AECL-9099

ATOMIC ENERGY OF CANADA LIMITED

L'ENERGIE ATOMIQUE DU CANADA, LIMITEE

ASSESSING NUMERICAL METHODS USED IN NUCLEAR AEROSOL TRANSPORT MODELS

EVALUATION DE METHODES NUMERIQUES EMPLOYEES DANS LES MODELES DE MIGRATION DES AEROSOLS

B. H. McDonald

Whiteshell Nuclear Research Establishment

Etablissement de recherches nucléaires de Whiteshell

Pinawa, Manitoba R0E 1L0

1987

ATOMIC ENERGY OF CANADA LIMITED

ASSESSING NUMERICAL METHODS USED IN NUCLEAR AEROSOL TRANSPORT MODELS

by

B.H. MCDONALD

Whiteshell Nuclear Research Establishment
Pinava, Manitoba, Canada, ROE 1LO
1987

AECL-9099

ÉVALUATION DE MÉTHODES NUMÉRIQUES EMPLOYÉES DANS LES MODÈLES DE MIGRATION DES AÉROSOLS

par

B.H. McDonald

RÉSUMÉ

On utilise plusieurs programmes de calcul pour prédire le comportement des aérosols nucléaires libérés en enceinte de réacteur lors d'accidents hypothétiques de réacteurs refroidis à l'eau. Chacun de ces programmes utilise des méthodes numériques pour discrétiser et intégrer les équations régissant le processus de migraton des aérosols. Les calculatrices n'exécutent que des opérations algébriques et ne produisent que des nombres. C'est dans les méthodes numériques qu'on peut arriver à comprendre ces nombres et qu'on peut les relier à la résolution réelle des équations. Dans ce rapport, on examine les méthodes numériques les plus couramment employées dans les programmes de calcul de la migration des aérosols en tant que cas particuliers d'une technique de résolution générale, la Méthode des Restes Pondérés. Il semblerait que les méthodes numériques employées dans les programmes de calcul puissent toutes donner des solutions raisonnables du problème mathématique lorsqu'on les emploie avec habilité et précaution.

L'Énergie Atomique du Canada, Limitée
Établissement de recherches nucléaires de Whiteshell
Pinawa, Manitoba R0E 1L0
1987

AECL-9099

ASSESSING NUMERICAL METHODS USED IN NUCLEAR AEROSOL TRANSPORT MODELS

by

B.H. MCDONALD

ABSTRACT

Several computer codes are in use for predicting the behaviour of nuclear aerosols released into containment during postulated accidents in water-cooled reactors. Each of these codes uses numerical methods to discretize and integrate the equations that govern the aerosol transport process. Computers perform only algebraic operations and generate only numbers. It is in the numerical methods that sense can be made of these numbers and where they can be related to the actual solution of the equations. In this report, the numerical methods most commonly used in the aerosol transport codes are examined as special cases of a general solution procedure, the Method of Weighted Residuals. It would appear that the numerical methods used in the codes are all capable of producing reasonable answers to the mathematical problem when used with skill and care.

Atomic Energy of Canada Limited
Whiteshell Nuclear Research Establishment
Pinawa, Manitoba, Canada, ROE 1LO
1987

AECL-9099

CONTENTS

1. INTRODUCTION

Computer codes, used to predict the time-dependent behaviour of aerosol particles released from the primary heat transport systems of nuclear reactors during postulated severe accidents, are usually classified, in terms of numerical methods, as being "Method of Moments", "Finite Difference", or "Finite Element" models. This classification arises from the fashion in which the particle size distribution equation is discretized in the particle size dimension. In this report, the three basic models are described and compared as variations of a single method, the general "Method of Weighted Residuals". The methods have more in common than perhaps many code developers have realized, and a somewhat tutorial approach to a discussion of them is taken in this report.

Solution of the particle size distribution equation may be thought of as a two-step process. The equation itself is an integro-differential equation in the two variables: time, and particle size or mass. The first step is to convert the equation to a system of ordinary differential equations in time-only dependent variables. These variables may be viewed as parameters from which the size distribution may be constructed through specified functional relationships. The integro-differential terms in the particle size dimension are recast as time-dependent algebraic functions of the variables by performing the integro-differential operations on the specified functional relationships. The second step in the process is to integrate the system of ordinary differential equations producing the time history of these variables.

The essential way in which moment, finite difference, and finite element methods differ is in the specification of the time-only dependent variables, the functional relationships from which the particle size distribution may be constructed from these variables, and the operations by which the integro-differential terms are converted to algebraic functions of these variables.

In moment methods, found in the RETAIN and HAA families of codes, one assumes that the particle size distribution is lognormal, in which case there are three variables: the total particle number density, the geometric mean mass, and the logarithm of the geometric standard deviation. The set of three ordinary differential equations is generated by integrating over the size dimension, with integrands weighted by the zeroth, first, and second powers of mass, i.e., the first three moments. The integrations are performed analytically, with agglomeration kernels and removal rates expanded as the first few terms of a power series. Although the method is computationally efficient, the restriction to a single shape for the size distribution and the need to express physical models as power series make the method somewhat limited in application.

In finite difference methods, found in CONTAIN, REMOVAL, and in the NAUA and AEROSIM families, one divides the size domain into a fixed number of ranges, or "bins", and assumes that particles exist only at the discrete bin-mid-point sizes. Particle densities at these discrete sizes become the time-dependent variables. An ordinary differential equation is generated for the time behaviour of each of these variables by integrating over each bin. The integrands are often weighted by mass, and the

integrations are performed numerically. Although the method is flexible, one must ensure that there are sufficient bins to accommodate the smallest and the largest particles, and to provide enough detail in the distribution. In CONTAIN, any particle that grows to a size larger than the largest expected particle is immediately put on the floor. This is one way around the problem, but code users should ensure that not too many particles are removed by this somewhat artificial device. Numerical integration over the bins must be done carefully to ensure that mass is conserved, especially for agglomeration. Care must be taken in allocating the mass of agglomerated particles when the combined mass does not exactly coincide with any of the bin-mid-point sizes.

In the finite element method, as found in AEROSOLS/B1, one also divides the size domain into a fixed number of bins, but then one assumes that the logarithm of the particle density varies linearly between bin-end-point values. Particle densities at the end points become the time-dependent variables, and the distribution is piecewise linear in particle size on a log scale. The ordinary differential equations are obtained by performing the size dimension integrations and differentiations numerically, using the assumed piecewise linearity, for each of the bin end-points. Advantages and disadvantages are similar to those for finite difference methods.

This study of code numerics was part of light-water reactor (LWR) aerosol code comparison exercise organized through the Organization for Economic Cooperation and Development (OECD) [1,2]. A listing of the codes reviewed may be found in Table 1.1. Other studies comparing aerosol behaviour code performance have been done [3-9]. Information included here has been based on documentation available from the code authors and users [10-24]. Incomplete documentation may have resulted in omissions, which are to be noted in the following text and tables.

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., ξ, 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:

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

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 ϕ, in the limit as t approaches ∞, 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., ϕ, S, ξ 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 (ϕ, R, ξ, 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., ϕ, 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 ρ 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 ξ. Alternatively, one may integrate Equation (2.6) over a small mass range:

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

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, ϕ, the condensation rate, ξ, 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

$$\begin{aligned}\frac{\partial}{\partial t} q_k(m,t) = {} & \int_0^m d\mu\ \phi(m-\mu,\mu)q_k(\mu,t)C(m-\mu,t) \\ & - q_k(m,t)\int_0^\infty d\mu\ \phi(m,\mu)C(\mu,t) \\ & + \delta_{ik}\xi(m,t)C(m,t) - \frac{\partial}{\partial m}[\xi(m,t)q_k(m,t)] \\ & - R(m,t)q_k(m,t) + S_k(m,t). \end{aligned} \quad (2.9)$$

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

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

All terms are equivalent to those preceding except for the term $\delta_{ik}\ \xi(m,t)\ C(m,t)$. The Kronecker delta function δ_{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.

3. 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 multi-compartment 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 [26]. Briefly, given an integro-differential equation of the form of (2.1), which may be expressed as

$$\frac{\partial}{\partial t} C(m,t) = F[C(m,t),m,t] \qquad (3.1)$$

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

$$\Lambda(m,t) = F[C(m,t),m,t] - \frac{\partial}{\partial t} C(m,t) \tag{3.2}$$

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

$$I(t) = {}_0\int^{\infty} dm\ W(m)\Lambda(m,t) = {}_0\int^{\infty} dm\ W(m)\left\{F[C(m,t),m,t] - \frac{\partial}{\partial t}C(m,t)\right\}. \tag{3.3}$$

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 (3.1) is solved.

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

$$C(m,t) = f[m,\underline{Z}(t)] \tag{3.4}$$

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

$$\underline{Z}(t) = [Z_1(t),\ Z_2(t),\ \ldots,\ Z_N(t)]. \tag{3.5}$$

If the $Z_i(t)$ variables are known, then C(m,t) can be constructed from (3.4). The N required ordinary differential equations are obtained by using N different weighting functions $W_i(m)$ in the equation

$${}_0\int^{\infty} dm\ W_i(m)f(m,\frac{dZ}{dt}) = {}_0\int^{\infty} dm\ W_i(m)F[f(m,\underline{Z}),m,t]\ ;\quad i = 1,\ldots N. \tag{3.6}$$

Note that this assumes the integrated residual, I(t) in Equation (3.3), vanishes throughout. This can be recast, with linear functions f, in an N x N system, represented as the matrix-vector equation:

$$[A]\frac{d\underline{Z}}{dt} = \underline{b}(\underline{Z},t) \tag{3.7}$$

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

$$\underline{Z}(t + \Delta t) = \underline{Z}(t) + \Delta t[A]^{-1}\underline{b}[\underline{Z}(t),t]. \tag{3.8}$$

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

$$\frac{d\underline{Z}^*}{dt} = [A]\frac{d\underline{Z}}{dt} = \underline{b}(\underline{Z},t) \tag{3.9}$$

whence

$$\underline{Z}^*(t + \Delta t) = \underline{Z}^*(t) + \Delta t\underline{b}[\underline{Z}(t),t]. \tag{3.10}$$

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 (3.6) and (2.1) are:

$$\int_0^\infty dm\ W_i(m) f(m, \frac{d\underline{Z}}{dt}) = \sum_{j=1}^{5} I_j \quad ; \quad i = 1,..N \tag{3.11a}$$

where

$$I_1 = \int_0^\infty dm\ W_i(m) \left[\frac{1}{2} \int_0^m d\mu\ \phi(\mu, m-\mu) f(\mu, \underline{Z}) f(m-\mu, \underline{Z}) \right] \tag{3.11b}$$

$$I_2 = \int_0^\infty dm\ W_i(m) \left[-f(m, \underline{Z}) \int_0^\infty d\mu\ \phi(\mu, m) f(\mu, \underline{Z}) \right] \tag{3.11c}$$

$$I_3 = \int_0^\infty dm\ W_i(m) [S(m,t)] \tag{3.11d}$$

$$I_4 = \int_0^\infty dm\ W_i(m) [-R(m,t) f(m, \underline{Z})] \tag{3.11e}$$

$$I_5 = \int_0^\infty dm\ W_i(m) \left\{ -\frac{\partial}{\partial m} [\xi(m,t) f(m, \underline{Z})] \right\} \tag{3.11f}$$

3.1 METHODS OF MOMENTS CODES

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

$$C_L(m,t) = \frac{C_T(t)}{[2\pi]^{1/2} 3\alpha_g(t)\ m} \exp\left\{ -\frac{1}{2} \left[\frac{\ln \frac{m}{m_g(t)}}{3\alpha_g(t)} \right]^2 \right\} \tag{3.1.1}$$

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 (3.6),

$$\frac{d\underline{Z}_i^*}{dt} = \int_0^\infty dm\ m^{i-1} F[C_L(m, \underline{Z}), m, t] \quad ; \quad i = 1,2,3. \tag{3.1.2}$$

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,

$$\underline{X} = \underline{Z}^*(t + \Delta t) \tag{3.1.3}$$

then

$$C_T(t + \Delta t) = X_1 \tag{3.1.4a}$$

$$m_g(t + \Delta t) = \frac{X_2^2}{[X_1^3 X_3]^{1/2}} \tag{3.1.4b}$$

$$\alpha_g(t + \Delta t) = \frac{1}{3} \ln\left[\frac{X_1 X_3}{X_2^2}\right]^{1/2} \tag{3.1.4c}$$

Agglomeration, source, and removal integrals are handled by expanding the agglomeration kernel, ϕ, and the removal rate, R, as power series in the size coordinates (m and μ). 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.

3.2 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

$$M_i = \frac{m_{i-1} + m_i}{2} \quad ; \quad m_{i-1} < m_i \quad ; \quad i = 1,..N \tag{3.2.1}$$

with $$h_i = m_i - m_{i-1} \quad ; \quad i = 1...N \tag{3.2.2}$$

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

$$A_i = [m_{i-1}, m_i) \; ; \; i = 1...N. \tag{3.2.3}$$

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

$$C(m,t) = C(M_i,t) = Z_i(t) \quad ; \quad i = 1,...N. \tag{3.2.4}$$

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

$$W_j(m) = \begin{cases} m & ; \; m_{j-1} \le m < m_j \\ 0 & ; \; \text{otherwise} \end{cases} \tag{3.2.5}$$

and for the NAUA family of codes

$$W_j(m) = \begin{cases} 1 & ; \; m_{j-1} \le m < m_j \\ 0 & ; \; \text{otherwise.} \end{cases} \tag{3.2.6}$$

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

$$Z_i^* = h_i Z_i \tag{3.2.7}$$

and for AEROSIM

$$Z_i^* = Z_i/M_i. \tag{3.2.8}$$

Application of the method of weighted residuals to the basic equation—leaving out for the moment the condensation term (3.11f)—results in the system of equations where A_i is the range of integration defined by Equation (3.2.3),

$$\int_{Ai} dm\ m^L \frac{dZ_i(t)}{dt} = \int_{Ai} dm\ m^L F[Z_i(t), m, t]\ ;\quad i = 1 \ldots N \tag{3.2.9a}$$

where

$$L = \begin{cases} 0 & \text{for NAUA versions} \\ 1 & \text{for all others.} \end{cases} \tag{3.2.9b}$$

A result proved in [4], which states that

$$\int_{Ai} dm\ m^L \int_0^m d\mu\ F(\mu, m-u) = \int_0^{m_i} d\mu \int_0^{m_i} d\nu\ (\mu+\nu)^L \theta_i(\mu+\nu) F(\mu,\nu) \tag{3.2.10a}$$

where

$$\theta_i(\mu+\nu) = \begin{cases} 1 & \text{for } m_{i-1} \le \mu+\nu < m_i \\ 0 & \text{otherwise} \end{cases} \tag{3.2.10b}$$

allows Equation (3.2.9a) to be expanded as follows:

$$\begin{aligned} \frac{d}{dt} \int_{Ai} dm\ m^L Z_i(t) = {} & \frac{1}{2} \sum_{j=1}^{i} \sum_{k=1}^{i} \int_{Aj} d\mu \int_{Ak} d\nu\ (\mu+\nu)^L\ \theta_i(\mu+\nu)\phi(\mu,\nu) Z_j Z_k \\ & - \sum_{j=1}^{N} \int_{Ai} dm\ m^L Z_i \int_{Aj} d\mu\ \phi(\mu,m) Z_j \\ & + \int_{Ai} dm\ m^L S(m,t) - \int_{Ai} dm\ m^L R(m,t) Z_i. \end{aligned} \tag{3.2.11}$$

Condensation is most readily included by using an upwind differencing scheme [8], resulting, for example, in the addition to the right side of (3.2.11) of I_5, where

$$I_5 = -\int_{Ai} dm\ m^L \nabla_{\xi z} \tag{3.2.12a}$$

$$\nabla_{\xi z} = \begin{cases} \dfrac{\xi(M_i) Z_j - \xi(M_{i-j}) Z_{i-1}}{m_i - m_{i-1}}\ ; & \xi(M_i) > 0 \qquad (3.2.12b) \\[2ex] \dfrac{\xi(M_{i+1}) Z_{i+1} - \xi(M_j) Z_i}{m_{i+1} - m_i}\ ; & \xi(M_i) < 0 \qquad (3.2.12c) \end{cases}$$

with suitable convections for mass range end-points. The particular integrals that remain to be discretized, from (3.2.11), are:

$$\psi_i^A = \int_{A_i} dm\ m^L \tag{3.2.13a}$$

$$\psi_{ij}^B = \int_{A_i} dm\ m^L \int_{A_j} d\mu\ \phi(\mu,m) \tag{3.2.13b}$$

and

$$\psi_{jk}^C = \int_{A_j} d\mu \int_{A_k} d\nu(\mu+\nu)^L \theta_i(\mu+\nu)\phi(\mu,\nu). \tag{3.2.13c}$$

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

$$\psi_i^A \frac{d}{dt} Z_i = \frac{1}{2} \sum_{j=1}^{i} \sum_{k=1}^{i} \psi_{jk}^C Z_j Z_k - Z_i \sum_{j=1}^{N} \psi_{ij}^B Z_j + S\psi_i^A - R\psi_i^A Z_i - \psi_i^A \nabla_{\xi z} \quad ; \quad i = 1....N. \tag{3.2.14}$$

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 3.1.

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.

3.3 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

$$Z_i = Z(m_i) \; ; \quad i = 0.....N \tag{3.3.1}$$

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

$$Z(m) = \frac{(m_i - m)\, Z_{i-1} + (m - m_{i-1})Z_i}{m_i - m_{i-1}} \; ; \quad m_{i-1} \leq m < m_i. \tag{3.3.2}$$

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

$$g_i(m) = 1 \; ; \quad m = m_i \tag{3.3.3a}$$

$$g_i(m) = 0 \; ; \quad m = m_j, \text{ with } i \neq j \tag{3.3.3b}$$

and

$$g_i(m) = \frac{m_{i+1} - m}{m_{i+1} - m_i} \; ; \quad m_i \leq m < m_{i+1} \tag{3.3.3c}$$

$$g_i(m) = \frac{m - m_{i-1}}{m_i - m_{i-1}} \; ; \quad m_{i-1} \leq m < m_i. \tag{3.3.3d}$$

With these definitions we have

$$Z(m) = \sum_{i=0}^{N} g_i(m) Z_i. \tag{3.3.4}$$

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

$$W_j = \delta(m - m_j). \tag{3.3.5}$$

Thus

$$\int_0^\infty W_j f(m)dm = f(m_j). \tag{3.3.6}$$

Application of this to the basic equation (3.11), leaving out the condensation term, results in

$$\begin{aligned} \frac{d}{dt} Z_i = & \int_0^{m_i} d\mu \; \phi(\mu, m_i - \mu)\, Z_i \sum_{j=0}^{N} g_j(\mu) Z_j \\ & - Z_i \int_0^\infty d\mu \; \phi(m_i, \mu) \sum_{j=0}^{N} g_j(\mu) Z_j \\ & + S(m_i, t) - R(m_i, t) Z_i + I_s. \end{aligned} \tag{3.3.7}$$

The condensation term is established using methods similar to those used in finite difference method, (3.3.12), and $I_5 = \nabla_{\xi z}$.

Integrals in (3.3.7) 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].

3.4 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.

4. 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 (3.8), 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 (3.2.14) 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 3.4, 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 3.10) 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].

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 4.1 through 4.3. The data shown are cumulative settled mass (Table 4.1), plated mass (Table 4.2), and leaked mass (Table 4.3) 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.

5. CLOSURE

The integro-differential equations describing the dynamic behaviour of wet aerosol particles present a formidable challenge to the code developer. Several different approaches have been taken in the development of the currently available codes for prediction of the behaviour of nuclear aerosols in containment. However, when viewed from the perspective of the general methods of weighted residuals, these various approaches tend to come together and may be seen as simply variations, or special cases, of the general method. It would be interesting to examine the Galerkin Method as a possible tool for handling the problem, for this method is perhaps the most well-known and widely used in the family [26]. It possesses convergence properties that are better understood, and is often exceptionally stable.

Moment methods, as applied in the context of nuclear aerosols, have unfortunately been restricted to only lognormal particle size distributions and to analytical integration of the agglomeration kernels and removal rates. There is no real reason why numerical integration procedures cannot be em[illegible]ed, for example as in the test code reported above. Numerical inte[illegible]on would remove the difficulty in modifying or extending the physical [illegible]ls. It is sometimes debatable whether the analytical integration of the first few terms of a power series representation of a nonlinear function produces a more accurate result than a fully numerical integration of the actual nonlinear function itself. As well, if one is willing to consider more than just the first three moments, useful in describing the log-normal distribution, then more complex and perhaps more representative particle size distributions could be generated.

Discrete, or "bin", methods provide much flexibility in the shape of the size distribution and in the methods of selection of the time-dependent variables and functional relationships. Users must ensure that a sufficient number of bins exists in a calculation to accommodate all expected particle sizes, and to ensure sufficient detail in the resulting distribution itself. During early testing of the discrete test code described above, it was sufficient for computer program debugging to specify three bins: one for tiny particles, one for average particles, and the other for huge particles. The detail was abysmal, and the numerical diffusion was horrendous, but the program ran well and quickly. The penalty for having too many bins for the detail required is excessive computing effort.

No matter which method is used to recast the original integro-differential equation into a set of coupled ordinary differential equations, one is still faced with performing the time integration. A simple backwards Euler scheme with automatic time-step-size control is employed in the local test code, but more accurate methods are found in practice, with predictor corrector schemes, imbedded Runge-Kutta algorithms, and so forth. The nonlinear nature of the system almost precludes choosing a "best" integrator. One is again faced with a trade-off: very accurate solutions require long computer runs, whereas fast-running computer codes, of necessity, must lose some of the detail (the high-frequency information will be lost in temporal integration).

It is always instructive to test codes on simple problems, where the answer is known, to characterize errors of discretization. Testing codes for their sensitivity to variation in parameters and input specification is also helpful for code characterization. Such testing can be useful for establishing some level of confidence when the code in run for a realistic scenario where the answer is not known.

Code users should remember that their codes are calculating algebraically a set of numbers that they hope approximate the true solution of the mathematical problem. Greater accuracy generally requires more numbers, and consequently, greater computing costs. At this time, there does not appear to be a single numerical method demonstrably superior to all the others for generating approximate solutions to the nonlinear integro-differential equations describing aerosol particle behaviour. When used with skill and care, each of the numerical methods found in these nuclear aerosol transport codes should be able to produce reasonable answers to the mathematical problem. Answers to the physical problem, however, depend on the physical models used in the codes, and some uncertainty still exists there.

6. SYMBOLS

A_i	Range of discrete bin i
$[A]$	Matrix of coefficients
$\underline{b}$	Vector of source terms
$C(m,t)$	Number of particles of mass m at time t
C_T	Particle total number density lognormal distribution
$C_L(m,t)$	Number of particles of mass m at time t (log-normal distribution)
$F(C(m,t),m,t)$	Function describing particle size time rate of change
$g_i(m)$	Interpolating polynomial function i
h_i	Length of discrete mass range i
I_j	Particular integral j
m_g	Geometric mean mass (lognormal 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
$q_k(m,t)$	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	Particle radius
$S(m,t)$	Source rate of particles of mass m at time t
t	Time
W_i	Method of residuals weighting function i
Z_i	Method of residuals time-only dependent variable i
α_g	Logarithm of geometric standard deviation (lognormal distribution).
δ_{ik}	Kronecker delta function
Δm	Increment in mass
Δt	Increment in time
$\nabla_{\xi z}$	Numerical condensation gradient function
$\phi(m,n)$	Agglomeration kernel: collision rate, particles of masses m and n
$\Theta(m,n)$	Discrete mass range selector function
$\Lambda(m,t)$	Residual in Method of Weighted Residuals
μ, ν	Particle mass
ψ	Particular numerical integration term
$\xi(m,t)$	Condensation rate of steam on particles of mass m at time t

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 and 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, pp. 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. 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 Studsvik Energiteknik AB and is obtainable from the NEA Data Bank).

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

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

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

21. 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.

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

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

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

25. 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.

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

27. 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.

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 3.1

TIME INTEGRATION METHODS

CODE	BASIC INTEGRATION METHOD
CONTAIN	RUNGE-KUTTA with time-step 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 4.1

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-Plett collision efficiency

* without Stephan Flow
+ dry selected for emulation, not clear what was used by JAERI

TABLE 4.2

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-Plett collision efficiency

* without Stephan Flow
+ dry selected for emulation, not clear what was used by JAERI

TABLE 4.3

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 $\pm$ 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-Plett collision efficiency

* without Stephan Flow
+ dry selected for emulation, not clear what was used by JAERI

ISSN 0067-0367

To identify individual documents in the series we have assigned an AECL—number to each.

Please refer to the AECL—number when requesting additional copies of this document from

Scientific Document Distribution Office
Atomic Energy of Canada Limited
Chalk River, Ontario, Canada
K0J 1J0

Price: A

ISSN 0067-0367

Pour identifier les rapports individuels faisant partie de cette série, nous avons affecté un numéro AECL—à chacun d'eux.

Veuillez indiquer le numéro AECL—lorsque vous demandez d'autres exemplaires de ce rapport au

Service de Distribution des Documents Officiels
L'Énergie Atomique du Canada, Limitée
Chalk River, Ontario, Canada
K0J 1J0

Prix: A