directly the formulae are as follows:

| CODE NAME | FORMULA FOR CARRIER GAS DENSITY $\rho_g$ |
|---|---|
| CONTAIN | 0.000121 P Mw/T<br>P - Pressure Mw - molecular weight of gas<br>T - temperature |
| REMOVAL | $\frac{\rho_1 P_1 + \rho_2 P_2}{P_1 + P_2}$ ; $\rho_1 = \frac{P_1 Mw_1}{RT}$<br>$P_i$ is the partial pressure<br>R is the universal gas constant<br>$n_k$ = relative number of moles of constituent gas k. |
| HAA-4 | $\frac{Mw \bar{P}}{RT}$ ; $\bar{Mw} = \frac{\Sigma Mw_k n_k}{\Sigma n_k}$ |
| RETAIN-2C | $\frac{1.0}{461510.0 (T/P) - 8.8}$ |

## TABLE 3.2

## CODE CALCULATED MEAN FREE PATH LENGTHS

In cases where the codes calculate gas mean free paths directly the formulae are as follows:

| CODE NAME | FORMULA FOR MEAN FREE PATH LENGTH $\lambda$ |
|---|---|
| CONTAIN | $(\eta/\rho_g)\ (0.000189\ Mw/T)^{1/2}$<br>$\eta$ - gas viscosity  T - temperature<br>$\rho_g$ - gas density<br>Mw - molecular weight of gas. |
| AEROSOLS/B1 | $\eta\left[\frac{\pi}{2\ P\rho_g}\right]^{1/2}$<br>P - pressure |
| HAARM | .000000118 T/373 |
| NAUA | $\eta\left[\frac{3}{P\ \rho_g}\right]^{1/2}$ |
| RETAIN-2C | $0.0124\ (\mu/\rho_g)\ \left[\frac{18.0}{83140000.0\ T}\right]^{1/2}$ |
| REMOVAL | $\frac{\mu_M}{\rho_M}\left[\frac{\pi\ Mw_M}{2\ RT}\right]$  $Mw_M = \frac{Mw_1\ P_1 + Mw_2\ P_2}{P_1 + P_2}$ |
| HAA-4 | $\frac{k\ T}{4\ 2^{1/2}\ \pi\ P\ r_m^2}$  $r_m^2 = \frac{\Sigma\ n_k\ r_{m,k}^2}{\Sigma\ n_k}$ |

where $r_{m,k}$ is the molecular radius of component k. Values of $r_{m,k}$ are built into HAA-4.

TABLE 3.3

CODE CALCULATED GAS VISCOSITY

In cases where the codes calculate gas viscosity directly the formulae are as follows, with temperature T in degrees K:

| CODE | FORMULA FOR GAS VISCOSITY $\eta$ |
|---|---|
| CONTAIN | $\frac{.0066164\ (0.003661\ T)^{3/2}}{T + 114}$ |
| HAARM | $\frac{.0089063}{(T + 110)} \left[\frac{T}{333.32}\right]^{3/2}$ |
| NAUA | $.00000556\ (1 + \frac{T}{128})$ |
| AEROSOLS/B1 | $\frac{.0000137\ (T/196)^{1/2}}{1.14(T/71.4)^{-0.15} + 0.52\ e^{-.01007843T}}$ |
| RETAIN-2C | $\frac{0.000001851\ T^{1/2}}{1 + 680.1/T}$ |
| REMOVAL | $\sum_{i=1}^{N} \frac{\eta_i X_i}{X_i + \sum_{j \neq i} X_j \Phi_{ij}}$ $\Phi_{i,j} = \frac{1 + (\eta_i/\eta_j)^{1/2}(Mw_i/Mw_j)}{4(2^{-1/2})(1 + Mw_i/Mw_j)^{1/2}}$ |

where $\eta_i$, $Mw_i$, $X_i$ are the viscosity, molecular weight, and mole fraction of component i

| | |
|---|---|
| HAA-4 | $\sum_{i=1}^{N} \frac{\eta_i}{1 + \sum_{j \neq i} x_j \Phi_{ij}}$ $\Phi_{i,j} = \frac{[1 + (\eta_i/\eta_j)^{1/2}(Mw_i/Mw_j)^{1/4}]^2}{2(2^{1/2})(1 + Mw_i/Mw_j)^{1/2}}$ |

where $\eta_i$, $Mw_i$ are the viscosity and molecular weight of component i

## TABLE 3.4

## CODE SPECIFIC CUNNINGHAM SLIP CORRELATIONS

To correct for slippage in the particle terminal settling velocity, a correction factor Cu(m) is usually prescribed:

$$Cu(m) = 1.0 + Kn(m) \{E_1 + E_2 \exp(-E_3/Kn(m)\}$$

The three coefficients vary slightly, depending upon the code:

| CODE | $E_1$ | $E_2$ | $E_3$ |
|---|---|---|---|
| CONTAIN | 1.37 | 0.40 | 1.1 |
| REMOVAL | 1.26 | 0.46 | 1.1 |
| OTHERS | 1.246 | 0.42 | 0.87 |

TABLE 4.1

COMBINING AGGLOMERATION RATES

| CODE NAME | FORMULA FOR TOTAL AGGLOMERATION RATE $\phi$ |
|---|---|
| AEROSOLS/B1 | $\phi_B + \phi_G + \phi_{TS} + \phi_{TI}$ |
| REMOVAL | $\phi_B + [\phi_G^2 + \phi_{TS}^2 + \phi_{TI}^2]^{1/2}$ |
| HAA-4 | $\phi_B + \phi_{TS} + [\phi_G^2 + \phi_{TI}^2]^{1/2}$ |
| OTHERS | $\phi_B + \phi_G + [\phi_{TS}^2 + \phi 2_{TS}^2]^{1/2}$ |

$\phi_G$ - Gravitational agglomeration rate
$\phi_B$ - Brownian agglomeration rate
$\phi_{TS}$ - Turbulent shear agglomeration rate
$\phi_{TI}$ - Turbulent Inertial agglomeration rate.

TABLE 5.1

CODE SPECIFIC BROCK FACTOR CONSTANTS

| CODE | $K_B$ | $C_M$ | $C_t$ |
|---|---|---|---|
| CONTAIN | 1.00 | 1.37 | INPUT DATA |
| AEROSIM | 1.00 | 1.00 | 2.48 |
| AEROSOLS | 1.27 | 1.37 | 2.00 |
| HAARM | 1.00 | 1.37 | 2.50 |
| HAA4 | 1.00 | 1.37 | 2.00 |
| REMOVAL | 1.00 | 1.25 | 2.10 |
| RETAIN | 1.00 | 1.00 | 2.50 |
| NAUA | 1.00 | 1.00 | 2.48 |

TABLE 7.1

TIME INTEGRATION METHODS

| CODE | BASIC INTEGRATION METHOD |
|---|---|
| CONTAIN | RUNGE-KUTTA with timestep control |
| AEROSIM | Uses the FACSIMILE routine; backward PECE algorithm |
| AEROSOLS | Uses the STEP routine; modified ADAMS-PECE |
| HAA4 | Choices: ADAMS-MOULTON with fixed or variable time incrementing; RUNGE-KUTTA with fixed time stepping |
| REMOVAL | RUNGE-KUTTA |
| RETAIN-S | Uses the MOLCOL routine; implicit with interpolation, extrapolation and smoothing; for stiff equations |
| HAARM-S | Same as RETAIN-S |
| NAUA | Standard EULER-CAUCHY with automatic time-stepping |
| RETAIN-2C | Special, one-equation-at-a-time algorithm based on analytical forms with automatic time-stepping. |

TABLE 9.1

EMULATION OF NAUA-4, DRY CASE,
WITH FUCHS COLLISION EFFICIENCY
BUT WITHOUT STEPHAN FLOW

| EVENT TIME (s) | LEAKED MASS (g) | SUSPENDED MASS ($g/m^3$) | PLATED MASS (kg) | SETTLED MASS (kg) |
|---|---|---|---|---|
| 1 | 0 | 0.07 | 0.00 | 0 |
| 10 | 0 | 0.69 | 0.00 | 0 |
| 100 | 2 | 6.87 | 0.00 | 0 |
| 1000 | 156 | 40.46 | 0.13 | 19 |
| 2100 | 430 | 44.54 | 0.25 | 145 |
| 5100 | 806 | 9.78 | 0.37 | 2242 |
| 10000 | 1290 | 18.35 | 0.62 | 2552 |
| 20000 | 1931 | 8.54 | 0.91 | 3580 |
| 30000 | 2201 | 2.64 | 1.03 | 3964 |
| 43200 | 2346 | 1.39 | 1.10 | 4027 |
| 86400 | 2500 | 0.23 | 1.16 | 4086 |
| 172800 | 2545 | 0.04 | 1.18 | 4096 |

TABLE 9.2

EMULATION OF SWNAUA, DRY CASE,
WITH FUCHS COLLISION EFFICIENCY
AND WITH STEPHAN FLOW

| EVENT TIME (s) | LEAKED MASS (g) | SUSPENDED MASS ($g/m^3$) | PLATED MASS (kg) | SETTLED MASS (kg) |
|---|---|---|---|---|
| 1 | 0 | 0.07 | 0.00 | 0 |
| 10 | 0 | 0.68 | 0.12 | 0 |
| 100 | 2 | 6.64 | 11.70 | 0 |
| 1000 | 130 | 31.63 | 455.25 | 14 |
| 2100 | 320 | 28.22 | 912.21 | 65 |
| 5100 | 677 | 12.94 | 1384.79 | 697 |
| 10000 | 1066 | 15.52 | 1524.03 | 1174 |
| 20000 | 1684 | 6.54 | 1619.58 | 2064 |
| 30000 | 1971 | 3.30 | 1652.94 | 2279 |
| 43200 | 2105 | 0.97 | 1657.32 | 2393 |
| 86400 | 2217 | 0.22 | 1659.25 | 2429 |
| 172800 | 2270 | 0.05 | 1659.36 | 2438 |

## TABLE 9.3

EMULATION OF SWNAUA, DRY CASE,
WITH PRUPPACHER-KLETT COLLISION EFFICIENCY
AND WITH STEPHAN FLOW

| EVENT TIME (s) | LEAKED MASS (g) | SUSPENDED MASS ($g/m^3$) | PLATED MASS (kg) | SETTLED MASS (kg) |
|---|---|---|---|---|
| 1 | 0 | 0.07 | 0.00 | 0 |
| 10 | 0 | 0.68 | 0.12 | 0 |
| 100 | 2 | 6.64 | 11.70 | 0 |
| 1000 | 130 | 31.64 | 455.99 | 13 |
| 2100 | 321 | 28.54 | 914.65 | 46 |
| 5100 | 737 | 21.73 | 1436.86 | 196 |
| 10000 | 1400 | 21.95 | 1687.82 | 682 |
| 20000 | 2279 | 9.40 | 1823.79 | 1714 |
| 30000 | 2673 | 5.04 | 1866.48 | 1976 |
| 43200 | 2927 | 2.16 | 1874.68 | 2114 |
| 86400 | 3140 | 0.31 | 1878.30 | 2204 |
| 172800 | 3205 | 0.05 | 1878.45 | 2217 |

TABLE 9.4

EMULATION OF SWNAUA, WET CASE,
WITH PRUPPACHER-KLETT COLLISION EFFICIENCY
AND WITH STEPHAN FLOW

| EVENT TIME (s) | LEAKED MASS (g) | SUSPENDED MASS ($g/m^3$) | PLATED MASS (kg) | SETTLED MASS (kg) |
|---|---|---|---|---|
| 1 | 0 | 0.07 | 0.00 | 0 |
| 10 | 0 | 0.69 | 0.12 | 0 |
| 100 | 2 | 6.64 | 11.46 | 0 |
| 1000 | 135 | 35.41 | 251.75 | 25 |
| 2100 | 378 | 39.93 | 251.85 | 129 |
| 5100 | 932 | 17.81 | 252.01 | 1583 |
| 10000 | 1431 | 19.80 | 252.28 | 2227 |
| 20000 | 2476 | 10.64 | 252.66 | 3221 |
| 30000 | 2858 | 4.57 | 252.82 | 3613 |
| 43200 | 3109 | 2.33 | 252.92 | 3727 |
| 86400 | 3368 | 0.40 | 253.02 | 3825 |
| 172800 | 3445 | 0.06 | 253.05 | 3842 |

TABLE 9.5

SURRY-AB FINAL SETTLED MASS (kg)

| CODE | ACTUAL | EMULATED | RATIO |
|---|---|---|---|
| HAA-4 (W,P-K) | 3150 | 3370 | 1.070 |
| HAA-4 (W,F) | 3240 | 3783 | 1.168 |
| HAA-4 (D,F) | 3110 | 2228 | 0.716 |
| HAARM-S (D,F)* | 4091 | 4095 | 1.001 |
| RETAIN-S (D,P-K) | 2740 | 2005 | 0.732 |
| NAUA-4 (D,F)* | 3990 | 4096 | 1.027 |
| SWNAUA (D,P-K) | 3028 | 2217 | 0.732 |
| SWNAUA (W,P-K) | 3040 | 3842 | 1.264 |
| SWNAUA (D,F) | 3300 | 2437 | 0.738 |
| REMOVAL (D,F)+ | 1970 | 2167 | 1.100 |
| AEROSOLS-B1 (D,F) | 3170 | 2444 | 0.771 |
| AEROSIM-M (D,F) | 3372 | 2418 | 0.717 |
| AEROSIM-M (W,F) | 3473 | 3820 | 1.100 |
| CONTAIN (W,F) | 3335 | 3901 | 1.170 |

ACTUAL = Results submitted by the actual code user
EMULATED = Results obtained with test code emulating actual code models
RATIO = Emulated Result divided by actual result; mean = 0.950 ± 0.204

W = Wet case: condensation of steam on aerosol particles
D = Dry case; no steam condensation on aerosol particles
F = Fuchs collision efficiency
P-K = Pruppacher-Klett collision efficiency

* without Stephan Flow
+ dry case, unclear what was used by JAERI

TABLE 9.6

SURRY-AB FINAL PLATED MASS (kg)

| CODE | ACTUAL | EMULATED | RATIO |
|---|---|---|---|
| HAA-4 (W,P-K) | 950 | 327 | 0.344 |
| HAA-4 (W,F) | 857 | 315 | 0.368 |
| HAA-4 (D,F) | 991 | 1870 | 1.887 |
| HAARM-S (D,F)* | 1.86 | 1.91 | 1.027 |
| RETAIN-S (D,P-K) | 1370 | 2090 | 1.526 |
| NAUA-4 (D,F)* | 1.37 | 1.18 | 0.861 |
| SWNAUA (D,P-K) | 1049 | 1878 | 1.790 |
| SWNAUA (W,P-K) | 1023 | 253 | 0.247 |
| SWNAUA (D,F) | 810 | 1659 | 2.048 |
| REMOVAL (D,F)+ | 2115 | 1928 | 0.912 |
| AEROSOLS-B1 (D,F) | 930 | 1653 | 1.777 |
| AEROSIM-M (D,F) | 724 | 1679 | 2.319 |
| AEROSIM-M (W,F) | 624 | 277 | 0.444 |
| CONTAIN (W,F) | 706 | 200 | 0.283 |

ACTUAL = Results submitted by the actual code user
EMULATED = Results obtained with test code emulating actual code models
RATIO = Emulated Result divided by actual result; mean = 1.131 ± 0.741

W = Wet case: condensation of steam on aerosol particles
D = Dry case; no steam condensation on aerosol particles
F = Fuchs collision efficiency
P-K = Pruppacher-Klett collision efficiency

* without Stephan Flow
+ dry case, unclear what was used by JAERI

TABLE 9.7

SURRY-AB FINAL LEAKED MASS (g)

| CODE | ACTUAL | EMULATED | RATIO |
|---|---|---|---|
| HAA-4 (W,P-K) | 3326 | 4087 | 1.229 |
| HAA-4 (W,F) | 2690 | 3453 | 1.284 |
| HAA-4 (D,F) | 3190 | 3015 | 0.945 |
| HAARM-S (D,F)* | 4029 | 4397 | 1.091 |
| RETAIN-S (D,P-K) | 6116 | 4646 | 0.760 |
| NAUA-4 (D,F)* | 2900 | 2545 | 0.878 |
| SWNAUA (D,P-K) | 2954 | 3205 | 1.085 |
| SWNAUA (W,P-K) | 2933 | 3445 | 1.175 |
| SWNAUA (D,F) | 1970 | 2270 | 1.152 |
| REMOVAL (D,F)+ | 2400 | 3366 | 1.403 |
| AEROSOLS-B1 (D,F) | 1700 | 2197 | 1.292 |
| AEROSIM-M (D,F) | 1800 | 2276 | 1.264 |
| AEROSIM-M (W,F) | 1700 | 2302 | 1.354 |
| CONTAIN (W,F) | 1691 | 734 | 0.434 |

ACTUAL = Results submitted by the actual code user
EMULATED = Results obtained with test code emulating actual code models
RATIO = Emulated Result divided by actual result; mean = 1.096 ± 0.265

W = Wet case: condensation of steam on aerosol particles
D = Dry case; no steam condensation on aerosol particles
F = Fuchs collision efficiency
P-K = Pruppacher-Klett collision efficiency

* without Stephan Flow
+ dry case, unclear what was used by JAERI

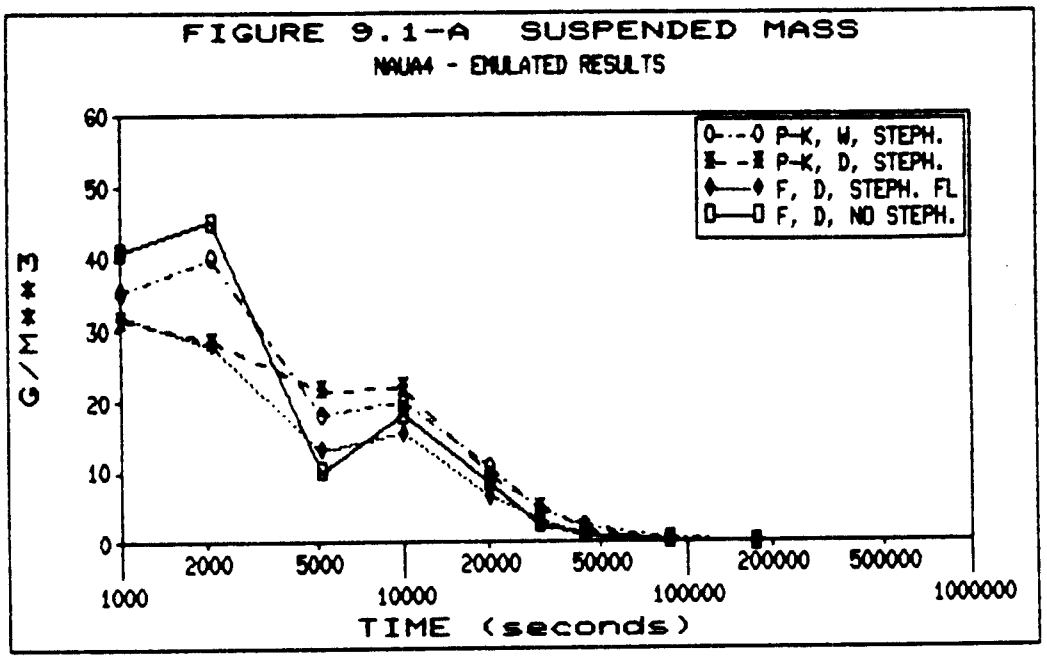


FIGURE 9.1-A SUSPENDED MASS
NAUA4 - EMULATED RESULTS
O--O P-K, W, STEPH.
X- -X P-K, D, STEPH.
♦—♦ F, D, STEPH. FL
□—□ F, D, NO STEPH.
G/M**3
60
50
40
30
20
10
0
1000
2000
5000
10000
20000
50000
100000
200000
500000
1000000
TIME (seconds)

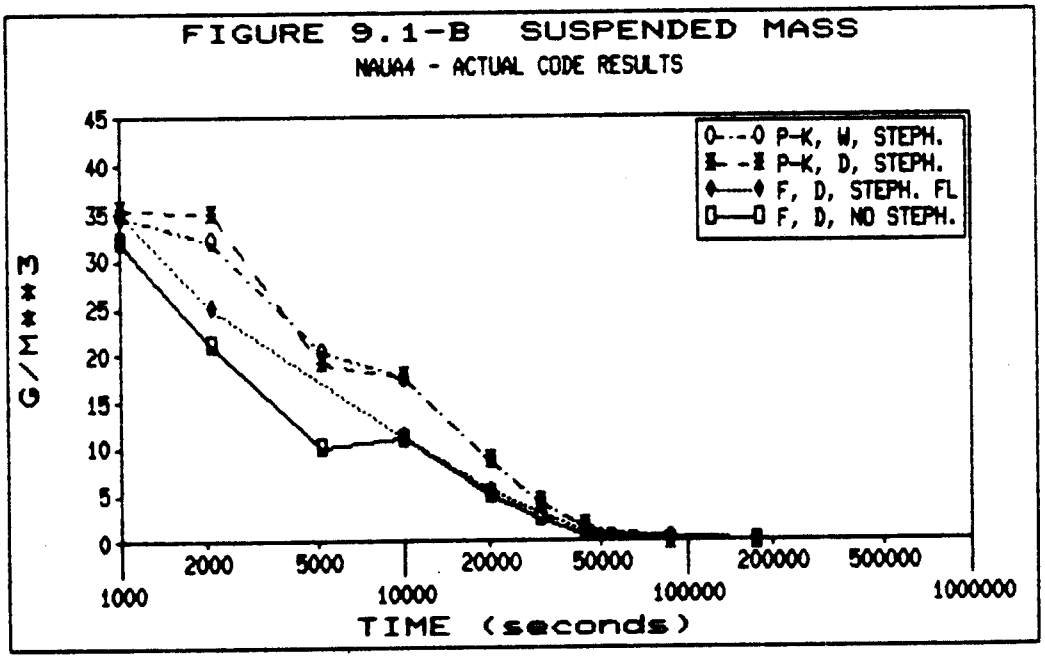


FIGURE 9.1-B SUSPENDED MASS
NAUA4 - ACTUAL CODE RESULTS
O--O P-K, W, STEPH.
X- -X P-K, D, STEPH.
♦—♦ F, D, STEPH. FL
□—□ F, D, NO STEPH.
G/M**3
45
40
35
30
25
20
15
10
5
0
1000
2000
5000
10000
20000
50000
100000
200000
500000
1000000
TIME (seconds)

FIGURE 9.2-A SETTLED MASS

NAUA4 - EMULATED RESULTS

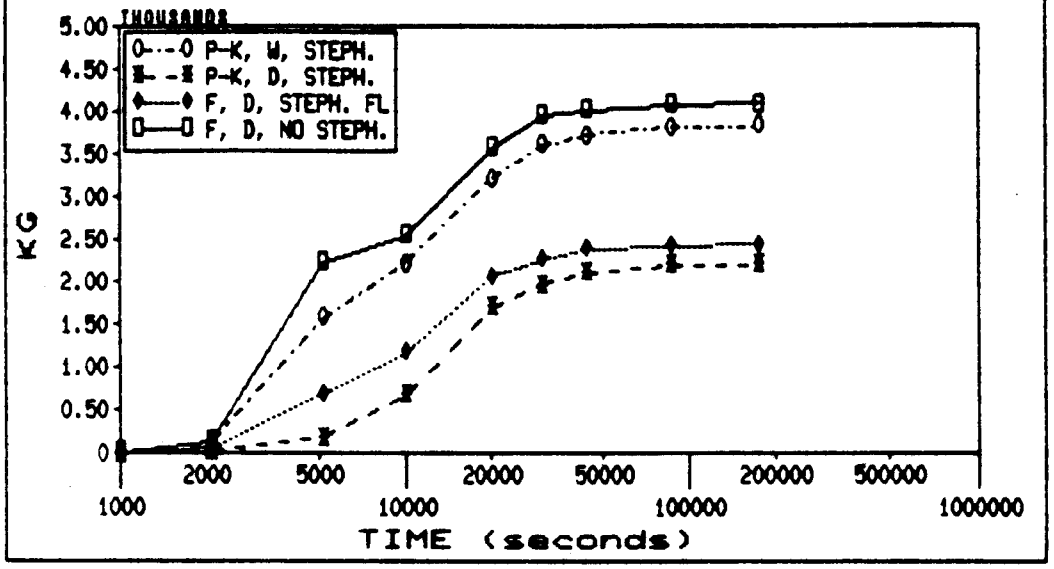
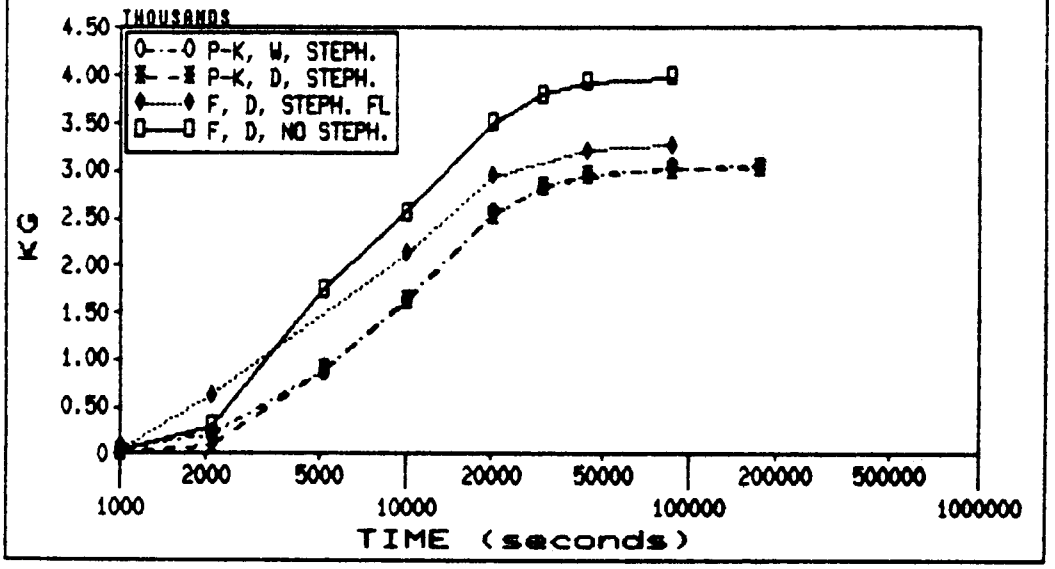



FIGURE 9.2-B SETTLED MASS

NAUA4 - ACTUAL CODE RESULTS



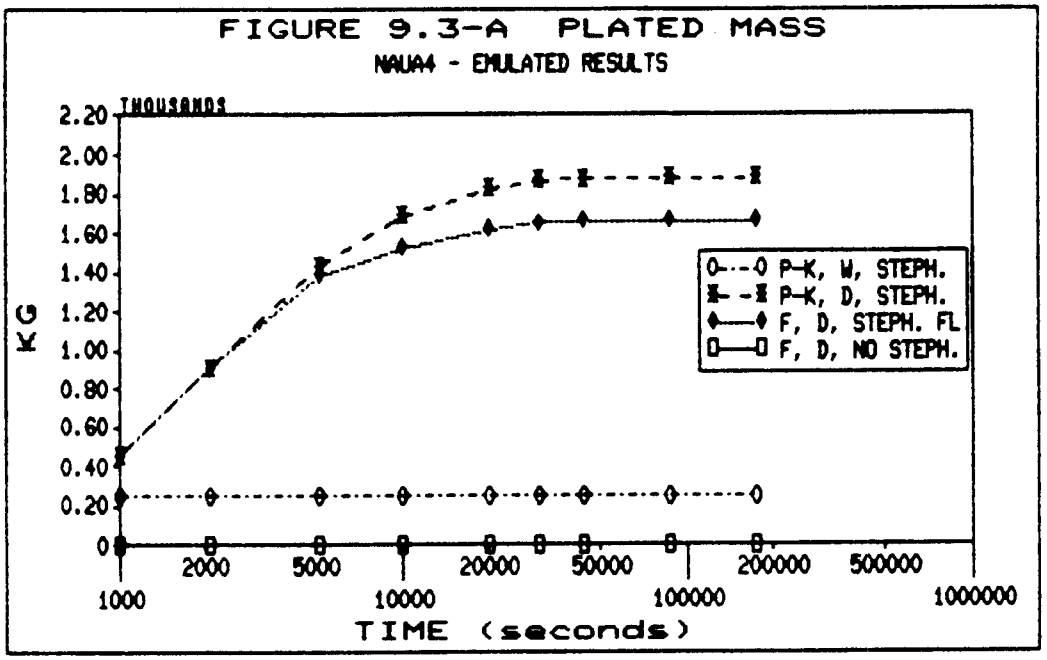


FIGURE 9.3-A PLATED MASS
NAUA4 - EMULATED RESULTS
THOUSANDS
2.20
2.00
1.80
1.60
1.40
1.20
1.00
0.80
0.60
0.40
0.20
0
KG
O---O P-K, W, STEPH.
X- -X P-K, D, STEPH.
♦—♦ F, D, STEPH. FL
□—□ F, D, NO STEPH.
1000
2000
5000
10000
20000
50000
100000
200000
500000
1000000
TIME (seconds)

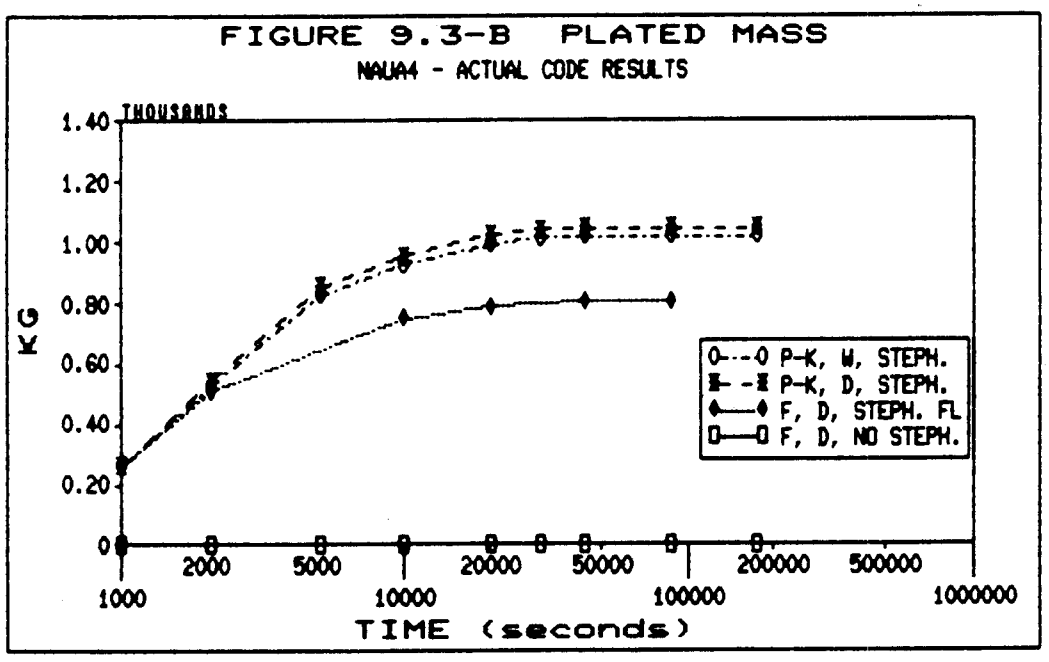


FIGURE 9.3-B PLATED MASS
NAUA4 - ACTUAL CODE RESULTS
THOUSANDS
1.40
1.20
1.00
0.80
0.60
0.40
0.20
0
KG
O---O P-K, W, STEPH.
X- -X P-K, D, STEPH.
♦—♦ F, D, STEPH. FL
□—□ F, D, NO STEPH.
1000
2000
5000
10000
20000
50000
100000
200000
500000
1000000
TIME (seconds)

FIGURE 9.4-A LEAKED MASS

NAUA4 - EMULATED RESULTS

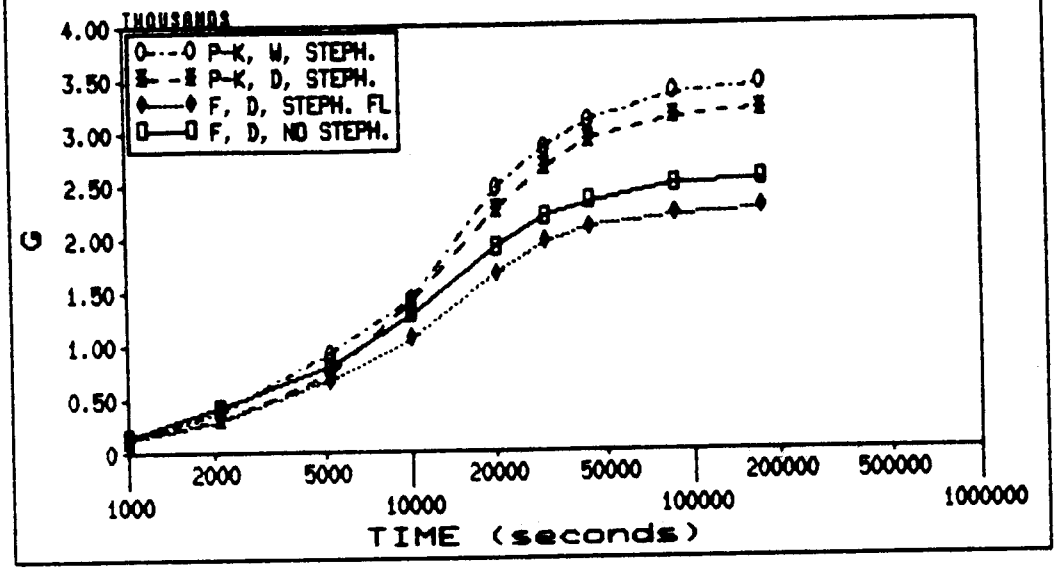



FIGURE 9.4-B LEAKED MASS

NAUA4 - ACTUAL CODE RESULTS