NUREG/CR-5162
SAND88-0745

CHARM: A Model for Aerosol Behavior in Time Varying Thermal-Hydraulic Conditions

Prepared by C.J. Wheatley

Sandia National Laboratories

Prepared for
U.S. Nuclear Regulatory
Commission

NUREG/CR-5162
SAND88-0745
R3

CHARM: A Model for Aerosol Behavior in Time Varying Thermal-Hydraulic Conditions

Manuscript Completed: April 1988
Date Published: August 1988

Prepared by
C.J. Wheatley

Sandia National Laboratories
Albuquerque, NM 87185

Prepared for
Division of Systems Research
Office of Nuclear Regulatory Research
U.S. Nuclear Regulatory Commission
Washington, DC 20555
NRC FIN A1342

ABSTRACT

CHARM is a computer model for the behavior of a one component aerosol in a single region with time-varying external conditions. It treats particle agglomeration due to Brownian motion, gravity and turbulence, and particle deposition due to Brownian motion, gravity, turbulence, thermophoresis and diffusiophoresis. Turbulence properties are estimated for flow through a region of arbitrary cross-sectional shape, with aerodynamically rough or smooth walls at any Reynolds number. The gas can be of any composition. The time-varying external conditions allowed for are the temperature, pressure and velocity of the gas, wall temperatures, and the rate, mass median radius and geometric standard deviation of the source. The model is simply modified to enable this list to be extended if needed. A new method of solving the governing equations, based on the finite element collocation method, enables the time-varying conditions to be treated accurately and economically. We describe in detail the models, the numerical methods, the execution of the computer code (including how to write the input data file and interpret results), and how to make simple modifications to the model. We discuss how the model could be implemented as a submodel of a larger one and what further work needs to be done to enable it to efficiently treat multicomponent aerosols, and condensation onto and evaporation from particles.

CONTENTS

1 INTRODUCTION

The work which has culminated in the production of the aerosol model CHARM described in this manual originates from the need to consider hypothetical accidents in power generating nuclear reactors for the purposes of assessing safety standards. Some of these hypothetical accidents are classified as severe, in which the reactor core, as a consequence of overheating, becomes severely degraded and fission products are released into the primary circuit and possibly into the containment building and the environment. It is essential to these assessments to calculate the transport of fission products from the fuel to their eventual destination. Many of them can be transported as aerosols for which agglomeration and deposition are important mechanisms to be considered. The behavior of the aerosol is coupled to the thermal-hydraulic conditions which can vary considerably both spatially and temporally; variations in temperature, pressure and turbulence are particularly important. Condensation onto and vaporization of the aerosol and self-heating due to radioactive decay of the fission products can also occur. Many computational cells and many aerosol components must be considered.

This poses a significant computational problem. The model we describe solves one aspect of this: how to efficiently compute aerosol agglomeration and deposition taking account of the time varying thermal-hydraulic conditions. Numerical methods for treating this were formulated and assessed by the author (Wheatley, 1988) who showed that one in particular, based on the finite-element collocation method, can give satisfactorily accurate results for practical problems and yet requires only modest computational overhead to update the agglomeration kernel to take account of time-dependencies in the external conditions.

CHARM is a modest extension of the computer code used by the author to assess the collocation method. It models aerosol behavior in a single computational cell; time-varying external conditions are assumed to have been calculated in advance and are supplied as data to the model. The aerosol particles have a single, constant composition and can agglomerate, deposit on surfaces and leak from the cell. A time-varying source of particles can also exist within the cell. The agglomeration and deposition models in MAEROS (Gelbard,1982) have been used in CHARM. However, models have been added to treat deposition by turbulence and to estimate boundary layer thicknesses and turbulence properties of the flow field. These extensions in conjunction with improvements to the input and output subroutines will enable the model to be widely used for practical problems.

Of course, this only goes part-way to modeling aerosol behavior in the wider context as discussed above. Principally, multicomponent and vaporization effects are ignored. However, it was considered worth-while to develop such a model based on the previous work of the author for the following reasons:

i We know of no other model which is able to efficiently treat aerosol behavior in time-varying external conditions. As an example, it can be applied to aerosols moving through a pipe with temperature variations along its length. Perhaps more important, it can be used to study whether variations of the external variables with time are important to aerosol behavior.

ii It can be used to study alternatives to the physical models: turbulent deposition in particular for which there is some uncertainty.

iii With minor modification, it can be incorporated into multi-cell models such as VICTORIA (Grimley et al., 1988) thereby enabling aerosol behavior to be fully coupled to the thermal-hydraulics and other phenomena - though, of course, simplifying assumptions have to be made about how to treat the particle composition, and condensation and evaporation.

iv With further development of methods, it could form the basis of a model which efficiently treats multicomponent and vaporization effects also.

We envisage, therefore, that CHARM could be used as it is or with minor modification, incorporated as part of another model, or extended in some major way to treat multicomponent and vaporization effects. To meet all these possible needs, we will describe in detail the models in CHARM, the numerical methods we use, how to compose the input data file and interpret the output, and how to execute and make minor modifications to the code. We include supplementary details about the overall operation of the model, what the subroutines do and a compiler source listing. We also discuss in general terms how it could be incorporated into multi-cell models, the developments needed to treat efficiently multicomponent and vaporization effects, and how it might form the basis of a model to treat these effects.

2 DESCRIPTION OF THE MODEL

2.1 Overview

The governing equation of the aerosol phenomena treated by CHARM is the following

$$\frac{\partial C(m,t)}{\partial t} = \frac{1}{2}\int_0^m K(\mu,m-\mu,t)C(\mu,t)C(m-\mu,t)d\mu$$

$$- C(m,t)\int_0^\infty K(\mu,m,t)C(\mu,t)d\mu - R(m,t)C(m,t) + S(m,t) \quad . \quad (1)$$

where $C(m,t)$ is the aerosol number concentration distribution. It is defined so that $C(m,t)dm$ is the number of particles per unit volume with mass in the range m to $m + dm$ at time t. Drake (1972) gives an excellent survey of this equation.

K is the agglomeration kernel; it is symmetric and takes account of particle-particle collisions due to Brownian motion, differential gravitational settling, and turbulence. The integrand of the first term with K corresponds to production of particles of mass m due to collision and coalescence of particles of masses $m - \mu$ and μ. The integrand of the succeeding term corresponds to destruction of particles of mass m due to collision and coalescence with particles of mass μ. K can depend explicitly on time through changes in the external conditions; gas temperature, pressure and velocity are particularly important.

R is the removal rate for particles of mass m. Leakage, and deposition onto surfaces induced by thermophoresis, diffusiophoresis, gravitational settling, Brownian diffusion, and turbulence all contribute to R which, like K, may be time-dependent.

S is the number concentration source rate for particles of mass m.

Implicit in Eq. (1) are four assumptions which we briefly mention. First, the aerosol is well-mixed throughout the cell. This may require the flow in the cell to be turbulent to promote mixing or the cell to be a small part of a larger region within which the aerosol exists and the aerosol has nearly uniform properties within the cell by virtue of its small size. Second, particles are characterized by their mass only, i.e. particles can have a shape which is a function of m but particles of given mass all have the same shape. Clearly, this breaks down when particles having the same mass have different shapes. This affects agglomeration and deposition rates. Such dependence, however, would be extremely difficult to treat computationally. Third, boundary layers and their effect on deposition are not treated in detail. Fourth, particles do not break-up into smaller particles.

Detailed expressions for K, R and S will be considered in the succeeding subsections where the primary aim will be to give formulae in detail as they appear in CHARM and to reference their origin. We will indicate ranges of validity and possible uncertainties where appropriate. We recall that we have chosen to base the models in CHARM on those in MAEROS except where extensions have been made to estimate relevant flow parameters and turbulent deposition. A good survey of these and similar models is given in Dunbar et al. (1984). We use S.I. units throughout except for molecular weights, which have units kg / kmole, and in the table in Section 4.4 where we show the units explicitly.

2.2 Gas properties

Agglomeration and deposition rates depend on properties of the gas in the bulk of the cell and adjacent to surfaces, where they are relevant to the calculation of diffusiophoresis. Some of the formulae we give are specific to air. However, we note that they are isolated in a single subroutine in CHARM and so are easily changed to accomodate other gases. Standard gas kinetic formulae can be used to combine component properties to obtain estimates for mixtures.

The gas density For the bulk gas, the temperature, T, pressure, P, and average molecular weight, W_g, are supplied as input to the model (when a gas has more than one component the average molecular weight is just the mass of one mole of the mixture). The density, ρ_g, is calculated assuming the gas to be ideal as follows

$$\rho_g = \frac{PW_g}{10^3 RT} \quad . \tag{2}$$

The factor 10^3 is included because W_g has units kg / kmole.

The dynamic viscosity η_g is estimated from a correlation of data for air. The correlation is as follows

$$\eta_g = \eta_r \left[\frac{T}{T_{r1}} \right]^{3/2} \frac{1}{1+T/T_{r1}} \quad , \tag{3}$$

where η_r and T_{r1} are constants with values 1.565×10^{-6} and 114.0 respectively. It would be a simple matter to replace this with correlations for other gases or mixtures of gases.

The mean free path l is calculated from standard kinetic theory as follows

$$l = \eta_g \left[\frac{\pi}{2P\rho_g} \right]^{1/2} \quad . \tag{4}$$

The vapor diffusivity For the calculation of deposition by diffusiophoresis we need properties of the gas adjacent to surfaces where we suppose a vapor is present which is either condensing onto or evaporating from the nearby surface. The diffusivity of the vapor in the gas is estimated from a correlation of data for diffusion of water vapor in air as follows

$$D_v = D_{v,r} \frac{P_r}{P} \left[\frac{T}{T_{r2}} \right]^{1.94} , \qquad (5)$$

where $D_{v,r}$, P_r and T_{r2} are constants with values 2.11×10^{-5}, 1.01325×10^{5} and 273.15 respectively. It would be straightforward to replace this with correlations for other vapors and other gas mixtures.

The vapor concentration The concentration of the vapor in the vapor/gas mixture adjacent to a surface, c_s, where the subscript s can stand for c, w or f according to whether the surface is the ceiling, wall or floor (see Sub-section 2.6 for the definition of these surfaces), is estimated from the ideal gas law as follows

$$c_s = \frac{f_s P W_v}{10^3 R T_s} , \qquad (6)$$

where f_s is the molar fraction of the vapor in the mixture, W_v is the molecular weight of the vapor and T_s is the temperature of the mixture adjacent to the surface.

2.3 Flow properties

Turbulence parameters of the bulk flow are needed for estimating agglomeration and deposition rates. Viscous and diffusion boundary layer thicknesses are needed for estimating Brownian and thermophoresis deposition.

The friction velocity u_* is calculated from formulae based on those given in Schlichting (1979) for flow through a cylindrical pipe of any aerodynamic roughness and others for flow through aerodynamically smooth pipes of arbitrary cross-sectional shape. We will indicate in more detail below where extensions have been made. By definition, u_* in terms of the Fanning friction factor, f_*, is given by

$$u_* = U \left[\frac{f_*}{2} \right]^{1/2} , \qquad (7)$$

where U is the mean flow speed in the cell relative to surfaces in the cell, usually the cell walls.

The Fanning friction factor f_* is implicitly determined from a correlation established by Colbrook (1939) for flow through cylindrical pipes of any aerodynamic roughness. The correlation is

$$\frac{1}{2\sqrt{f_*}} = 1.74 - 2\log_{10}\left[\frac{2z_s}{d_h} + \frac{18.7}{Re2\sqrt{f_*}}\right], \qquad (8)$$

where z_s is the equivalent sand roughness of the pipe surface adjacent to the flow, d_h is the diameter of the pipe, and Re is the pipe Reynolds number equal to $\rho_g d_h U/\eta_g$.

This correlation reduces to the quadratic resistance formula for fully rough flow ($\rho_g z_s u_*/\eta_g > 70$), derived by von Karman, and Prandtl's universal law of friction for smooth pipes ($\rho_g z_s u_*/\eta_g < 5$). In either limit, the correlation has been verified up to Reynolds numbers which exceed 10^6 and there is theoretical justification for extrapolating the correlation to much larger Reynolds numbers.

The equivalent sand roughness z_s equals the height of protrusions on sand roughened pipes of equivalent roughness (this originates from the pioneering experiments of Nikuradse who measured the pressure drop along pipes roughened with sand of varying grades at Reynolds numbers ranging from somewhat less than 10^3 to greater than 10^6). z_s in general must be determined empirically but values have been established for common materials which we give in the table below, reproduced from Schlichting (1979).

z_s for some common materials

material	z_s (m)
reinforced concrete	.0003 - .003
cast iron	.00026
galvanized steel	.00015
structural and forged steel	.000045
drawn pipes	.0000015

Note that certain types of protrusion, such as regularly spaced ribs perpendicular to the flow, can give rise to values of z_s significantly larger than the height of the protrusion.

The hydraulic diameter d_h is just the diameter of the pipe. However, experiments by a number of authors have shown that the Fanning friction factor of aerodynamically smooth pipes of non-circular cross-section (such as square, rectangular, triangular and annuli) equals that of circular pipes over a wide range of Reynolds number when d_h is generalized as given in Eq. (9) below,

despite the complications of secondary flows induced in non-circular pipes. d_h in general is given by

$$d_h = \frac{4A}{p} , \tag{9}$$

where A is the cross-sectional area and p is the "wetted" perimeter of the pipe perpendicular to the flow. We use this generalization for aerodynamically rough pipes also, even though it has only been verified for smooth pipes.

The critical Reynolds number The above formulae for u_* are valid when the flow is turbulent and fully developed which, for a cylindrical pipe, is the case when $Re \geq 2300$ and well downstream of the pipe inlet. For convenience, we calculate u_* from these formulae when $Re \geq 2300$ irrespective of the cross-sectional shape of the pipe and the downstream distance from the inlet (and other axial changes). We note that the critical Reynolds number (based on d_h given by Eq. (9)) in general depends on the pipe shape.

We do not have a treatment for other turbulent flows, non-fully developed flows in particular, and simply assume that u_* is zero when $Re < 2300$.

The turbulent energy dissipation rate per unit mass ϵ_* is estimated by equating the rate of energy consumption needed to promote steady flow along a pipe of given length to the rate of energy dissipation due to turbulence with the result

$$\epsilon_* = \frac{4Uu_*^2}{d_h} . \tag{10}$$

For fully developed turbulent flow in pipes, Laufer (1954) shows that direct viscous energy dissipation is negligible compared to energy dissipation due to turbulence. Consequently, the above formula is valid and provides an estimate of the turbulent energy dissipation rate averaged of the pipe cross section. We note, however, that Laufer also showed the turbulent energy dissipation rate varies considerably with distance from the pipe centre, being least in the core region and greatest near the wall. An alternative weighting in the averaging process can therefore lead to a substantially different estimate of ϵ_* but we retain Eq. (10) in the absence of any present indication that an alternative weighting should be used.

The viscous boundary layer thickness δ_* is estimated from (Monin and Yaglom, 1971)

$$\delta_* = \frac{\eta_g}{\rho_g u_*} . \tag{11}$$

An empirical constant of ~5 might have been included in the numerator on the right hand side of Eq. (11) (Monin and Yaglom, 1971) but we prefer to leave this out since other empirical constants in the equations in which δ_* appears make this redundant.

The diffusion boundary thickness δ_D is needed later for estimating deposition by Brownian diffusion from a turbulent flow. It is estimated from (Keller, 1973)

$$\delta_D = \delta_* / Sc^{1/3} , \tag{12}$$

where Sc is the particle Schmidt number given by

$$Sc = \frac{\eta_g}{kTB} . \tag{13}$$

k is the Boltzmann constant and B is the particle mobility defined in Sub-section 2.5.

2.4 Initial and source distributions

For convenience we use analytic formulae for these although more general formulations could be accommodated. The initial and source number concentration distribution are chosen to be log-normal. Consequently, the initial density distribution, $Y(m,t) = mC(m,t)$, is given by

$$Y(m,t) = \frac{N}{(2\pi)^{1/2} \log_e(\sigma^3)} e^{-\{\log_e^2(m/m_g)/2\log_e^2(\sigma^3)\}} . \tag{14}$$

where the moments N, σ and m_g are defined in Sub-section 3.3. These moments, however, are not the most convenient to specify values for and so we choose instead ρ, σ and $r_{5\theta}$ as the primary parameters. ρ and σ are defined in Sub-section 3.3 also. $r_{5\theta}$ is the radius of the spherically equivalent particle of mass $m_{5\theta}$ and density ρ_p.

N and m_g in terms of ρ and $r_{5\theta}$ are

$$m_{5\theta} = \frac{4}{3} \pi \rho_p r_{5\theta}^3 , \tag{15}$$

$$N = \frac{\rho}{m_{5\theta}} e^{\log_e^2(\sigma^3)/2} , \tag{16}$$

and

$$m_g = m_{5\theta} e^{-\log_e^2(\sigma^3)} . \tag{17}$$

Similar expressions apply for the source density distribution, $mS(m,t)$, except N and ρ are replaced by dN/dt and $d\rho/dt$ where these are, respectively, the number concentration generation rate (the number generation rate divided by the cell volume) and the density generation rate (the mass generation rate divided by the cell volume) of the source. Note that we use the same symbols for other moments of both the airborne and source distributions.

2.5 Particle mobilities

The particle mobility, B, is estimated from Stoke's mobility law, B_{St}, with a slip correction due to Cunningham (1910), Cu, as follows

$$B = B_{St}\, Cu \ . \tag{18}$$

B_{St} is given by

$$B_{St} = \frac{1}{6\pi\chi_d\eta_g r} \ , \tag{19}$$

where the dynamic shape factor, χ_d, is an empirical dimensionless correction factor introduced to account for the aerodynamic effects associated with non-spherical particles, and r is the radius of the spherically equivalent particle of mass m.

Cu is given by

$$Cu = 1 + k_a Kn + k_q Kn\, e^{-k_b/Kn} \ , \tag{20}$$

where Kn is the particle Knudsen number which equals $1/r$ (i.e. the mean free path/the equivalent particle radius) and k_a, k_q and k_b are empirical dimensionless constants. The default values we use for k_q and k_b (0.4 and 1.1 respectively) are from Davies (1945). Our default value for k_a is slightly larger than the value obtained by Davies (1.37 c.f. 1.257).

2.6 Deposition rates

Five deposition mechanisms are considered which we describe in turn. We describe how the contributions are combined at the end of this sub-section.

We shall use ceiling, wall, and floor to denote surfaces exposed to aerosol (within the cell and at the cell boundaries) which are horizontal and downward facing, vertical, and horizontal and upward facing respectively. Surfaces of given orientation are not distinguished in temperature.

The deposition velocity due to gravitational settling v_g is given by

$$v_G = gmB \ , \tag{21}$$

where g is the acceleration due to gravity. Strictly, $(\rho_p - \rho_g)/\rho_p$ should appear as a multiplicative factor in this equation but, keeping to our constraint to duplicate the models in MAEROS, we ignore this small error.

The deposition velocity due to turbulence v_* is estimated from a correlation of the Liu and Agarwal (1974) data for deposition in smooth pipes. We first define the dimensionless deposition velocity, $\bar{v}_*$, and the dimensionless relaxation time, $\bar{\tau}$, by

$$v_* = \bar{v}_* u_* \tag{22}$$

and

$$\bar{\tau} = \tau \rho_g u_*^2 / \eta_g \ , \tag{23}$$

where τ is the particle relaxation time given by

$$\tau = \frac{2\rho_p r^2 Cu}{9\eta_g \chi_d} = mB \ . \tag{24}$$

For $\bar{\tau}$ between ~.1 to ~10 the Liu and Agarwal data are well correlated by

$$\bar{v}_{*1} = 6\times10^{-4}\ \bar{\tau}^2 \tag{25}$$

For $\bar{\tau}$ between ~30 to ~1000 their data are well correlated by

$$\bar{v}_{*2} = .213\ \bar{\tau}^{-1/8} \tag{26}$$

By combining $\bar{v}_{*1}$ and $\bar{v}_{*2}$, we find a good correlation across the whole range of $\bar{\tau}$ to be as follows

$$1/\bar{v}_*^2 = 1/\bar{v}_{*1}^2 + 1/\bar{v}_{*2}^2 \ . \tag{27}$$

This correlation is applied in CHARM without restriction on $\bar{\tau}$.

Liu and Agarwal concluded that the dominant deposition mechanism in their experiments was penetration of the laminar sub-layer by particle inertia generated from the turbulence. They showed that the contribution from Brownian diffusion across the laminar sub-layer was always negligible for the particle sizes examined. Therefore, we assume this latter mechanism is not accounted for in our estimate for v_*. It can be important for small particles and so we treat it next as a separate contribution.

The deposition velocity due to Brownian diffusion v_B takes account of Brownian diffusion across the laminar sub-layer from a turbulent flow. It is given by

$$v_B = .0594\ kTB/\delta_p = .0594\ u_* Sc^{-2/3} , \quad (28)$$

where kTB is the Stokes-Einstein particle diffusivity. The factor .059 is a departure from MAEROS; it is a dimensionless empirical correction factor from Brockmann et al. (1982).

We note that our treatment of turbulent deposition, including Brownian deposition from a turbulent flow, does not distinguish aerodynamically rough or smooth surfaces. In fact, the Liu and Agarwal experiments were done in aerodynamically smooth pipes. And, it is clear from the Brownian deposition formula that a laminar sub-layer is supposed to exist so this also is applicable to aerodynamically smooth surfaces. However, experiments by Wells and Chamberlain (1957) and Chamberlain (1967) show deposition is increased when surfaces are rough. We, therefore, need to consider generalizing our treatment.

Hahn et al. (1985) show the correlation of Kader and Yaglom (1977), which is applicable to deposition of submicron particles to rough surfaces, agrees with experiment over a wide range of particle Schmidt numbers. This appears to be a suitable candidate with which to generalize our equation for v_B. However, the situation for v_* appears to be less clear and requires further study.

The thermophoretic deposition velocity v_{Ts}, where s can stand for c, w or f, is estimated from

$$v_{Ts} = \frac{T - T_s}{T} \frac{9\pi\eta_g^2 r\ B\ Br}{\rho_g \delta_*} , \quad (29)$$

where Br is a dimensionless factor obtained by Brock (1962) given by

$$Br = b_k \frac{1}{1 + 3b_m Kn} \frac{1}{2 + 1/(a_g/a_p + b_t Kn)} , \quad (30)$$

where b_k, b_m and b_t are dimensionless constants (1.0, 1.37 and 1.0 respectively) and a_g and a_p are the thermal conductivities of the gas and particle material respectively.

Note the temperature gradient has been approximated as $(T - T_s)/\delta_*$ where, following Dunbar et al. (1984) and in the absence of any specific model in MAEROS, δ_* is chosen as an estimate of the distance overwhich the bulk gas temperature decreases to the surface temperature. It would clearly be better to use the thermal boundary layer thickness instead but the error introduced by using δ_* is most likely small compared to other uncertainties.

This and similar formulae for v_{Ts} have been reviewed by Derjaguin and Yalamov (1972). From comparisons with experiment they conclude that the Brock correction factor leads to underestimates of v_{Ts} for particles with Kn $\lesssim$ 1 by a factor $\simeq$ 2. A different transport mechanism applies for particles with Kn >> 1. Deraguin and Yalamov obtain an estimate for v_{Ts} in this limit which they show to agree with experiment. The formula above overestimates v_{Ts} in this limit by a factor of order Kn.

The diffusiophoretic deposition velocity v_{Ds}, where s can stand for c, w or f, is estimated from

$$v_{Ds} = D_v \frac{dc_s}{dx} \frac{1}{c_s} \frac{f_s}{f_s + (1 - f_s)\sqrt{(W_g/W_v)}} , \tag{31}$$

where dc_s/dx is the outward facing vapor concentration gradient near the ceiling, wall or floor according to whether s equals c, w or f. Comparison with Derjaguin and Yalamov (1972) shows this estimate to be applicable in the limit Kn $\gtrsim$ 1 and $f_s \ll 1$ and when the flow of vapor to or from the surface is purely diffusive. Note that we could estimate dc_s/dx in a similar way to that used to estimate the temperature gradient in the formula for v_{Ts}.

The net deposition velocity to any surface v_s is estimated by combining the contributions in the following way

$$v_c = v_B + v_* + v_{Tc} + v_{Dc} - v_G , \tag{32}$$

$$v_w = v_B + v_* + v_{Tw} + v_{Dw} , \tag{33}$$

and

$$v_f = v_B + v_* + v_{Tf} + v_{Df} + v_G . \tag{34}$$

Note that since either or both of v_{Ts} and v_{Ds} could be negative (i.e. when T_s is greater than T or dc_s/dx is negative) and v_G is subtracted in Eq. (32), some of the v_s may be negative. We therefore only apply the equations above when the result for v_s is positive and otherwise set v_s to zero.

The deposition rate to a surface per unit cell volume, λ_s, is given by

$$\lambda_s = v_s A_s / V , \tag{35}$$

where A_s is the area of surface s exposed to aerosol. R(m,t) is then given by

$$R(m,t) = \lambda_c + \lambda_w + \lambda_f + \lambda_l , \tag{36}$$

where λ_1 is the cell leakage rate, defined as the number of volume changes in the cell per second.

We have not included all possible deposition mechanisms in our treatment above. Examples left out are: Brownian diffusion in a laminar flow; deposition mechanisms associated with electrostatic charges; and inertial deposition due to non-linear mean streamlines (streamlines may be non-linear due to bends in the flow path or secondary flows induced by turbulence, which can occur in straight flow paths). This last mechanism can be important and would be an obvious next candidate for which to find a suitable treatment.

2.7 Agglomeration rates

Four agglomeration mechanisms are considered which are described in turn. We start by giving the formula for a commonly occurring factor and we describe at the end of this sub-section how the contributions for each mechanism are combined. For notational convenience the agglomeration rate formulae are given corresponding to the j^{th} and k^{th} collocation points (see Sub-section 3.1 for their definition).

The particle terminal velocity u_G is the terminal velocity due to gravity of a particle of mass m and is estimated as

$$u_G = gmB \quad . \tag{37}$$

Notice this is just v_G but we prefer to use a different symbol here to avoid confusion. As for v_G, a factor $(\rho_p-\rho_g)/\rho_p$ should be included but we ignore this small error to keep the equation identical to that used in MAEROS.

The Brownian agglomeration rate ϕ_B is estimated as

$$\phi_B(m_j,m_k) = 4\pi kT\ (B_j+B_k)\ \chi_c(r_j+r_k)\ Fu(m_j,m_k) \quad , \tag{38}$$

where B_j and B_k are the particle mobilities at the j^{th} and k^{th} collocation points respectively, χ_c is the collision shape factor which corrects the collision cross-section of particles when they depart from sphericity and Fu is a factor originally introduced by Fuchs (1964) to correct for particles small in size compared to the molecular mean free path. Fu is given by

$$1/Fu(m_j,m_k) = 1/Fu_1(m_j,m_k) + 1/Fu_2(m_j,m_k) \quad , \tag{39}$$

where Fu_1 is

$$Fu_1(m_j,m_k) = \chi_s \frac{r_j+r_k}{kT(B_j+B_k)} \left[\frac{8kT}{\pi} \left(\frac{1}{m_j} + \frac{1}{m_k} \right) \right]^{1/2} \quad . \tag{40}$$

Fu_2 is modified from unity according to Sitarski and Seinfeld (1977)

$$Fu_2(m_j,m_k) = 1 + \frac{2\sqrt{(\tilde{a}_j^2+\tilde{a}_k^2)}}{r_j+r_k} , \tag{41}$$

where $\tilde{a}$ is

$$\tilde{a} = \frac{(r+a)^3 - (r^2+a^2)^{3/2}}{3ra} - r \tag{42}$$

and

$$a = B\left[\frac{2kTm}{\pi}\right]^{1/2} \tag{43}$$

χ_s in Eq. 40 is the particle-particle sticking efficiency which is the probability that particles stick to one another when they collide. Note that it is included as a multiplicative factor in Eq. (41) only, as is done in the MAEROS model. It is not clear why it is not included in Eq. (42) also.

<u>The gravitational agglomeration rate</u> ϕ_G is estimated as

$$\phi_G(m_j,m_k) = \pi\chi_s\chi_{Fu}\chi_c^2(r_j+r_k)^2|u_G(m_j)-u_G(m_k)| . \tag{44}$$

It arises from collisions of particles travelling under the influence of gravity at different terminal velocities.

<u>The collision efficiency</u> This is a correction factor which is applied to account for the deflection of the particle stream lines from straight-lines when they approach one another. We use the correction factor derived by Fuchs (1964), χ_{Fu}, given by

$$\chi_{Fu}(m_j,m_k) = \frac{3}{2}\frac{\min(r_j,r_k)^2}{(r_j+r_k)^2} , \tag{45}$$

where *min* stands for the minimum value of r_j and r_k. r_j and r_k are just the radii of the equivalent spherical particles evaluated at the j^{th} and k^{th} collocation point respectively.

We note that the formula above is the same as that derived by Pruppacher and Klett (1978) except the factor 3/2 is replaced by 1/2. Dunbar et al. (1984) have reviewed these formulae and conclude that the Pruppacher and Klett formula, though still based on a number of approximations, has firmer foundation.

The turbulent shear agglomeration rate ϕ_{*S} is estimated from (Saffman and Turner, 1956)

$$\phi_{*S}(m_j,m_k) = \chi_s\chi_c{}^3(r_j+r_k)^3 \left[\frac{8\rho_g\pi\epsilon_*}{15\eta_g} \right]^{1/2} \tag{46}$$

It accounts for the action of turbulent shear causing particles which follow the instantaneous stream lines to collide with one another.

The turbulent inertia agglomeration rate ϕ_{*I} is estimated from (Saffman and Turner, 1956)

$$\phi_{*I}(m_j,m_k) = \chi_s\chi_c{}^2(r_j+r_k)^2 \left[\frac{512\rho_g\pi^3\epsilon_*{}^3}{15\eta_g} \right]^{1/4} |u_G(m_j)-u_G(m_k)|/g \ . \tag{47}$$

It accounts for particles colliding with one another when, due to their inertia, they are unable to follow the instantaneous stream lines.

In principle, a collision efficiency factor should be included in ϕ_{*S} and ϕ_{*I}, analogously to that included in ϕ_G. Dunbar et al. (1984) assume this factor is the same as that which appears in ϕ_G (here estimated as χ_{Fu}). However, it is by no means clear that the same factor applies since the flow field near to particles approaching one another due to gravity is not the same as that when particles approach one another due to turbulence. We leave this factor out, as in MAEROS.

The combined agglomeration rate The contributions defined above are combined as follows

$$K(m_j,m_k,t) = \phi_B(m_j,m_k) + \phi_G(m_j,m_k) + \{\phi_{*S}(m_j,m_k)^2 + \phi_{*I}(m_j,m_k)^2\}^{1/2} \ . \tag{48}$$

According to Saffman and Turner, the turbulence contributions are added quadratically. However, Dunbar et al. (1984) point out that the reasoning used by Saffman and Turner also applies to the gravitational contribution which should therefore be added to $K(m_j,m_k,t)$ in the same way. We do not do this to maintain consistency with MAEROS.

We conclude by noting that we have not included all possible agglomeration mechanisms. Examples left-out are agglomeration in laminar shear flows and body force effects (e.g. van der Waals and electrostatic forces). See Drake (1972) for a review.

3 THE NUMERICAL METHOD

3.1 Discretization of the governing equations

We discretize Eq. (1) with respect to mass using the collocation finite-element method. In this method the governing equation is required to hold at a set of collocation points only (n in total). A finite-element expansion based on values of variables at the collocation points is used. Here, this will be needed for the estimation of the agglomeration integrals.

The particular method we use is identical to that studied by the author (Wheatley, 1988) who showed that accurate solutions to the equation could be obtained with a small number of collocation points and the agglomeration kernel evaluated on the n^2 pairs of collocation points only. Consequently, the agglomeration kernel, which is in general time-dependent, can be economically recalculated as the integration of the equation advances in time.

We discretize m on a logarithmic scale as follows

$$\log_e(m_i) = \log_e(m_1) + (i-1)h, \quad i=1, \ldots n \ . \tag{49}$$

h is a constant which can be found from m_1, m_n, the smallest and largest values of discretize mass respectively, and n. We choose h to be constant (i.e. independent of i) for reasons explained in Wheatley (1988).

With this choice for m_i it is convenient to choose the mass distribution, given by $Y(m,t) = mC(m,t)$, as the dependent variable in Eq. (1), which becomes

$$\frac{\partial Y(m,t)}{\partial t} = \int_0^m K(\mu,m-\mu,t)Y(\mu,t)Y(m-\mu,t)d\log_e(\mu)$$

$$- Y(m,t)\int_0^\infty K(\mu,m,t)Y(\mu,t)d\log_e(\mu) - R(m,t)Y(m,t) + mS(m,t) \ , \tag{50}$$

where we have used $K(\mu,\nu,t) = K(\nu,\mu,t)$ and, for economy of display here and later, the integration limits are shown for μ rather than $\log_e(\mu)$.

When this equation is evaluated at the points m_i, i = 1, ...n, we can see that although the extended trapezoidal rule could be used to estimate the second integral on the right hand side the first integral will be troublesome. This is where a finite-element expansion is needed. We choose to expand the integrands as follows

$$K(\mu,\nu,t)Y(\mu,t)Y(\nu,t) \simeq \sum_{j,k} K_{jk}Y_jY_kg_j(\log_e(\mu))g_k(\log_e(\nu)) \ , \tag{51}$$

where Y_i and K_{ij} are shorthand for $Y(m_i,t)$ and $K(m_i,m_j,t)$ respectively and the i^{th} element g_i is defined in terms of a basic element g (g is also used to denote the acceleration due to gravity but no confusion should arise) by

$$g_i(x) = g((x-x_i)/h) \tag{52}$$

and x_i is $\log_e(m_i)$. The precise form of g will be discussed later but we note here that we always choose it so that $g(0) = 1$ and $g(\pm1)$, $g(\pm2)$, $g(\pm3)$, ... = 0. This guarantees that the expansion in Eq. (51) is consistent in the sense that the equation is satisfied identically when μ and ν are located at any of the collocation points.

We have chosen the particular expansion in Eq. (51) for three reasons. First, it entails the agglomeration kernel to be evaluated only at the collocation points. Second, since K and Y have not been expanded separately, the summations which result in the discretized equation are at most over two indices which cuts down on computational labor. And third, as we shall see, we obtain the same result for the second integral in Eq. (50) as that obtained by applying the trapezoidal rule. (It was shown in Wheatley (1988) that the trapezoidal rule is particularly accurate for integrals of this type.)

We now use Eq. (51) in Eq. (50) which we evaluate at the i^{th} collocation point to obtain the following closed set of equations, being the discretized form of the aerosol equation

$$\frac{\partial Y_i}{\partial t} = \sum_{j,k} P^i_{jk} K_{jk} Y_j Y_k - Y_i \sum_j D^i_j K_{ij} Y_j - R_i Y_i + m_i S_i \ , \tag{53}$$

where R_i and S_i are shorthand for $R(m_i,t)$ and $S(m_i,t)$ respectively and we have used $g_j(x_k) = \delta_{jk}$, where δ_{jk} is the Kronecker delta. The terms with P^i_{jk} and D^i_j correspond to the particle production and destruction terms respectively on the right hand side of Eq. (50). The indices in P^i_{jk} and D^i_j run from 1 to n.

D^i_j is

$$D^i_j = \int_{\varnothing}^{\infty} g_j(\log_e(\mu))d\log_e(\mu) = h \int_{-\infty}^{\infty} g(x)dx = h \ , \tag{54}$$

as desired, where we require g(x) to be chosen so that $\int_{-\infty}^{\infty} g(x)dx = 1$. (The superscript in D^i_j is clearly redundant but we retain it to maintain notational consistency with Wheatley, 1988.)

P^i_{jk} is

$$P^i_{jk} = n_{jk} \int_{\varnothing}^{m_i} g_j(\log_e(\mu)) g_k(\log_e(m_i - \mu))d\log_e(\mu) \ . \tag{55}$$

This is simplified somewhat by making the transformation $y = \log_e(\mu/m_j)/h$ to obtain

$$P^i_{jk} = h\, n_{jk} \int_{-\infty}^{\bar{j}} g(y)\, g\left(\tfrac{1}{h}\log_e(1 - e^{(y-\bar{j})/h}) + \bar{k}\right)dy \ , \qquad (56)$$

where $\bar{j} = i - j$ and $\bar{k} = i - k$. The integral must be calculated numerically and the integration range must be divided-up into sub-ranges since g is generally non-smooth - see under CHARMCOE in Appendix B for details. The indices in the coefficient P^i_{jk} in principle take all values from 1, ...n, however, only a small fraction of the coefficients are non-zero. It is straightforward to find the conditions on i, j and k for this to be so. It can be shown that they depend only on $\bar{j}$ and $\bar{k}$. The non-zero values of the coefficients are conveniently stored consecutively using an indexing based on the derived conditions.

The multiplicative factor n_{jk} is introduced as a correction to conserve mass as we now discuss.

By integrating Eq. (50) with respect to m from 0 to ∞ the following mass balance equation is obtained

$$\frac{\partial \rho}{\partial t} = -\int_0^\infty R(m,t)Y(m,t)dm + \int_0^\infty mS(m,t)dm \ . \qquad (57)$$

This equation multiplied by the volume states that the rate of increase of airborne mass equals the rate supplied by sources less the rate removed by deposition and leakage. Clearly, the agglomeration terms have cancelled as one would expect. However, this is not the case for the discretized equivalent to Eq. (57) without the factor n_{jk} in P^i_{jk}. In many applications, the removal and source terms in Eq. (50) can be small compared to the agglomeration terms. It is consequently important that the agglomeration terms cancel exactly to ensure that the removal and source terms are not swamped by cancellation errors. We therefore choose n_{jk} to achieve this.

The trapezoidal rule is used to estimate ρ (Eq. (67) in Sub-section 3.3) from which we obtain our discretized form of the mass balance equation

$$\frac{\partial \rho}{\partial t} = h\left\{\sum_{i,j,k} m_i P^i_{jk} K_{jk} Y_j Y_k - \sum_{i,j} m_i D^i_j K_{ij} Y_i Y_j - \sum_i m_i R_i Y + \sum_i m_i{}^2 S_i\right\}. \qquad (58)$$

The first two terms on the right hand side must cancel exactly, whatever the values of Y_i for i = 1, ...n. A sufficient condition is therefore to require that the sum of all coefficients of terms involving $Y_r Y_s$ for any r and s must be zero. We find that P^i_{jk} and D^i_j must be related by

$$\sum_i m_i (P^i_{rs} + P^i_{sr}) = m_r D^r_s + m_s D^s_r = h\ (m_r + m_s)\ . \qquad (59)$$

We see this amounts to $n(n+1)/2$ constraints on n_{rs} only. It is therefore convenient to choose it to be symmetric.

Six options for g are provided in CHARM. The element shown by Wheatley (1988) to give the best results when n is small is

$$\begin{aligned} g(x) &= 1 - |x| \ , \ |x| < 1 \\ g(x) &= 0 \quad\ \ , \ |x| \geq 1 \end{aligned} \qquad (60)$$

It gives rise to continuous piece-wise linear finite-element expansions. It has the particular advantage here that g is non-negative everywhere and so all the P^i_{jk} are non-negative. This is sufficient to ensure that no component of the solution of the discretized equation changes sign, as must be so on physical grounds. See Appendix D for details of the other basic elements.

3.2 The solution method

The ODE's are solved with the Fortran subroutine DEBDF written by Shampine and Watts (1979). It is based on the variable order backward differentiation method for stiff ODE's due to Hindmarsh. n + 5 coupled ODE's are solved in all; the additional five equations are integrators for the source mass, the leaked mass, and the masses deposited on the wall, floor and ceiling respectively.

The local absolute error in Y_i is constrained during the integration as follows

$$\delta Y_i < \epsilon / h\ \text{minimum}(N, \rho/m_i)\ , \qquad (61)$$

where ϵ is a relative tolerance parameter supplied by the user. This test is designed to result in estimates for N and ρ with a relative accuracy equal to or less than ϵ. It is more efficient than requiring $\delta Y_i < \epsilon Y_i$ say since it is not necessary to integrate the tails of the distribution as accurately as the bulk in order to obtain accurate estimates for the quantities of main interest.

However, the chosen tolerance criterion permits components of Y to become negative, even though the discretized equation may not admit a change of sign. We have not found this to be troublesome when using the linear finite-element but we cannot rule-out that changes of sign may cause difficulties in some cases. Substituting $\log_e Y_i$ for Y_i as the independent variable or tightening the tolerance required of δY_i in the tails may solve the trouble should it arise.

3.3 Moments of the discretized distribution

The moments we consider are ρ, the aerosol density, N, the aerosol number concentration, $m_{5\emptyset}$, the mass median mass, m_g, the geometric mean mass, and σ, the cube root of the geometric standard deviation with respect to mass. These are defined by

$$\rho = \int_{\emptyset}^{\infty} mY(m,t)d\log_e(m) \ , \tag{62}$$

$$N = \int_{\emptyset}^{\infty} Y(m,t)d\log_e(m) \ , \tag{63}$$

$$\rho/2 = \int_{\emptyset}^{m_{5\emptyset}} mY(m,t)d\log_e(m) \ , \tag{64}$$

$$N \log_e(m_g) = \int_{\emptyset}^{\infty} \log_e(m)Y(m,t)d\log_e(m) \ , \tag{65}$$

and

$$N \log_e{}^2(\sigma^3) = \int_{\emptyset}^{\infty} \log_e{}^2(m/m_g)Y(m,t)d\log_e(m) \ . \tag{66}$$

The definitions of m_g and σ are based on the number concentration distribution; the cube power in the equation for σ is conventional.

ρ times the cell volume is the total airborne aerosol mass. Similarly, N times the cell volume is the total number of airborne aerosol particles. $m_{5\emptyset}$ is sometimes called the mass median particle size. Approximately half the airborne particles have mass less than m_g. σ is a measure of the spread of the distribution. For a log-normal number distribution 68% of the particles have masses in the range m_g/σ^3 to $m_g\sigma^3$ and 68% of the airborne mass derives from particles with masses in the range $m_{5\emptyset}/\sigma^3$ to $m_{5\emptyset}\sigma^3$. Often, these relationships hold reasonably well for distributions found in practice.

We now consider the numerical estimation of these moments. Their evaluation with the extended trapezoidal rule is straightforward for all except $m_{5\emptyset}$ and illustrated only for ρ. The estimate for ρ is

$$\rho = h \sum_i m_i Y_i \ , \tag{67}$$

where Y_i is shorthand for $Y(m_i,t)$ and the summation extends over all values for which the indicated index is defined. $Y(m,t)$ has been assumed to decrease to negligible value between m_1 and m_1e^{-h} and between m_n and m_ne^{h}.

$m_{5\emptyset}$ generally lies between adjacent grid points so we estimate the integral in Eq. (64) by using a finite-element expansion for the integrand. When the linear finite-element is chosen, this is equivalent to using the extended trapezoidal

rule to estimate the contribution to the integral up to the grid point immediately below m_{50} and then using linear interpolation between the grid points either side of m_{50} to estimate the remainder.

3.4 The treatment of time-dependencies

We refer here to time-dependent input variables which we suppose are to be supplied as tables of values at discreet times. There are two aspects to this. First, how are values to be estimated at intermediate times and, second, how should these time-dependencies be handled within the model.

With regard to the first aspect, we desire the flexibility to treat both continuous and discontinuous variations. For example, the source mass release rate may change discontinuously at certain times and vary continuously otherwise. This is simply handled by linearly interpolating between data points supplied at consecutive (non-equal) times and requiring two sets of data points to be supplied at discontinuities, one set to be used for interpolation before the discontinuity, and the second to be used for interpolation after the discontinuity. It is convenient to use constant extrapolation from the extreme data points when they do not span the range of times covered in the calculational problem. Further details are given in Sub-section 4.2.

With regard to the second aspect, we anticipate that it can be too costly computationally in some cases to continuously update the time-dependent variables and those that depend on them during the integration of the governing equations, despite the efficient treatment of the agglomeration kernel by the discretization method we use. So, although the option to continuously update these variables can be provided, some alternative must be allowed for. The simplest alternative, and the one we choose, is to update these variables at discreet times and otherwise to keep them constant. Further details are given in Sub-section 4.2 - also see the discussion in Sub-section 6.4.

4 THE INPUT DATA FILE

4.1 Overview

Input data is read from tape4 (FORTRAN unit number 4) which is assigned to the file CHARMDAT on the local file area. The file is assumed to have no more than 72 columns per line.

The data is read with list directed read statements with a loop over each statement to cause reading to restart at the next line when a character other than a "," or "/" is encountered on the current line. This permits considerable flexibility over the format of the data file. The salient points are as follows:

i Lines with characters other than a "," or "/" and not counting free format numbers are ignored. A character can therefore be intentionally inserted into a line to enable it to be treated as a comment line. For example a "$" could be inserted in column 1, as is done in the example considered in Sub-section 4.4.

ii All data items are read in free format. For example, this allows the real number 1.0 to be entered as 1, 1.0 or 1.e0 and any number of spaces can separate data items which, for a single read list, can be entered on more than one concurrent line. It is important that data items are entered with an acceptable format since otherwise the current line will be treated as a comment line and reading for the current read list will recommence at the next line.

iii Items in a read list can be skipped by inserting a space followed by a comma (,) where the data item would otherwise go. The remainder of the read list can be skipped by inserting a "/". This allows default values to be assigned to variables merely by skipping over those variables when they occur in the read lists.

The reader is referred to the ANSI standard for Fortran 77 (American National Standards Institute, 1978) for detailed rules regarding what permissible formats the data may take consistent with list directed read statements. However, with the example in Sub-section 4.4 the user should find it easy to compose his or her own data files.

Some data items are checked for valid values (see Sub-section 6.5 for details) but generally this is not done. Guidance is given in the next sub-section on suitable values for all data items.

All data items must have S. I. units except molecular weights must have units kg / kmole (i.e. g / mole).

A number of input variables are treated as time-dependent. The user must enter a table of times to which corresponding values of these variables must be entered later in the file. Interpolation formulae are used in the model to estimate values of these variables at any time between zero (the start of the problem) and the problem end time. The minimum and maximum times in the table need not span the time period of the problem. Variables are extrapolated with constant values when this happens. Adjacent times in the table can be equal to enable variables to change discontinuously. If fewer values for a variable are entered than needed then missing values are copied from preceding values in the supplied table. A default value exists for the first data item of each variable.

In the next sub-section we will define the variables in each read list in turn, give acceptable ranges and, where appropriate, recommend values. Default values of all data items are given in Sub-section 4.3 and we discuss an example in Sub-section 4.4.

4.2 Detailed description

In the following, all variables in one read list are shown on a single line.

Output flags

A group of output variables is associated with each flag, as described in Section 5. The values of the variables in each group will be written on the output file OUT every q^{th} output step, where q is the value of the flag corresponding to the group. Variables independent of time are only written at the zeroth time step. No information is written when the flag is zero. The flags must be integers and are read in the following order:

cell properties	gas properties	aerosol constants
source moments	source distributions	flow properties
masses	radii	mobilities
deposition rates	agglomeration rates	
indexing	production coefficients	normalization factors
mass balances	tolerances	
moments	number distribution	mass distribution

Time step information

Output is written at intervals given by TIMESTEP(i) until the problem time equals TIMEEND(i), when output is written at TIMEEND(i) and i is increased by 1.

The last value of TIMEEND, TIMEEND(NTIME), defines the time at which the calculation stops.

NTIME must be a positive integer no greater than 20. TIMESTEP(i) and TIMEEND(i) must be positive real numbers and NTIME values each must be entered. Consecutive values of TIMEEND must increase. These variables are read in the following order:

NTIME
TIMESTEP(1), TIMEEND(1), TIMESTEP(2), TIMEEND(2), etc.

Number of columns on the output file

The output file, OUT, can be written with either a maximum of 80 or 132 columns per line. Enter an integer value on one line. Any integer not equal to 80 is interpreted to mean that the file can have up to 132 characters per line.

Times at which data for the time-dependent variables are to be provided

See the general discussion in the preceding sub-section. Values for the time-dependent variables are to be supplied corresponding to the times TIMEDATA(1) to TIMEDATA(NDATA). NDATA is the number of data points per variable and must be a positive integer no greater than 20. Consecutive values of TIMEDATA must not decrease. These variables are read in the following order:

NDATA
TIMEDATA(1), TIMEDATA(2), etc.

Frequency with which the time-dependent variables are to be revised

The quantity to be entered here (THHYSTEP) is the maximum time which is allowed to elapse since the time-dependent variables were last revised before they are revised again. THHYSTEP must be a non-negative real number. The time-dependent variables are revised continuously when THHYSTEP is zero.

Cell data

The aerosol evolves in a region called here a cell. We refer to all upward facing horizontal surfaces, vertical surfaces and downward facing horizontal surfaces within the cell as the ceiling (c), wall (w) and floor (f) respectively.

The surface areas must be non-negative real numbers and entered in the following order:

A_c A_w A_f

The surface temperatures are treated as time-dependent variables. They must be non-negative real numbers and no more than NDATA sets of values in the following order must be entered:

$T_c(1)$, $T_w(1)$, $T_f(1)$, $T_c(2)$, $T_w(2)$, $T_f(2)$, etc.

The cell volume, which is defined as the volume of free space within the cell, must be a positive real number and the leak rate, which is defined as the number of volume changes of gas in the cell per second, must be a non-negative real number. They must be entered in the following order:

V λ_1

The hydraulic diameter and the equivalent sand roughness are defined in Subsection 2.3. They must be non-negative real numbers and entered in the following order:

d_h z_s

Gas data

The gas temperature, pressure and velocity are treated as time-dependent variables. The temperature, pressure and, gas and vapor molecular weights must be positive numbers and the remaining variables must be non-negative real numbers. The data must be entered in the following order:

T(1), P(1), U(1), T(2), P(2), U(2), etc.

W_g a_g W_v
f_c f_w f_f
$\frac{dc_c}{dx}$ $\frac{dc_w}{dx}$ $\frac{dc_f}{dx}$

Boundary layer data

These data provide an option for the user to over-ride the calculation of boundary layer thicknesses in the model. This can be done by setting the value of BLFLAG to be non-zero and providing values for δ_* and δ_D. A single value only is required for δ_D which is assumed to apply irrespective of particle size. BLFLAG must be integer and δ_* and δ_D must be non-negative real numbers. Deposition by thermophoresis is ignored when δ_* is zero and deposition by Brownian diffusion is ignored when δ_D is zero. The data must be entered in the following order:

BLFLAG δ_* δ_D

Initial aerosol

The initial aerosol is assumed to be log-normal in the number concentration distribution and is parameterized by the moments σ, r_{s0} and ρ (see Sub-section 3.3 for the definition of these quantities). σ and r_{s0} must be positive real numbers and ρ must be a non-negative real number. They must be entered in the following order:

σ r_{s0} ρ

Source aerosol

The source aerosol is assumed to be log-normal in the number concentration distribution and is parameterized by the moments σ, r_{s0} and $d\rho/dt$. $d\rho/dt$ is defined to be the mass release rate of the source divided by the cell volume. The moments are treated as time-dependent variables. σ and r_{s0} must be positive real numbers and $d\rho/dt$ must be a non-negative real number. They must be entered in the following order:

$\sigma(1)$, $r_{s0}(1)$, $\rho(1)$, $\sigma(2)$, $r_{s0}(2)$, $\rho(2)$, etc.

Definition of collocation points

The collocation points can be specified in two ways. The first is to specify the number of points and the masses of the lower and upper points. Alternatively, values for e^b, m_1 and m_n can be specified. In either case, the model will calculate the intermediate points according to Eq. (49) (except m_n may be increased slightly in the second case). The second option is selected when the input value for n is set to zero.

m_1 and m_n should be chosen so that the aerosol mass and number distributions are comfortably encompassed between the two values. e^b should not be chosen so large that the distributions span only a small number of points. See Sub-sections 6.3 and 6.4 for further guidance on how to choose values for these quantities. The defaults set by the model will be reasonably satisfactory in most cases.

One of six different basic elements can be selected by assigning a value to NELEMENT. Element 2 is the linear finite-element which should be used in all normal circumstances. Element 3 may be useful when very accurate results are required but it will give unreliable results when the collocation point spacing is large. The other elements (NELEMENT = 1, 4, 5 and 6) were included as options during development of the numerical method and have since not been removed. We do not recommend using any of these alternative options.

n must be a non-negative integer and e^b, m_1 and m_n must be positive real numbers. n is set to 100 when a value greater than 100 is specified in the input or when the second option is selected and the calculated number of points exceeds 100. NELEMENT must be a positive integer no greater than 6. These variables must be entered in the following order:

n	e^b
m_1	m_n
NELEMENT	

Tolerance specifications

ϵ is a relative error parameter. It is used to set tolerances for the integration of the differential equations and set a relative tolerance for the location of zeros of functions and the values of integrals. η defines an absolute tolerance on the value of a function whose zero is sought. ς determines when bisection is used in favor of inverse quadratic interpolation to locate the zero of a function.

A reasonable range from which to select a value for ϵ is 10^{-3} to 10^{-8}. The model may not work satisfactorily when ϵ is made too large or too small. A value of 10^{-6} has so far proved to be generally satisfactory. Further guidance on choosing ϵ is given in Sub-sections 6.3 and 6.4. We would recommend that the user always use the default values for η and ς.

Up to MAXCALLS evaluations of a function can be made in an attempt to locate its zero to the desired tolerance. This may need to be increased over the default value when ϵ is made very small.

MAXTRYS determines how many integration steps are attempted by DEBDF to integrate the equations to the next specified time (the number of attempted steps is 500 times MAXTRYS). This may need to be increased over the default value when ϵ is made exceedingly small or when output is required infrequently.

MAXCALLS and MAXTRYS must be positive integers. ϵ must be a positive real number and η and ζ must be real numbers.

These variables must be entered in the following order:

ϵ	MAXCALLS	MAXTRYS
η	ζ	

Aerosol physics data

These comprise physical properties of the aerosol material and constants which appear in the models for agglomeration and deposition. χ_s, k_a, k_q, k_b and b_k must be non-negative real numbers and the remaining data items must be positive real numbers. χ_c and χ_d are unity for spherical particles and greater than unity otherwise. These data items must be entered in the following order:

ρ_p	a_p	
χ_c	χ_d	χ_s
k_a	k_q	k_b
b_k	b_m	b_t

4.3 The default data

We have described in Sub-section 4.1 how default values for data items can be assigned. This will be illustrated in the example in Sub-section 4.4. Here we define the default values.

It is convenient to do this by constructing a data file, shown below, whose effect is to assign all data items with their default values. This will make it easy for the user to decide when he or she needs to over-ride a default value. Of course, the data file which follows would not be used in practice since the default values can be assigned in their entirety merely by putting a / in column 1 of every line of the data file corresponding to a read list.

Some values require explanation. THHYSTEP is assigned an exceedingly large value (10^{10}) so that the time-dependent variables are recalculated only when a new set of interpolation formulae apply which, when NDATA is 1, will not occur. All temperatures and the pressure are standard (i.e. 20°C and 1 standard atmosphere). The value assigned to the equivalent sand roughness is that for

structural and forged steel - see Sub-section 2.3 for clarification. The gas in the bulk of the cell is taken to be pure air and the vapor adjacent to surfaces promoting diffusiophoresis is taken to be steam. The aerosol particle properties are for sodium oxide and the remaining aerosol physics data are the default values used in MAEROS.

The default input data

```
$ ****************************************
$ Input data file for the CHARM aerosol code
$ ****************************************
$
$ Output flags
$ ************
$
$   cell properties      gas properties          aerosol constants
$         1                    1                       1
$
$   source moments     source distribution       flow properties
$         1                    0                       1
$
$       masses               radii                  mobilities
$         1                    1                       0
$
$  deposition rates    agglomeration rates
$         0                    0
$
$      indexing      production coefficients    normalization factors
$         0                    0                       0
$
$    mass balances         tolerances
$         1                    0
$
$       moments        number distribution       mass distribution
$         1                    0                       0
$
$ Time step information
$ *********************
$
$ ntime
    1
$
$   timestep      timeend
     10.e0         10.e0
$
$ Number of columns on output file
$ ********************************
$
   80
$
$ Times when time-dependent data is provided
$ ******************************************
$
$ ndata
    1
$
$ times in consecutive order
$
   0.e0
$
$ Frequency with which the time-dependent variables are to be revised
$ ********************************************************************
$
1.e10
$
$ Cell data
$ *********
```

```
§
§   ceiling area          wall area          floor area
       0.e0                 0.e0               0.e0
§
§   ceiling temp.         wall temp.         floor temp.
     293.15e0             293.15e0           293.15e0/
§
§    volume            leak rate
      1.e5               0.e0
§
§ hydraulic diameter              equivalent roughness
        0.e0                             4.5e-5
§
§ Gas data
§ ********
§
§    temperature          pressure            velocity
      293.15e0          1.01325e5             0.e0/
§
§ molecular weight    thermal conductivity     vapor molecular weight
     28.96e0                .0255e0                  18.015e0
§
§ vapor mole fraction near the...
§       ceiling              wall               floor
          0.e0               0.e0               0.e0
§
§ vapor conc. gradient (kg/m**4) near the...
§       ceiling              wall               floor
          0.e0               0.e0               0.e0
§
§ Boundary layer data
§ ********************
§
§ b.l. flag           viscous b.l.          diffusion b.l.
     0                    0.e0                  0.e0
§
§ Initial aerosol
§ ***************
§
§    sigma            rad50                 density
      2.e0            .5e-6                   0.e0
§
§ Source aerosol
§ **************
§
§    sigma            rad50          density generation rate
      2.e0            .5e-6                   0.e0/
§
§ Definition of collocation points
§ ********************************
§
§   ncoll      spacing
      0         10.e0
§
§    mlower         mupper
     4.e-21          4.e-9
§
§   element number
          2
§
§ Tolerance specification
§ ***********************
§
§    eps             maxcalls             maxtrys
   1.e-6               30                    10
§
§    eta              zeta
   0.e0               .5e0
§
```

```
§ Aerosol physics data
§ ********************
§
§ particle density          particle thermal conductivity
      2.8e3                          .6375e0
§
§ collision shape factor     dynamic shape factor      sticking efficiency
         1.e0                         1.e0                      1.e0
§
§  a knudsen-weber            q knudsen-weber          b knudsen-weber
        1.37e0                       .4e0                     1.1e0
§
§     k brock                    cm brock                 ct brock
        1.e0                       1.37e0                   1.e0
§
§ ***************************
§ End of the input data file
§ ***************************
```

4.4 An example

The preceding discussion is illustrated with an example. The physical problem we consider is that of a sodium pool fire releasing sodium oxide aerosol into a containment building for 10 hours. The details of the problem are taken from Dunbar et al. (1984) and are reproduced in the table below. This is also the test problem considered in Wheatley (1988). The data file for this problem is shown at the end of this sub-section; it was used to generate the output file shown in Appendix A and is discussed in Sub-section 5.3 (We note two minor differences with the calculations presented in Wheatley (1988). First, the factor 0.0594 in Eq. (28) was not included and, second, b_k, b_m and b_t were chosen as 1.0, 1.0 and 2.48 respectively to enable like-for-like comparisons with results given in Dunbar et al. obtained from the PARDISEKO model).

Parameters for the example problem.

Aerosol composition	Sodium oxide
Particle density	2800 kgm^{-3}
Source distribution	Log-normal in C(m,t)
Source rate	2 tonnes per hour
Source σ	2 (no units)
Source r_{50}	.5 μm
Source duration	10 hours
Problem duration	34 hours
Containment volume	180000 m^3
Wall area	20000 m^2
Floor area	2800 m^2
Wall and floor temperatures	90 °C
Containment leak rate	1% by volume per day
Gas composition	Air
Gas temperature	100 °C
Pressure	100000 Pa
Turbulent energy dissipation rate	0. m^2s^{-3}

It can be seen from the file that liberal use has been made of the ability to add comment lines. This enables a title, labels and explanations to be included to clarify the file. It is recommended that this is always done.

Some of the output flags have been assigned the value 100. This is done because although the variables assigned to these flags are in general time-dependent they are constant for this particular data file. The value 100 is chosen to exceed the number of output steps and so these variables are written on the output file at the zeroth step only.

The time step information is chosen so that output is written at 5 minute intervals for the first hour and also for half an hour after the source emission rate has decreased to zero (at 10 hours). Output is otherwise written at hourly intervals.

The only dependent variable to change with time is the source emission rate which is constant except at 10 hours when it changes discontinuously. NDATA is therefore 2 and the corresponding two times are both 10 hours. THHYSTEP is assigned the default value 10^{10} so the time-dependent variables are recalculated only at 10 hours.

The cell and gas data are straightforward to follow. The comma beneath the label CEILING AREA causes the ceiling area to be assigned its default value of zero.

The values assigned to the hydraulic diameter and the equivalent sand roughness are irrelevant since the gas velocity is zero. Only one set of data values is entered for the gas temperature, pressure and velocity since these are constant throughout the calculation. The line with these values must be terminated with a "/", as shown, so that the current read list is terminated and reading continues with the next read list (the vapor mole fractions). There is no diffusiophoresis in the problem so the vapor mole fractions and concentration gradients are assigned default values of zero.

The boundary layer flag is set to unity so that values can be assigned to the boundary layer thicknesses. There is no initial aerosol so default values are assigned to the moments of the initial aerosol.

As noted previously, the mass generation rate of the source aerosol changes discontinuously at 10 hours so two values are entered for this quantity; the first value applies up to 10 hours and the second applies after 10 hours. Note that the commas in the second line of data cause $\sigma(2)$ and $r_{50}(2)$ to be assigned their values in the previous line.

The tolerance variables are assigned default values. For the aerosol physics data, the collision and dynamic shape factors are increased to 1.5 but the sticking efficiency is assigned its default value.

The example input data file

```
$ ****************************************
$ Input data file for the CHARM aerosol code
$ ****************************************
$
$ ***************** The example input data file. ***************
$ *** The sodium pool fire problem from Dunbar et al. (1984). ***
$ ***************************************************************
$
$ Output flags
$ ************
$
$   cell properties      gas properties          aerosol constants
          100                 100                      1
$
$   source moments    source distribution        flow properties
          1                   100                      100
$
$        masses              radii                  mobilities
           1                   1                       100
$
$  deposition rates   agglomeration rates
          100                 100
$
$       indexing     production coefficients    normalization factors
           1                   1                       1
$
$    mass balances         tolerances
           1                   1
$
$        moments       number distribution        mass distribution
```

```
          1                   3                         3
$
$ Time step information
$ **********************
$
$ ntime
    4
$
$    timestep      timeend
      300.e0       3600.e0
     3600.e0      36000.e0
      300.e0      37800.e0
     3600.e0     122400.e0
$
$ Number of columns on output file
$ ********************************
$
   80
$
$ Times when time-dependent data is provided
$ ******************************************
$
$ ndata
    2
$
$ times in consecutive order
$
 36000.e0  36000.e0
$
$ Frequency with which the time-dependent variables are to be revised
$ ********************************************************************
$
/
$
$ Cell data
$ *********
$
$    ceiling area          wall area         floor area
                             20.e3             2.8e3
$
$    ceiling temp.         wall temp.        floor temp.
      363.15e0              363.15e0          363.15e0/
$
$     volume             leak rate
    180000.e0       1.1574074074e-7
$
$ hydraulic diameter            equivalent roughness
/
$
$ Gas data
$ ********
$
$     temperature          pressure             velocity
        373.15e0             1.e5/
$
$ molecular weight     thermal conductivity     vapor molecular weight
/
$
$ vapor mole fraction near the...
$       ceiling               wall               floor
/
$
$ vapor conc. gradient (kg/m**4) near the...
$       ceiling               wall               floor
/
$
$ Boundary layer data
$ *******************
$
```

```
$ b.l. flag          viscous b.l.          diffusion b.l.
    1                   1.e-3                 1.e-4
$
$ lnitial aerosol
$ ***************
$
$    sigma           rad50                 density
/
$
$ Source aerosol
$ ***************
$
$    sigma           rad50          density generation rate
     2.e0            .5e-6             3.088419753e-6
       ,               ,                  0.e2/
$
$ Definition of collocation points
$ ********************************
$
$   ncoll     spacing
/
$
$    mlower       mupper
/
$
$   element number
/
$
$ Tolerance specification
$ ************************
$
$   eps           maxcalls          maxtrys
/
$
$   eta             zeta
/
$
$ Aerosol physics data
$ *********************
$
$ particle density        particle thermal conductivity
/
$
$ collision shape factor     dynamic shape factor      sticking efficiency
      1.5e0                       1.5e0/
$
$  a knudsen-weber             q knudsen-weber           b knudsen-weber
/
$
$     k brock                     cm brock                  ct brock
/
$
$ ***************************
$ End of the input data file
$ ****;***********************
```

5 OUTPUT FILES

5.1 Overview

Output is written to two files.

The first is tape5 (Fortran unit number 5) which is the terminal when the model is executed interactively. The output will comprise: a copy of the input data file (which provides an identity for the succeeding output), one line of information for each successful integration to a time when output has been requested (to enable the progress of the calculation to be monitored), and error messages. The file is assumed to have a width of no more than 80 characters and no form control characters are written.

The second is tape6 (Fortran unit number 6) which is identified with the file OUT. OUT will appear in the local file area, over writing any pre-existing file of that name. It can be sent to an output device, viewed with an editor or permanently stored for later use. The output will comprise: a copy of the input data file, one line of information (identical to that written on tape5) for each successful integration to a time when output has been requested, computational results or information, and error messages. The file is assumed to have a width no greater than 80 or 132 characters, according to the value of COLUMNS chosen by the user, and no form control characters are written.

The one line of information and the succeeding computational results are described in detail in the remainder of this section. Error messages are discussed in Sub-section 6.5.

The one line of information comprises: the output step number, the current time and a mass check. The output step number starts at zero (i.e. at time equals zero) and increases by unity each time the governing equations have been integrated to a time when output has been requested. The mass check provides a measure of how accurately the governing equations have been solved. It is a mass balance (in kg) and with perfect arithmetic should be exactly zero.

We conclude this sub-section with some general points about the computational results.

Quantities appearing on the output have S. I. units throughout except molecular weights have units kg / kmole.

The computational results or information are divided into 20 groups. The output flags set by the user in the input data file determine which groups are written, except that the first group, the mass budget, is always written. Of the

remainder, some comprise entirely time-independent quantities and are therefore only written at the zeroth output step and only if the corresponding output flag has a value different from zero. Some groups comprise entirely quantities which are (or may be, depending on the thermal-hydraulic conditions) time-dependent. If the output flag corresponding to each of these groups is non-zero then the group is written in its entirety when the output step number is a multiple of the flag. A few groups comprise a mixture of time-independent and time-dependent quantities. Generally, the time-independent quantities are written at the zeroth step only otherwise the groups are treated as though they comprise entirely time-dependent quantities.

This scheme provides the user with considerable flexibility over what information is written without the necessity of always writing exceedingly large output files.

5.2 Definition of output groups and variables

In this sub-section, we describe what information is written for each group in order of appearance on the output file. The titles in brackets in what follows are those which appear on the output file corresponding to the defined variables.

The mass budget

This group comprises the airborne aerosol mass and the accumulated mass (since time equals zero) on floor, wall and ceiling, leaked and released from the source.

Airborne aerosol moments

This group comprises σ (SIGMA), r_{50} (RAD50), ρ (MDENSITY), N (NDENSITY), m_g (GEOMMEAN), and m_{50} (MASS50) for the airborne aerosol.

Source moments

This group comprises σ (SIGMA), r_{50} (RAD50), $d\rho/dt$ (MDENSITY), dN/dt (NDENSITY), m_g (GEOMMEAN), and m_{50} (MASS50) for the source.

The airborne mass distribution

This group comprises $m_i Y_i$ for $i = 1, \ldots n$.

The airborne number distribution

This group comprises Y_i for i = 1, ...n.

The source mass and number distributions

This group comprises $m_i^2 S_i$ and $m_i S_i$ for i = 1, ...n.

Collocation information

This group comprises the identification number of the chosen basic finite-element (NELEMENT), the half-width of this element (HWIDTH), n (NCOLL), m_{i+1}/m_i i.e. e^h (SPACING), and m_n/m_1 (RANGE).

Indexing for the production coefficients

This group comprises the information which enables all values of i, j and k to be found for which P^i_{jk} is non-zero. The value of P^i_{jk} depends only on the values of $\bar{j} = i - j$ (JBAR) and $\bar{k} = i - k$ (KBAR). The following quantities are written for each permissible value of $\bar{j}$: the minimum and maximum values of $\bar{k}$ for which P^i_{jk} is non-zero (KBARMIN and KBARMAX), the number of these values (NKBAR), and an index (INDEX) which is used to locate the non-zero values of P^i_{jk} stored consecutively in a one-dimensional array.

The production coefficients

This group comprises all non-zero values of P^i_{jk}. INDEX($\bar{j}$)+$\bar{k}$ is the position of P^i_{jk} in the list; $\bar{j}$ and $\bar{k}$ must lie in valid ranges which can be determined from the previous output group.

The normalization factors

This group comprises all values of n_{jk}. They are written in the following order: n_{11}, n_{12}, n_{13}, ..., n_{21}, n_{22}, etc.

The collocation points

This group comprises m_i for i = 1, ...n.

Particle radii

This group comprises r_i for i = 1, ...n.

Tolerance information

This group comprises ϵ (EPS), η (ETA), ζ (ZETA), the maximum number of function calls CO5WHE can make in order to find one zero (MAXCALLS), and the maximum number of times DEBDF will be called to integrate the governing equations to the next requested time (MAXTRYS).

Aerosol physics data

This group comprises χ_c (CSHPFCTR), χ_d (DSHPFCTR), χ_s (STICKEFF), k_a (AKNUDWEB), k_q (QKNUDWEB), k_b (BKNUDWEB), ρ_p (PDENSITY), α_p (PTHRMCON), b_k (KBROCK), b_m (CMBROCK) and b_t (CTBROCK).

Cell data

This group comprises V (VOLUME), λ_1 (LEAKRATE), d_h (HYDRDIAM), z_e (EQVROUGH), A_c (AREACLNG), A_w (AREAWALL), A_f (AREAFLOR), T_c (TEMPCLNG), T_w (TEMPWALL) and T_f (TEMPFLOR).

Gas data

This group comprises T (TEMP), P (PRESS), U (VELOCITY), ρ_g (GDENSITY), η_g (DYNVISC), $\bar{l}$ (MNFRPATH), W_g (MOLWT), α_g (GTHRMCON), W_v (MOLWTV), D_v (DIFFUSV), f_c (VMFRCLNG), f_w (VMFRWALL), f_f (VMFRFLOR), dc_c/dx (VCGRCLNG), dc_w/dx (VCGRWALL), dc_f/dx (VCGRFLOR), c_c (VCONCLNG), c_w (VCONWALL) and c_f (VCONFLOR).

Flow data

This group comprises ϵ_* (EDDYDISS), u_* (USTAR), δ_* (VBLTHICK) and δ_{Bi} for i = 1 ...n (DBLTHICK).

Particle mobilities

This group comprises B_i for i = 1, ...n (MOBILITY).

Deposition rates

This group comprises λ_{ci} for i = 1, ...n, λ_{wi} for i = 1, ...n and λ_{fi} for i = 1, ...n.

The agglomeration kernel

This group comprises all values of K_{jk}. They are written in the following order: K_{11}, K_{12}, K_{13}, ..., K_{21}, K_{22}, etc.

5.3 An example

An example output file, OUT, is shown in Appendix A. It was obtained from the example input data file considered in Sub-section 4.4. In addition to illustrating the previous discussion we have included it in full to assist users in the validation of their implementations of CHARM. We expect it to be straightforward to follow so we only make a few points of clarification here.

Since all the output flags are non-zero, all input data items, except the data for the time-dependent external variables, are shown at the zeroth step on the output file, whether assigned values in the input data file or not. This enables all the parameters of the problem, including all model constants, to be verified.

The collocation information was derived entirely from default values. This will be adequate in most cases, however, we would recommend repeating the calculation with a larger values of n in order to check that the results have converged sufficiently well in this parameter. See Sub-section 6.4.

It can be seen that all the normalization factors except n_{nn} are close to unity. This provides confidence in the calculations resulting in the production coefficients. n_{nn} is significantly different from unity because there are no collocation points beyond m_n.

At output steps 1 and beyond one can see that the mass balance (MASS CHECK) is never larger than the order of 4×10^{-5} kg which is satisfactorily small in relation to the total mass of aerosol released by the source, 2×10^{4} kg. This is a good indication that ϵ is suitably small (10^{-6}). Nevertheless, we would recommend repeating the calculation with a different value of ϵ, say 10^{-4} or 10^{-8}, in order to check that the results have converged sufficiently well as a function of this parameter. See Sub-section 6.4.

For the first hour and also for half an hour beyond the termination of the source, output is written every 5 minutes so that detail is not lost during these rapid transients. The remaining output is written at hourly intervals.

One can also see that the aerosol mass and number distributions have only been written every third output step. This can amount to a considerable saving of paper when n is large! It can be seen from the distributions that Y_1 and Y_n always remain small compared to N, and m_1Y_1 and m_nY_n always remain small compared to ρ. This indicates that m_1 and m_n have been chosen suitably small and large respectively.

6 EXECUTION

6.1 Machine attributes

The information given here may be helpful to those who wish to implement the model on a non-Cray computer or under an operating system different from CTSS (the Cray Time-Sharing System).

CHARM was developed and tested entirely on Cray-1 and Cray-XMP computers. The operating system used was CTSS, developed at the Los Alamos National Laboratories, USA.

The standard word length is 64 bits, which is used for integers and single precision real numbers. Double precision real numbers use 128 bits.

Of particular importance are the exponent range and the number of significant digits of single precision real numbers. They have 15 exponent bits which give a range 10^{-2466} to 10^{2466} and 48 mantissa bits which gives just over 14 significant decimal digits.

We have used the Cray Fortran compiler version 1.14f with code optimization and vectorization. However, with the exception that up to 8 characters are used for symbol names to enhance readability, the code is written in standard ANSI Fortran 77 and no vectorization specific subroutines or functions are used.

Three CLAMS (Common Los Alamos Mathematics Software) library subroutines are called which are not supplied with the source. Sufficient guidance is given in the documentation here to enable alternatives to be implemented without too much difficulty should this be necessary.

6.2 On-line execution

We will assume here and in Sub-section 6.3 that the user is using a Cray computer with CTSS.

The user is supplied with a HISTORIAN PLUS source file called CHARMHIS (this is done so that modifications can be made to the model with a minimum of difficulty - an introduction to HISTORIAN PLUS is given in Section 7). An executable image can be made from CHARMHIS with the following commands:

```
historn(i=charmhis,c=charmcft)
cft i=charmcft,b=charmldr,maxblock=2560
ldr i=charmldr,x=charm
```

The HISTORIAN PLUS command, HISTORN, causes the HISTORIAN PLUS source file to be translated into a compiler source file CHARMCFT. Next, the Cray Fortran command, CFT, causes CHARMCFT to be translated into a binary relocatable file, CHARMLDR (the MAXBLOCK option sets the maximum code block size for optimization and vectorization). Lastly, the executable image, CHARM, is created from CHARMLDR with the loader command, LDR.

CHARM can be executed by typing the following:

charm

The default cpu time (1.0) and priority (1.0) are assumed. Alternatively, the following could be entered:

charm / t p

where t is the cpu time allocation in minutes and p is a priority with value in the range 0.001 to 5.0.

The data file CHARMDAT must be present on the local file area before the image is executed.

Output will appear on the terminal and the file OUT will be created. OUT can be printed, viewed or stored after execution is complete.

6.3 Machine resources

We are concerned here with memory requirements and computing time. We emphasize that any particular values we give are specific to the compiler and machine we use. However, the general aspects of the discussion may be more widely useful.

The executable code requires just over 76k decimal words of memory (1 word = 8 bytes). The space needed for variables is dominated by just three arrays which require just over 32k words of memory. Savings can be made by inhibiting vectorization and optimization (at the expense of cpu time) and by reducing the maximum allowed value of n (presently 100) and reducing array dimensions correspondingly.

The cpu time required by the model depends on a number of factors. In general, however, the cpu time will be dominated by one or both the time taken to solve the ODE's and the time taken to update the time-dependent variables.

Clearly, important factors are the problem duration and the stiffness of the ODE's to be solved. High or very transient source mass release rates tend to

cause stiffness. Often, however, these aspects are beyond the control of the user.

Important factors within the control of the user are the accuracy specified for the integration of the ODE's (i.e. the size of ϵ), the number of collocation points, and the precise way the time-dependent variables are to be handled.

The cpu time will increase with decreasing ϵ but the strength of the dependence is not immediately apparent. In calculations in which the cpu time was dominated by the integration of the ODE's we have found only modest increases in cpu time e.g. < 40% when ϵ was reduced from 10^{-6} to 10^{-8}. This is believed to be due to the use of a high order integration method which is thereby capable of giving significant improvements in accuracy with modest reductions in the integration time step.

The cpu time can be very strongly dependent on the specified number of collocation points. Again, in calculations in which the cpu time was dominated by the integration of the ODE's, we have found the cpu time to depend on n to a power midway between 2 and 3. This is largely due to the need of the ODE solver to solve an $n^{\times}n$ system of linear equations for each integration step. We would recommend that a calculation should first be done with n equal to 10 or so to obtain a base line for the cpu time and to check that the problem has been properly formulated before doing a calculation with a much larger value of n.

The cpu time can be dominated by the time taken to update the time-dependent variables when they are updated at frequent intervals. An example which illustrates this was considered in detail in Wheatley (1988). In some cases it may be advantageous to update the time-dependent variables continuously (i.e. set THHYSTEP equal to zero) rather than have them updated at small time intervals since there is an overhead associated with each time the ODE solver is reset (see the description of CHARM in Appendix B).

Clearly there is a trade-off between accuracy of the results and cpu time which is dictated by the values chosen for ϵ, n and THHYSTEP. Optimum values for these quantities depend on the needs of individual users and may need careful consideration. See the next sub-section also.

6.4 Accuracy

There are three parameters which strongly affect the accuracy of the results. These are ϵ, e^b (which is related to n) and THHYSTEP. We assume that m_1 and m_2 are always chosen so that they do not compromise the accuracy (see Sub-section 4.2). The precise way these quantities are related to the accuracy of the

results does not concern us here, instead we consider how the accuracy can be established, which we recommend always should be done.

First, one should decide what the important output quantities are. Often, these are the variables in the mass balance output group which can usually be obtained to given accuracy with less demand on computing resources than the airborne aerosol moments or the airborne aerosol distribution, for example.

Second, one should decide what level of accuracy is required. Often, input data and modelling uncertainties do not warrant high accuracy; 2 or 3 significant figures are usually quite satisfactory. However, higher accuracy may be needed to be sure the calculation is numerically stable and has converged.

The number of accurate significant digits can be established by doing repeat calculations in turn with ϵ reduced by a factor of 100, e^h increased by a factor of 2 and (if appropriate) THHYSTEP reduced by a factor of two and in each case establishing how many significant digits in the results remain unchanged when compared with the base calculation. Note that there is little point in choosing THHYSTEP much smaller than the minimum time period between consecutive non-equal values of TIMEDATA.

6.5 Error messages

Error messages are written on tapes (Fortran unit numbers) 5 and 6. Either they arise from fatal errors, in which case execution of the model stops, or they are warnings arising from errors which may not be serious. The messages on tape 5 may be accompanied by System generated error messages when the model is run under CTSS.

Whenever any of these messages occurs a golden rule is always to first check the input data file for errors. However, should this not prove to cure the problem, in the following we give explanations of what each of the error messages mean and suggest likely causes.

We assume that the user has not modified CHARM. If this is not so and the trouble was not solved with the hints below then the user is advised to carefully check his or her code modifications.

*** CHARMIN fails: NTIME is le 0 or gt than 20 ***

Fatal. NTIME is limited by an array dimension to be positive and no greater than 20. Check the input data file!

*** CHARMIN fails: NDATA is le 0 or gt 20 ***

Fatal. NDATA is limited by array dimensions to be positive and no greater than 20. Check the input data file!

*** CHARMIN warning: end-of-file read - could be an error in the data ***

Non-fatal. If the input data file has been read correctly and is of the right length then CHARMIN should not go on to read the end-of-file marker. Check the input data file!

*** CHARMIND fails: NONZERO is XXX ***

Fatal. The number of non-zero production coefficients is limited by an array dimension to 300. The error occurs when the limit is exceeded. This may arise when a large number of collocation points have been specified and/or one of the higher order elements has been chosen. Check the input data file or increase the dimension of the array PIJK in the common block COEF.

*** CHARMCOE fails: YLOWER is ge YUPPER ***

Fatal. YLOWER and YUPPER are the lower and upper limits of integration in the integral for the production coefficients. They are chosen so that the integrand is non-zero everywhere between the limits. The indexing calculated in CHARMIND and the logic at the start of CHARMCOE should guarantee that this error never occurs. Non-occurrence of this message provides confidence that this part of the calculation has been done correctly.

*** CHARMCOE warning: IERROR is -1 ***

Non-fatal. IERROR is an error flag generated by GAUS8. The value -1 indicates that the lower and upper limits of the integration range are too close to enable the requested tolerance for the integral to be met. This may arise if the mass range of the collocation points is exceedingly large but should not normally occur. Check the input data file! The consequent error in the production coefficient is unlikely to be significant in the subsequent calculations.

*** CHARMCOE warning: IERROR is 2 ***

Non-fatal. IERROR is an error flag generated by GAUS8. The value 2 indicates that GAUS8 was unable to calculate the integral to the requested tolerance. Experience has shown this is usually due to the presence of rounding error in the integrand. This may arise if the mass range for the collocation points is exceedingly large but should not normally occur. Check the input data file! The consequent error in the production coefficient is unlikely to be significant in the subsequent calculations.

*** CHARMFLO fails: IFAIL is 1 ***

Fatal. IFAIL is an error flag generated by CO5WHE. The value 1 means that the function CHARMFAN, for which the location of a zero has been requested, has the same sign at the lower and upper limits of the search range. This should not occur when reasonable values for the equivalent sand roughness, the hydraulic diameter and the flow speed have been chosen. Check the input data file!

*** CHARMFLO fails: IFAIL is 2 ***

Fatal. IFAIL is an error flag generated by CO5WHE. The value 2 means that CO5WHE was unable to locate the zero of CHARMFAN to the specified tolerance after MAXCALLS evaluations of it. Check the tolerance specifications in the input data file!

*** CHARMDIF fails: LRWORK is too small ***

Fatal. LRWORK is the dimension of a real work array required by CHARMDIF. This error will not occur unless the user has modified the code to allow the maximum number of collocation points to exceed 100 and failed to increase the dimension of the array RWORK accordingly.

*** CHARMDIF fails: LIWORK is too small ***

Fatal. LIWORK is the dimension of an integer work array required by CHARMDIF. This error will not occur unless the user has modified the code to allow the maximum number of collocation points to exceed 100 and failed to increase the dimension of the array IWORK accordingly.

*** CHARMDIF fails: IDID is -1 ***

Fatal. IDID is an error flag generated by DEBDF. The value -1 indicates that MAXTRYS times 500 integration steps have been attempted without succeeding in integrating to the time requested of DEBDF. Check that the requested output times are at reasonable intervals. Check the tolerance parameter EPS; it should not be too small. Check the initial and source aerosol distributions, especially the airborne density and the source emission rate. If all these seem reasonable then increase MAXTRYS.

*** CHARMDIF fails: IDID is -2 ***

Fatal. IDID is an error flag generated by DEBDF. The value -2 indicates that the error tolerances requested of DEBDF are too stringent. Check the value of EPS specified in the input data file.

*** CHARMDIF fails: IDID is -3 ***

Fatal. IDID is an error flag generated by DEBDF. The value -3 indicates that the computed solution has a zero component with a zero component in the absolute error test for that component. This is unlikely to arise but may do so when no airborne and source aerosols exist. Check the input data file!

*** CHARMDIF fails: IDID is -6 ***

Fatal. IDID is an error flag generated by DEBDF. The value -6 indicates that DEBDF had repeated convergence test failures on the last attempted step. Check that the problem as specified in the input data file is physically realistic.

*** CHARMDIF fails: IDID is -7 ***

Fatal. IDID is an error flag generated by DEBDF. The value -7 indicates that DEBDF had repeated error test failures on the last attempted step. Check that the problem as specified in the input data file is physically realistic.

*** CHARMDIF fails: IDID is -33 ***

Fatal. IDID is an error flag generated by DEBDF. The value -33 indicates that DEBDF encountered trouble from which it cannot recover. DEBDF will print a message on tape5 explaining the trouble. It is usually caused by invalid input to DEBDF. This error is not expected to occur unless CHARM has been modified by the user.

*** CHARMMOM fails: IFAIL is 1 ***

Fatal. IFAIL is an error flag generated by CO5WHE. The value 1 means that the function CHARMM5O, for which the location of a zero has been requested, has the same sign at the lower and upper limits of the search range. This should not normally occur. Check the input data file! The source and initial aerosols may not be physically realistic or the collocation points may have been badly chosen.

*** CHARMMOM fails: IFAIL is 2 ***

Fatal. IFAIL is an error flag generated by CO5WHE. The value 1 means that CO5WHE was unable to locate the zero of CHARMM5O to the specified tolerance after MAXCALLS evaluations of it. Check the tolerance specifications in the input data file!

*** CHARMOUT warning: the mass fraction in the top bin is XXX ***

Non-fatal. This is written when greater than 10% of the airborne aerosol mass is associated with the last collocation point. Check the collocation point specifications and the problem specification in the input data file.

*** CHARMOUT warning: the mass fraction in the bottom bin is XXX ***

Non-fatal. This is printed when greater than 10% of the airborne aerosol mass is associated with the first collocation point. Check the collocation point specifications and the problem specification in the input data file.

*** CHARMOUT warning: the number fraction in the top bin is XXX ***

Non-fatal. This is printed when greater than 10% of the airborne particles are associated with the last collocation point. Check the collocation point specifications and the problem specification in the input data file.

*** CHARMOUT warning: the number fraction in the bottom bin is XXX ***

Non-fatal. This is printed when greater than 10% of the airborne particles are associated with the first collocation point. Check the collocation point specifications and the problem specification in the input data file.

7 CODE MODIFICATIONS

7.1 Scope

Although CHARM can be applied to a wide range of aerosol problems without modification, we expect situations will arise when the user will find it desirable to make relatively minor modifications to the model to suit his or her specific purposes. The sort of modifications we have in mind are: changes to the physical models for gas and flow properties and agglomeration and deposition rates; adding an output file to interface with graph plotting routines; adding variables to the existing input and output subroutines; replacing the library subroutines GAUS8, SSORT and DEBDF with alternatives; and adding new variables to those treated as time-dependent. Such modifications can be done with little difficulty. We outline here a general approach which we recommend should always be followed and give two examples. The treatment of multicomponent aerosols, condensation and evaporation would require non-trivial extensions to CHARM and considerable development work should an efficient treatment be desired. This is discussed in Section 9.

7.2 General method

The general method we advocate is to use a software package such as HISTORIAN PLUS (OPCODE Inc., 1985) or UPDATE (CRAY Research Inc. 1984) which will automatically execute and keep track of changes made to CHARM. Thereby, changes for a single purpose can be kept in a single compact file, revoked at any time to restore the original model and added to others with ease.

You will see from the compiler source listing in Appendix D that each line of code has a label in columns 73-80. These were generated by HISTORIAN PLUS (but are equally compatible with UPDATE). To modify the code, a modification file is made which comprises a sequence of commands (subsequently called directives to conform with HISTORIAN PLUS nomenclature) referring to these labels. HISTORIAN PLUS is then used to execute the directives in conjunction with the HISTORIAN PLUS source file to generate a new compiler source.

A basic list of HISTORIAN PLUS directives is as follows (the corresponding UPDATE directives are similar):

*/	Anything following this on the same line is a comment.
*insert "label"	The lines of Fortran following this directive are inserted after the line in the source labelled with "label".
*before "label"	The lines of Fortran following this directive are inserted before the line in the source labelled with "label".
*delete "label"	The line in the source with label "label" is deleted and replaced with the lines of Fortran following this directive.

*ident "name" This is required at the start of the modification file. "name" is used to generate labels for the Fortran which is inserted according to the succeeding directives.

*call "name" This can appear anywhere in the lines of Fortran which follow the *insert, before* and *delete* directives and causes the block of Fortran in the source with labels generated from "name" to be inserted at this point. The block of Fortran must have been identified originally as suitable for copying in this way (i.e. it must have been defined as a "common deck"). All common blocks in the source were generated from "common decks".

*read "fname" The sequence of directives in the file "fname" are read.

Details of these and other directives can be found in the reference manual for HISTORIAN PLUS but it is hoped that the above outline and the examples which follow will serve to illustrate the utility of the approach.

7.3 Two examples

The first example shows how to obtain values of the dimensionless relaxation time on the output file. This may be desired to determine whether the correlation for turbulent deposition is being used within its range of validity. The HISTORIAN PLUS modification file, USERMOD1, is:

```
*/
*/ Modifications to CHARM to output the dimensionless relaxation time.
*/
*ident relax
*insert agglom.3
     +                  ,dimrel(100)
*insert chmdep.77
      dimrel(icoll)=dimrelax
*insert chmout.339
      if(columns.eq. 80)write(ntape6,9056)
      if(columns.eq.132)write(ntape6,8056)
      if(columns.eq. 80)write(ntape6,9003)(i,dimrel(i),i=1,ncoll)
      if(columns.eq.132)write(ntape6,8003)(i,dimrel(i),i=1,ncoll)
*insert chmout.518
9056  format('Dimensionless relaxation times...',/
     +5(4x,'   dimrelax '))
*insert chmout.539
8056  format('Dimensionless relaxation times...',/
     +8(4x,'   dimrelax '))
```

The first insert directive causes a new array DIMREL to be included in the common block labelled AGGLOM which appears in the subroutines CHARMDEP and CHARMOUT. The succeeding inserts should be relatively straightforward to follow in conjunction with the source listing.

This modification file can be implemented with the following HISTORIAN PLUS commands:

```
historian(i=charmhis,n=charmopl)
historian(i=usermod1,p=charmopl,c=charmcft,f)
```

CHARMHIS is the HISTORIAN PLUS file. The first command creates a binary old-program-library which contains the information HISTORIAN PLUS needs for subsequent modification runs. The second command is a modification run which generates the file CHARMCFT which can be read by a Fortran compiler. f at the end of this command indicates that nothing must be left-out of the CHARMCFT file.

The second example shows how the code can be modified to make the leakage rate a time-dependent variable. The modification file, USERMOD2, is:

```
*/
*/ Modifications to CHARM to enable the leak rate to depend on time.
*/
*ident leakrate
*insert thrmhydr.10
     +                  ,lkrta0(20),lkrta1(20),lkrtdata(20)
*insert thrmhydr.11
     +                  ,lkrta0,lkrta1,lkrtdata
*delete chmin.119
14    read(ntape4,*,err=14,end=100)volume
34    read(ntape4,*,err=34,end=100)(lkrtdata(idata),idata=1,ndata)
*insert chmblo.81
      data lkrtdata/0.e0,19*-1.e0/
*insert chmith.56
      call charmwth(lkrta0,lkrta1,lkrtdata)
*insert chmuth.25
      leakrate = lkrta0(ithhy) + lkrta1(ithhy) * timemean
*delete chmout.267
*insert chmout.272
      write(ntape6,9030)volume,leakrate
```

The first two insert directives add three real arrays, a data array and two interpolation formulae coefficient arrays, to the THRMHYDR common block. The

delete directive which follows causes VOLUME to be read on one line of the data file and the leak rate data to be read on subsequent lines. The next three insert directives result in, in turn, the leak rate data array to be initialized, the coefficients in the interpolation formulae for the leak rate to be calculated from the data and the leak rate to be calculated from the interpolation formulae. The last two directives allow the leak rate to be written on the output file more than once.

This modification file can be implemented in the same way as the first example except that USERMOD1 must be replaced by USERMOD2 in the arguments of the HISTORIAN PLUS commands. Alternatively, the two files USERMOD1 and USERMOD2 can be implemented in conjunction by defining a third file called USERMODS which reads:

```
*read usermod1
*read usermod2
```

USERMOD1 must now be replaced by USERMODS in the arguments of the HISTORIAN PLUS commands.

8 CHARM AS A SUB-MODEL OF A LARGER MODEL

We discuss here some general aspects of how CHARM could be implemented as a sub-model of a larger model. Though some modifications to CHARM would be needed, most of the subroutines would not be affected as a consequence of the modular design of the code. The details, of course, will depend on the particular application.

In general we envisage the subsuming model to be one which calculates at least velocities, pressures and temperatures of the flow and surface temperatures, in each cell of a multicell problem, and convection and possibly mixing of aerosol from cell to cell (the cells could be Eulerian or Lagrangian). In some applications the gas composition may not be constant and we suppose this would also be calculated for the bulk gas and adjacent to walls when a component of the gas is condensing or evaporating there.

The role of CHARM in this general context would be to calculate aerosol behavior within each cell, much as is does now for one cell only. This has to be done explicitly, by which we mean the aerosol equations are solved separately from the equations governing the thermal-hydraulics and aerosol transport during each time step. The problem is therefore seen as two-fold: time step control; and arranging the data interface between CHARM and the subsuming model and input and output.

Without condensation or evaporation of the aerosol, aerosol behavior within a cell will not affect the thermal-hydraulics to any significant extent except possibly in strong thermal radiation fields (of course, the converse is not true). This means the maximum allowable thermal-hydraulic time step need not be reduced as a consequence of the presence of aerosol. Some limit may need to be applied, however, when the thermal-hydraulic conditions change considerably over one time step. There is two way coupling between aerosol behavior within a cell and cell to cell transport of aerosol which could in some cases call for the maximum allowable aerosol transport time step to be reduced. There are no internal constraints on the CHARM time step which is simply chosen to match the subsuming model time step.

Two way coupling can exist between intra-cell aerosol processes and the thermal-hydraulics when condensation and evaporation of the aerosol can occur (Clement, 1984). This can entail severe time step limitations or else require that the split between the thermal-hydraulic calculations and the intra-cell aerosol calculations is modified. Since CHARM in its present version does not treat condensation and evaporation, this need not concern us further here.

We assume the subsuming model is able to supply CHARM with thermal-hydraulic information and an aerosol distribution, derived from the aerosol transport calculations, for each cell corresponding to an instant of time in the problem. We assume also that this data is not conveniently supplied for two instants of problem time simultaneously. This eliminates the possibility of interpolating data in time during the calculations done by CHARM. The subroutines CHARMITH, CHARMWTH, CHARMUTH, CHARMSLN and CHARMZLN are therefore not needed (we suppose the source distribution is to be supplied by the subsuming model) and, during one time step, the agglomeration kernel, deposition and source rates need be calculated only once per cell. The ODE solver in CHARM has to be reinitialized at the start of each time step for each cell. All this means the main program, CHARM, can be considerably simplified.

All the data which depends on cell location and/or time has to be supplied to CHARM at the start of each time step for each cell. This data includes: the thermal-hydraulic data, the cell geometry and associated data such as the surface roughness and leak rate, the aerosol distribution, the source distribution, and the gas composition when it is time-dependent. The aerosol distribution has to be resupplied to the subsuming model at the end of the time step. Small negative components (see Sub-section 3.2) could be set to zero and the distribution renormalized to exactly conserve mass. Deposited and leaked masses and moments of the distribution could also be supplied when these are not to be output by CHARM.

The cell and time-independent data includes: the collocation information and quantities derived from this such as P^i_{jk} and n_{jk}, the aerosol physics data and the tolerance data. Input data relevant to this category can be read by CHARM and this category of data can be stored within CHARM. CHARMIN and CHARMBLO can therefore be considerably simplified. The subroutines which calculate the derived data need be called once only.

As noted in Sub-section 6.3, just three arrays make up nearly one half the size of the executable code. It may be important with some machines and depending on the size of the subsuming model to make sure these arrays do not take up more space than needed. This can be done by making the dimensions of these arrays dynamic and having the subsuming model provide space for them.

9 EXTENSIONS TO CHARM

9.1 Multicomponent aerosols

We discuss what further work needs to be done to enable CHARM to efficiently treat multicomponent aerosols.

When the aerosol has variable composition and is composed of s components and agglomeration and deposition for particles of given mass can be adequately characterized by their average composition then the aerosol behavior can be characterized by a system of s coupled integro-differential equations (Simons, 1982). The discretization method described in Section 3 for a single component aerosol is simply generalized to discretize this new system of equations. A system of $n \times s$ coupled ODE's results. Directly solving this system of equations can pose a problem when the equations are stiff and s is large since the solution time of stiff ODE solvers is approximately proportional to $(n \times s)^3$. For example, Grimley et al. (1988) consider 40 or so components for which the computing time would be increased by a factor of order 10^4 over that for a one component aerosol.

An alternative to the direct method of solution is, by exploiting the special structure of the Jacobian, to reduce the equations to a set of n coupled ODE's to be solved first and s-1 sets of n coupled linear ODE's whose solution depends on the solution obtained to the first set of ODE's. An example of this approach was considered by Stock et al. (1987) who compared the two methods. However, they considered a two component aerosol only and the discretization method was not the same for the two solution methods. It is therefore unclear from their results which solution method would be most efficient in general.

So, further assessment of the two methods is needed to determine which is optimum. The modifications required of CHARM to treat multicomponent aerosols using either of the two solution methods are believed to be straightforward.

9.2 Condensation and evaporation - one species

We now discuss how CHARM could be extended to treat condensation and evaporation of one component onto and from the aerosol.

Extra terms need to be added to the aerosol equations to treat condensation and evaporation. These terms cause the equations to become exceedingly stiff. Considerable progress with this problem has been made recently by Gelbard (1987) who treats an aerosol with one volatile component. His method is to split the problem into two parts and treat each part in separate numerical steps. The first part deals with condensation and evaporation only and is based on the

method of characteristics. Particles of different size are coupled only through mass and energy balances within the volume. The second part deals with agglomeration, deposition and leakage only, as considered in this manual.

It would seem essential to split the problem in this way if a collocation method is used to treat the agglomeration terms since evaporation and condensation introduce into Eq. (1) a term of the form $\frac{\partial}{\partial m} C(m,t)E(m,t)$, where $E(m,t)$ is the particle growth law. This term is ill-defined at the grid points in the collocation method A finite difference approximation could of course be attempted for these terms but it seems likely that it would lead to poor results when e^h is large in comparison with unity. In any case, many fixed grid methods introduce artificial spreading of the aerosol distribution when the aerosol is evolving dominantly by condensation and evaporation (Tsang and Brock, 1983) which, as Gelbard points out, is largely overcome with his method of splitting.

9.3 Condensation and evaporation - many species

However, further work is required to treat aerosols with more than one volatile component, particularly when chemical reactions in the vapor phase must be taken into account. The problem here is the computational cost involved in maintaining chemical equilibrium in the vapor phase during the condensation and evaporation step.

10 CONCLUSIONS

We have described the models in CHARM and given detailed information to enable others to use and modify the code.

We have tried to ensure the code is "bug" free, both by design of the code architecture, transparency of the FORTRAN (we do not claim always to have succeeded in this respect!), and by testing. However, we cannot guarantee that it is free of bugs. Should you find bugs or encounter difficulties associated with the way the code operates, please communicate your findings to the author so that other users may enjoy the benefits.

We would also be happy to receive details of applications or developments you make of CHARM. Indeed, we hope Section 2 makes clear that there is considerable scope for developing the models for the gas and flow properties, and agglomeration and deposition.

Despite the lengthy details about CHARM included here, we do not wish to leave the impression that the code is difficult to use or modify. On the contrary, we have gone to some length in the design of the code to ensure the opposite. Try it!

We believe that the code can form the basis of extensions to treat multicomponent aerosols, condensation and evaporation as was discussed in section 9. However, further development and testing of methods is needed should a computationally efficient model be desired.

REFERENCES

AMERICAN NATIONAL STANDARDS INSTITUTE, ANS X3.9-1978, 1978.

BROCK, J. R., *J. Colloid Interface Sci.*, 17 (1962), 768.

BROCKMANN, J., McMURRY, P. H. AND LIU, B. Y. H., *J. Colloid Interface Sci.*, 88 (1982), 522.

CHAMBERLAIN, A. C., *Proc. Roy. Soc.*, A296 (1967), 45.

CLEMENT, C. F., "Aerosol nucleation and growth and their coupling to thermal-hydraulics," TP 1088, Publications Office, AERE Harwell, Oxfordshire, England OX11 ORA, 1984.

COLEBROOK, C. F., "Turbulent flow in pipes with particular reference to the transition region between the smooth and rough pipe laws," J. Institution Civil Engineers, 12 (1939), 133.

CRAY RESEARCH INC., "UPDATE reference manual," SR-0013, 1440 Northland Drive, Mendota Heights, Minnesota 55120, 1984.

CUNNINGHAM, E., *Proc. Roy. Soc.*, A83 (1910), 357.

DAVIES, C. N., *Proc. Phys. Soc.*, 57 (1945), 258.

DERJAGUIN, B. V. AND YALAMOV, YU. I., "The theory of thermophoresis and diffusiophoresis of aerosol particles and their experimental testing," in HIDY, G. M. AND BROCK, J. R. (Eds.), "International reviews in aerosol physics and chemistry," Vol. 3, Pergamon Press, 1972.

DRAKE, R. L., "A general mathematical survey of the coagulation equation," in HIDY, G. M. AND BROCK, J. R. (Eds.), "International reviews in aerosol physics and chemistry," Vol. 3, Pergamon Press, 1972.

DUNBAR, I. H., FERMANDJIAN, J., BUNZ, H., L'HOMME, A., HIMENO, Y., KIRBY, C. R., L'HIAUBET, G. AND MITSUTSUKA, N., "Comparison of sodium aerosol codes," Commission of the European Communities, EUR 9172 en, 1984.

FORD, J. R., "Common Los Alamos Mathematical Software," Computing Information Centre, Los Alamos National Laboratory, New Mexico 87545, 1984.

FUCHS, N. A., "The mechanics of aerosols," Pergamon Press, 1964.

GELBARD, F., "MAEROS user manual," Sandia National Laboratories, Albuquerque, New Mexico, NUREG/CR-1391, 1982.

GELBARD, F., "Aerosol growth in a nuclear reactor containment environment," Commission of the European Communities, September 9-11, 1987.

GRIMLEY, A. J., MAUDLIN, P. M., WHEATLEY, C. J. AND CAMP, W. J., "The VICTORIA computer code: Two dimensional species continuity with equilibrium chemistry and aerosol behavior," Unpublished, 1988.

HAHN, L. A., STUKEL, J. J., LEONG, K. H. AND HOPKE, P. K., *J. Aerosol Sci.*, 16 (1985), 81.

KADER, B. A. AND YAGLOM, A. M., *Int. J. Heat Mass Transfer*, 20 (1977), 345.

KELLER, K., "Aerosol behavior in closed containers," Kernforschungszentrum Karlsruhe, KfK 1758, 1973

LAUFER, J., "The structure of turbulence in fully developed pipe flow," National Advisory Committee for Aeronautics, NACA R 1174, 1954.

LIU, B. Y. H. AND AGARWAL, J. K., *J. Aerosol Sci.*, **5** (1974), 145.

MONIN, A. S., YAGLOM, A. M., "Statistical fluid mechanics: Mechanics of turbulence," Vol. 1, MIT Press, 1971.

OPCODE, INC., "HISTORIAN PLUS user's manual," P.O. Box 10998-537, Austin, Texas 78766-1998, 1985.

PRUPPACHER, H. R. AND KLETT, J. D., "Microphysics of clouds and precipitation," D. Reidel, 1978.

SAFFMAN, P. G. AND TURNER, J. S., *J. Fluid Mech.*, 1 (1956), 16.

SCHLICHTING, H., "Boundary layer theory," McGraw Hill, 1979.

SHAMPINE, L. F. AND WATTS, H. A., "DEPAC - Design of a user oriented package of ODE solvers," Sandia National Laboratories, Albuquerque, New Mexico, SAND-79-2374, 1979.

SIMONS, S., *Ann. Nucl. Energy*, **9** (1982), 473.

SITARSKI, M. AND SEINFELD, J. H., *J. Colloid Interface Sci.*, **61** (1977), 261.

STOCK, J. D. R., SIMONS, S., WILLIAMS, M. M. R., DUNBAR, I. H. AND RAMSDALE, S. A., *Annals Nuclear Energy*, **14** (1987), 1.

TSANG, T. H. AND BROCK, J. R., *Aerosol Sci. Technology*, **2** (1983), 311.

WELLS., A. C. AND CHAMBERLAIN A. C., *Br. J. Appl. Phys.*, **18** (1957), 1973.

WHEATLEY, C. J., "Solution of the aerosol equation with subdomain and collocation finite-element methods," Unpublished, intended for submission to *J. Comp. Phys.*, (1988).

APPENDIX A - THE EXAMPLE OUTPUT DATA FILES

We list here the output files obtained from the example input data file shown in Sub-section 4.4. We have removed the copies of the input data file from both output files. These output files are discussed in Sub-section 5.3.

The first file was written on the terminal. Following this you would also see:

```
charm     ctss time     3.356 seconds
cpu=    3.003 i/o=     .245 mem=      .109
```

The second file is the main output file, OUT.

The file written on the terminal

```
step no. =    0    time =  0.0000e+00    mass check =  0.0000e+00
step no. =    1    time =  3.0000e+02    mass check = -8.3442e-08
step no. =    2    time =  6.0000e+02    mass check = -7.1820e-08
step no. =    3    time =  9.0000e+02    mass check = -3.5321e-07
step no. =    4    time =  1.2000e+03    mass check = -3.5868e-07
step no. =    5    time =  1.5000e+03    mass check = -3.6546e-07
step no. =    6    time =  1.8000e+03    mass check = -3.7419e-07
step no. =    7    time =  2.1000e+03    mass check = -3.2025e-07
step no. =    8    time =  2.4000e+03    mass check = -1.1730e-07
step no. =    9    time =  2.7000e+03    mass check = -3.6089e-06
step no. =   10    time =  3.0000e+03    mass check = -3.6441e-06
step no. =   11    time =  3.3000e+03    mass check = -3.8971e-06
step no. =   12    time =  3.6000e+03    mass check = -3.8432e-06
step no. =   13    time =  7.2000e+03    mass check = -3.1102e-05
step no. =   14    time =  1.0800e+04    mass check =  5.0609e-05
step no. =   15    time =  1.4400e+04    mass check =  4.7371e-05
step no. =   16    time =  1.8000e+04    mass check =  4.8542e-05
step no. =   17    time =  2.1600e+04    mass check =  4.5610e-05
step no. =   18    time =  2.5200e+04    mass check =  4.5708e-05
step no. =   19    time =  2.8800e+04    mass check =  4.7011e-05
step no. =   20    time =  3.2400e+04    mass check =  4.7252e-05
step no. =   21    time =  3.6000e+04    mass check =  4.6496e-05
step no. =   22    time =  3.6300e+04    mass check =  4.1648e-05
step no. =   23    time =  3.6600e+04    mass check =  4.1020e-05
step no. =   24    time =  3.6900e+04    mass check =  4.0848e-05
step no. =   25    time =  3.7200e+04    mass check =  4.1042e-05
step no. =   26    time =  3.7500e+04    mass check =  4.1395e-05
step no. =   27    time =  3.7800e+04    mass check =  4.2587e-05
step no. =   28    time =  4.1400e+04    mass check =  4.5807e-05
step no. =   29    time =  4.5000e+04    mass check =  4.5840e-05
step no. =   30    time =  4.8600e+04    mass check =  4.5914e-05
step no. =   31    time =  5.2200e+04    mass check =  4.6001e-05
step no. =   32    time =  5.5800e+04    mass check =  4.5963e-05
step no. =   33    time =  5.9400e+04    mass check =  4.5963e-05
step no. =   34    time =  6.3000e+04    mass check =  4.5946e-05
step no. =   35    time =  6.6600e+04    mass check =  4.6273e-05
step no. =   36    time =  7.0200e+04    mass check =  4.6034e-05
step no. =   37    time =  7.3800e+04    mass check =  4.5466e-05
step no. =   38    time =  7.7400e+04    mass check =  4.5340e-05
step no. =   39    time =  8.1000e+04    mass check =  4.5453e-05
step no. =   40    time =  8.4600e+04    mass check =  4.5437e-05
step no. =   41    time =  8.8200e+04    mass check =  4.5415e-05
step no. =   42    time =  9.1800e+04    mass check =  4.5375e-05
step no. =   43    time =  9.5400e+04    mass check =  4.5587e-05
step no. =   44    time =  9.9000e+04    mass check =  4.5619e-05
step no. =   45    time =  1.0260e+05    mass check =  4.5690e-05
step no. =   46    time =  1.0620e+05    mass check =  4.5870e-05
step no. =   47    time =  1.0980e+05    mass check =  4.5813e-05
step no. =   48    time =  1.1340e+05    mass check =  4.5813e-05
step no. =   49    time =  1.1700e+05    mass check =  4.5813e-05
step no. =   50    time =  1.2060e+05    mass check =  4.5813e-05
step no. =   51    time =  1.2240e+05    mass check =  4.5814e-05
```

```
**********************************************************************************

**************************************************************
step no. =    0   time =  0.0000e+00   mass check =  0.0000e+00
**************************************************************

Mass budget...
   air-borne     flor dep.     wall dep.     clng dep.      source        leaked
  0.0000e+00  0.0000e+00  0.0000e+00  0.0000e+00  0.0000e+00  0.0000e+00
Airborne aerosol moments...
     sigma         rad50       mdensity      ndensity      geommean       mass50
  0.0000e+00  0.0000e+00  0.0000e+00  0.0000e+00  0.0000e+00  0.0000e+00
Source moments...
     sigma         rad50       mdensity      ndensity      geommean       mass50
  2.0000e+00  5.0000e-07  3.0864e-06  1.8292e+10  1.9419e-17  1.4661e-15
Airborne mass distribution...
        m-distr           m-distr           m-distr           m-distr           m-distr
   1  0.0000e+00    2  0.0000e+00    3  0.0000e+00    4  0.0000e+00    5  0.0000e+00
   6  0.0000e+00    7  0.0000e+00    8  0.0000e+00    9  0.0000e+00   10  0.0000e+00
  11  0.0000e+00   12  0.0000e+00   13  0.0000e+00
Airborne number distribution...
        n-distr           n-distr           n-distr           n-distr           n-distr
   1  0.0000e+00    2  0.0000e+00    3  0.0000e+00    4  0.0000e+00    5  0.0000e+00
   6  0.0000e+00    7  0.0000e+00    8  0.0000e+00    9  0.0000e+00   10  0.0000e+00
  11  0.0000e+00   12  0.0000e+00   13  0.0000e+00
Source mass distribution...
        m-distr           m-distr           m-distr           m-distr           m-distr
   1  3.3845e-15    2  1.6832e-12    3  2.4563e-10    4  1.0518e-08    5  1.3215e-07
   6  4.8719e-07    7  5.2702e-07    8  1.6728e-07    9  1.5580e-08   10  4.2580e-10
  11  3.4145e-12   12  8.0341e-15   13  5.5469e-18
Source number distribution...
        n-distr           n-distr           n-distr           n-distr           n-distr
   1  8.4611e+05    2  4.2080e+07    3  6.1407e+08    4  2.6294e+09    5  3.3037e+09
   6  1.2180e+09    7  1.3176e+08    8  4.1821e+06    9  3.8951e+04   10  1.0645e+02
  11  8.5381e-02   12  2.0085e-05   13  1.3867e-09
Collocation information...
 nelement    ncoll        hwidth           spacing           range
     2         13      1.0000e+00      1.0000e+01      1.0000e+12
Indexing for production coefficients...
  jbar   kbarmin kbarmax   nkbar    index    jbar   kbarmin kbarmax   nkbar    index
     0        0       12       13        1       1        0       12       13       14
     2        0        1        2       27       3        0        1        2       29
     4        0        1        2       31       5        0        1        2       33
     6        0        1        2       35       7        0        1        2       37
     8        0        1        2       39       9        0        1        2       41
    10        0        1        2       43      11        0        1        2       45
    12        0        1        2       47
Production coefficients in indexing order...
     pijk          pijk          pijk          pijk          pijk          pijk
  2.9905e-01  1.8370e-01  1.8547e-02  1.8304e-03  1.8280e-04  1.8278e-05
  1.8278e-06  1.8278e-07  1.8278e-08  1.8278e-09  1.8278e-10  1.8278e-11
  1.8278e-12  9.1753e-01  8.2254e-02  2.1534e-04  2.0208e-06  2.0084e-08
  2.0072e-10  2.0071e-12  2.0071e-14  2.0071e-16  2.0071e-18  2.0071e-20
  2.0071e-22  2.0071e-24  9.9326e-01  6.7385e-03  9.9934e-01  6.6450e-04
  9.9993e-01  6.6360e-05  9.9999e-01  6.6351e-06  1.0000e+00  6.6350e-07
  1.0000e+00  6.6350e-08  1.0000e+00  6.6350e-09  1.0000e+00  6.6350e-10
  1.0000e+00  6.6350e-11  1.0000e+00  6.6350e-12  1.0000e+00  6.6350e-13
Normalization factors...
        njk           njk           njk           njk           njk           njk
   1  8.9159e-01    2  9.3955e-01    3  9.9461e-01    4  9.9947e-01    5  9.9995e-01
   6  9.9999e-01    7  1.0000e+00    8  1.0000e+00    9  1.0000e+00   10  1.0000e+00
  11  1.0000e+00   12  1.0000e+00   13  1.0000e+00
   1  9.3955e-01    2  8.9159e-01    3  9.3955e-01    4  9.9461e-01    5  9.9947e-01
   6  9.9995e-01    7  9.9999e-01    8  1.0000e+00    9  1.0000e+00   10  1.0000e+00
  11  1.0000e+00   12  1.0000e+00   13  1.0000e+00
```

```
  1  9.9461e-01    2  9.3955e-01    3  8.9159e-01    4  9.3955e-01    5  9.9461e-01
  6  9.9947e-01    7  9.9995e-01    8  9.9999e-01    9  1.0000e+00   10  1.0000e+00
 11  1.0000e+00   12  1.0000e+00   13  1.0000e+00
  1  9.9947e-01    2  9.9461e-01    3  9.3955e-01    4  8.9159e-01    5  9.3955e-01
  6  9.9461e-01    7  9.9947e-01    8  9.9995e-01    9  9.9999e-01   10  1.0000e+00
 11  1.0000e+00   12  1.0000e+00   13  1.0000e+00
  1  9.9995e-01    2  9.9947e-01    3  9.9461e-01    4  9.3955e-01    5  8.9159e-01
  6  9.3955e-01    7  9.9461e-01    8  9.9947e-01    9  9.9995e-01   10  9.9999e-01
 11  1.0000e+00   12  1.0000e+00   13  1.0000e+00
  1  9.9999e-01    2  9.9995e-01    3  9.9947e-01    4  9.9461e-01    5  9.3955e-01
  6  8.9159e-01    7  9.3955e-01    8  9.9461e-01    9  9.9947e-01   10  9.9995e-01
 11  9.9999e-01   12  1.0000e+00   13  1.0000e+00
  1  1.0000e+00    2  9.9999e-01    3  9.9995e-01    4  9.9947e-01    5  9.9461e-01
  6  9.3955e-01    7  8.9159e-01    8  9.3955e-01    9  9.9461e-01   10  9.9947e-01
 11  9.9995e-01   12  9.9999e-01   13  1.0000e+00
  1  1.0000e+00    2  1.0000e+00    3  9.9999e-01    4  9.9995e-01    5  9.9947e-01
  6  9.9461e-01    7  9.3955e-01    8  8.9159e-01    9  9.3955e-01   10  9.9461e-01
 11  9.9947e-01   12  9.9995e-01   13  1.0000e+00
  1  1.0000e+00    2  1.0000e+00    3  1.0000e+00    4  9.9999e-01    5  9.9995e-01
  6  9.9947e-01    7  9.9461e-01    8  9.3955e-01    9  8.9159e-01   10  9.3955e-01
 11  9.9461e-01   12  9.9947e-01   13  1.0000e+00
  1  1.0000e+00    2  1.0000e+00    3  1.0000e+00    4  1.0000e+00    5  9.9999e-01
  6  9.9995e-01    7  9.9947e-01    8  9.9461e-01    9  9.3955e-01   10  8.9159e-01
 11  9.3955e-01   12  9.9461e-01   13  1.0001e+00
  1  1.0000e+00    2  1.0000e+00    3  1.0000e+00    4  1.0000e+00    5  1.0000e+00
  6  9.9999e-01    7  9.9995e-01    8  9.9947e-01    9  9.9461e-01   10  9.3955e-01
 11  8.9159e-01   12  9.3955e-01   13  1.0012e+00
  1  1.0000e+00    2  1.0000e+00    3  1.0000e+00    4  1.0000e+00    5  1.0000e+00
  6  1.0000e+00    7  9.9999e-01    8  9.9995e-01    9  9.9947e-01   10  9.9461e-01
 11  9.3955e-01   12  8.9159e-01   13  9.9888e-01
  1  1.0000e+00    2  1.0000e+00    3  1.0000e+00    4  1.0000e+00    5  1.0000e+00
  6  1.0000e+00    7  1.0000e+00    8  1.0000e+00    9  1.0000e+00   10  1.0001e+00
 11  1.0012e+00   12  9.9888e-01   13  3.3439e+00
Collocation points...
        mass          mass          mass          mass          mass
  1  4.0000e-21    2  4.0000e-20    3  4.0000e-19    4  4.0000e-18    5  4.0000e-17
  6  4.0000e-16    7  4.0000e-15    8  4.0000e-14    9  4.0000e-13   10  4.0000e-12
 11  4.0000e-11   12  4.0000e-10   13  4.0000e-09
Radii at the collocation points...
       radius        radius        radius        radius        radius
  1  6.9867e-09    2  1.5052e-08    3  3.2429e-08    4  6.9867e-08    5  1.5052e-07
  6  3.2429e-07    7  6.9867e-07    8  1.5052e-06    9  3.2429e-06   10  6.9867e-06
 11  1.5052e-05   12  3.2429e-05   13  6.9867e-05
Tolerance information...
      eps           eta           zeta          maxcalls      maxtrys
  1.0000e-06  0.0000e+00  5.0000e-01        30            10
Aerosol physics data...
  cshpfctr =  1.5000e+00  dshpfctr =  1.5000e+00  stickeff =  1.0000e+00
  aknudweb =  1.3700e+00  qknudweb =  4.0000e-01  bknudweb =  1.1000e+00
  pdensity =  2.8000e+03  pthrmcon =  6.3750e-01
    kbrock =  1.0000e+00    cmbrock =  1.3700e+00   ctbrock =  1.0000e+00
Cell data...
    volume =  1.8000e+05  leakrate =  1.1574e-07
  hydrdiam =  0.0000e+00  eqvrough =  4.5000e-05
  areaclng =  0.0000e+00  areawall =  2.0000e+04  areaflor =  2.8000e+03
  tempclng =  3.6315e+02  tempwall =  3.6315e+02  tempflor =  3.6315e+02
Gas data...
      temp =  3.7315e+02     press =  1.0000e+05  velocity =  0.0000e+00
  gdensity =  9.3345e-01   dynvisc =  2.1688e-05  mnfrpath =  8.8969e-08
     molwt =  2.8960e+01  gthrmcon =  2.5500e-02
    molwtv =  1.8015e+01   diffusv =  3.9159e-05
  vmfrclng =  0.0000e+00  vmfrwall =  0.0000e+00  vmfrflor =  0.0000e+00
  vcgrclng =  0.0000e+00  vcgrwall =  0.0000e+00  vcgrflor =  0.0000e+00
  vconclng =  0.0000e+00  vconwall =  0.0000e+00  vconflor =  0.0000e+00
Flow data...
  eddydiss =  0.0000e+00      ustar =  0.0000e+00  vbllthick =  1.0000e-03
Diffusion boundary layer thickness...
      dblthick      dblthick      dblthick      dblthick      dblthick
  1  1.0000e-04    2  1.0000e-04    3  1.0000e-04    4  1.0000e-04    5  1.0000e-04
```

```
   6  1.0000e-04   7  1.0000e-04   8  1.0000e-04   9  1.0000e-04  10  1.0000e-04
  11  1.0000e-04  12  1.0000e-04  13  1.0000e-04
Mobilities at the collocation points...
        mobility         mobility         mobility         mobility         mobility
   1  5.3958e+12   2  1.1983e+12   3  2.7824e+11   4  6.9072e+10   5  2.0005e+10
   6  6.9286e+09   7  2.7413e+09   8  1.1711e+09   9  5.2176e+08  10  2.3748e+08
  11  1.0021e+08  12  5.0475e+07  13  2.3381e+07
Rate of deposition onto floors...
        dep.rate         dep.rate         dep.rate         dep.rate         dep.rate
   1  2.2801e-06   2  2.0180e-06   3  1.8672e-06   4  1.7106e-06   5  1.5780e-06
   6  1.6339e-06   7  2.6097e-06   8  7.8452e-C8   9  3.2385e-05  10  1.4540e-04
  11  6.6705e-04  12  3.0814e-03  13  1.4272e-02
Rate of deposition onto walls...
        dep.rate         dep.rate         dep.rate         dep.rate         dep.rate
   1  1.6263e-05   2  1.4362e-05   3  1.3217e-05   4  1.1917e-05   5  1.0399e-05
   6  8.6500e-06   7  6.6885e-06   8  4.9772e-06   9  3.8335e-06  10  3.1910e-06
  11  2.8832e-06  12  2.7041e-06  13  2.6286e-06
Rate of deposition onto ceilings...
        dep.rate         dep.rate         dep.rate         dep.rate         dep.rate
   1  0.0000e+00   2  0.0000e+00   3  0.0000e+00   4  0.0000e+00   5  0.0000e+00
   6  0.0000e+00   7  0.0000e+00   8  0.0000e+00   9  0.0000e+00  10  0.0000e+00
  11  0.0000e+00  12  0.0000e+00  13  0.0000e+00
Agglomeration kernel...
        agg.rate         agg.rate         agg.rate         agg.rate         agg.rate
   1  8.2218e-15   2  1.2023e-14   3  2.3269e-14   4  4.5735e-14   5  9.0165e-14
   6  1.8255e-13   7  3.7960e-13   8  8.0311e-13   9  1.7150e-12  10  3.6791e-12
  11  7.9097e-12  12  1.7022e-11  13  3.6651e-11
   1  1.2023e-14   2  6.8148e-15   3  7.2192e-15   4  1.1199e-14   5  2.0530e-14
   6  4.0800e-14   7  8.4476e-14   8  1.7854e-13   9  3.8115e-13  10  8.1750e-13
  11  1.7573e-12  12  3.7815e-12  13  8.1423e-12
   1  2.3269e-14   2  7.2192e-15   3  3.6009e-15   4  3.5472e-15   5  5.3991e-15
   6  9.9687e-15   7  2.0005e-14   8  4.1693e-14   9  8.8434e-14  10  1.8913e-13
  11  4.0817e-13  12  8.7450e-13  13  1.8875e-12
   1  4.5735e-14   2  1.1199e-14   3  3.5472e-15   4  1.9052e-15   5  1.9339e-15
   6  2.9442e-15   7  5.4100e-15   8  1.0832e-14   9  2.2591e-14  10  4.8124e-14
  11  1.0403e-13  12  2.2874e-13  13  5.1725e-13
   1  9.0165e-14   2  2.0530e-14   3  5.3991e-15   4  1.9339e-15   5  1.1781e-15
   6  1.2555e-15   7  1.9127e-15   8  3.5323e-15   9  7.2827e-15  10  1.6310e-14
  11  4.0054e-14  12  1.1113e-13  13  3.5673e-13
   1  1.8255e-13   2  4.0800e-14   3  9.9687e-15   4  2.9442e-15   5  1.2555e-15
   6  8.7596e-16   7  1.0543e-15   8  1.9276e-15   9  4.8434e-15  10  1.5463e-14
  11  5.8208e-14  12  2.4305e-13  13  1.0705e-12
   1  3.7960e-13   2  8.4476e-14   3  2.0005e-14   4  5.4100e-15   5  1.9127e-15
   6  1.0543e-15   7  7.4551e-16   8  2.6613e-15   9  1.1293e-14  10  4.9902e-14
  11  2.2562e-13  12  1.0335e-12  13  4.7670e-12
   1  8.0311e-13   2  1.7854e-13   3  4.1693e-14   4  1.0832e-14   5  3.5323e-15
   6  1.9276e-15   7  2.6613e-15   8  6.8561e-16   9  3.8927e-14  10  2.1399e-13
  11  1.0206e-12  12  4.7511e-12  13  2.2038e-11
   1  1.7150e-12   2  3.8111e-13   3  8.8434e-14   4  2.2591e-14   5  7.2827e-15
   6  4.8434e-15   7  1.1293e-14   8  3.8927e-14   9  8.5779e-16  10  8.1155e-13
  11  4.5515e-12  12  2.1859e-11  13  9.0208e-10
   1  3.6791e-12   2  8.1750e-13   3  1.8913e-13   4  4.8124e-14   5  1.6310e-14
   6  1.5463e-14   7  4.9902e-14   8  2.1399e-13   9  8.1155e-13  10  8.4483e-16
  11  1.7358e-11  12  9.7689e-11  13  4.7004e-10
   1  7.9097e-12   2  1.7573e-12   3  4.0817e-13   4  1.0403e-13   5  4.0054e-14
   6  5.8288e-14   7  2.2562e-13   8  1.0206e-12   9  4.5515e-12  10  1.7258e-11
  11  6.3878e-16  12  3.7286e-10  13  2.1011e-09
   1  1.7022e-11   2  3.7815e-12   3  8.7450e-13   4  2.2874e-13   5  1.1113e-13
   6  2.4305e-13   7  1.0335e-12   8  4.7511e-12   9  2.1859e-11  10  9.7689e-11
  11  3.7286e-10  12  6.3595e-16  13  8.0220e-09
   1  3.6651e-11   2  8.1423e-12   3  1.8875e-12   4  5.1725e-13   5  3.5673e-13
   6  1.0705e-12   7  4.7670e-12   8  2.2038e-11   9  1.0208e-10  10  4.7004e-10
  11  2.1011e-09  12  8.0220e-09  13  6.3462e-16

*****************************************************************************

*************************************************************
step no. =    1   time =  3.0000e+02   mass check = -8.3442e-08
*************************************************************
```

```
Mass budget...
   air-borne    flor dep.    wall dep.    clng dep.     source       leaked
  1.6640e+02  8.0680e-02  1.8715e-01  0.0000e+00  1.6667e+02  2.8904e-03
Airborne aerosol moments...
     sigma        rad50      mdensity     ndensity     geommean      mass50
  1.8[illegible]2e+00  5.2914e-07  9.2442e-04  3.6018e+12  4.0855e-17  1.7376e-15
Source moments...
     sigma        rad50      mdensity     ndensity     geommean      mass50
  2.0000e+00  5.0000e-07  3.0864e-06  1.8292e+10  1.9419e-17  1.4661e-15

*****************************************************************************

*************************************************************
step no. =    2   time =  6.0000e+02   mass check = -7.1820e-08
*************************************************************

Mass budget...
   air-borne    flor dep.    wall dep.    clng dep.     source       leaked
  3.3225e+02  3.3359e-01  7.3329e-01  0.0000e+00  3.3333e+02  1.1549e-02
Airborne aerosol moments...
     sigma        rad50      mdensity     ndensity     geommean      mass50
  1.8661e+00  5.9020e-07  1.8459e-03  6.3885e+12  7.5102e-17  2.4113e-15
Source moments...
     sigma        rad50      mdensity     ndensity     geommean      mass50
  2.0000e+00  5.0000e-07  3.0864e-06  1.8292e+10  1.9419e-17  1.4661e-15

*****************************************************************************

*************************************************************
step no. =    3   time =  9.0000e+02   mass check = -3.5321e-07
*************************************************************

Mass budget...
   air-borne    flor dep.    wall dep.    clng dep.     source       leaked
  4.9757e+02  7.8986e-01  1.6113e+00  0.0000e+00  5.0000e+02  2.5957e-02
Airborne aerosol moments...
     sigma        rad50      mdensity     ndensity     geommean      mass50
  1.9069e+00  6.5451e-07  2.7643e-03  4.5489e+12  1.0506e-16  3.2884e-15
Source moments...
     sigma        rad50      mdensity     ndensity     geommean      mass50
  2.0000e+00  5.0000e-07  3.0864e-06  1.8292e+10  1.9419e-17  1.4661e-15
Airborne mass distribution...
        m-distr          m-distr          m-distr          m-distr          m-distr
   1  5.1273e-15    2  1.1346e-11    3  6.6519e-09    4  9.3500e-07    5  3.1827e-05
   6  3.0701e-04    7  6.2499e-04    8  2.1562e-04    9  1.9452e-05   10  6.6626e-07
  11  7.2226e-09   12  1.8851e-11   13  8.3635e-15
Airborne number distribution...
        n-distr          n-distr          n-distr          n-distr          n-distr
   1  1.2818e+06    2  2.8366e+08    3  1.6630e+10    4  2.3375e+11    5  7.9567e+11
   6  7.6753e+11    7  1.5625e+11    8  5.3906e+09    9  4.8631e+07   10  1.6656e+05
  11  1.8056e+02   12  4.7127e-02   13  2.0909e-06

*****************************************************************************

*************************************************************
step no. =    4   time =  1.2000e+03   mass check = -3.5668e-07
*************************************************************

Mass budget...
   air-borne    flor dep.    wall dep.    clng dep.     source       leaked
  6.6232e+02  1.4992e+00  2.7971e+00  0.0000e+00  6.6667e+02  4.6096e-02
Airborne aerosol moments...
     sigma        rad50      mdensity     ndensity     geommean      mass50
  1.9535e+00  7.1154e-07  3.6796e-03  4.5703e+12  1.2822e-16  4.2251e-15
Source moments...
     sigma        rad50      mdensity     ndensity     geommean      mass50
  2.0000e+00  5.0000e-07  3.0864e-06  1.8292e+10  1.9419e-17  1.4661e-15
```

```
**********************************************************************************

**********************************************************
step no. =    5   time =  1.5000e+03   mass check = -3.6546e-07
**********************************************************

Mass budget...
   air-borne    flor dep.    wall dep.    clng dep.     source       leaked
  8.2646e+02  2.5319e+00  4.2708e+00  0.0000e+00  8.3333e+02  7.1945e-02
Airborne aerosol moments...
     sigma        rad50      mdensity     ndensity     geommean      mass50
  1.9968e+00  7.6952e-07  4.5914e-03  4.5540e+12  1.4571e-18  5.3445e-18
Source moments...
     sigma        rad50      mdensity     ndensity     geommean      mass50
  2.0000e+00  5.0000e-07  3.0864e-06  1.8292e+10  1.9419e-17  1.4661e-18

**********************************************************************************

**********************************************************
step no. =    6   time =  1.8000e+03   mass check = -3.7419e-07
**********************************************************

Mass budget...
   air-borne    flor dep.    wall dep.    clng dep.     source       leaked
  9.8990e+02  3.9848e+00  6.0153e+00  0.0000e+00  1.0000e+03  1.0348e-01
Airborne aerosol moments...
     sigma        rad50      mdensity     ndensity     geommean      mass50
  2.0343e+00  8.3527e-07  5.4994e-03  4.5269e+12  1.5884e-18  6.8347e-18
Source moments...
     sigma        rad50      mdensity     ndensity     geommean      mass50
  2.0000e+00  5.0000e-07  3.0864e-06  1.8292e+10  1.9419e-17  1.4661e-18
Airborne mass distribution...
        m-distr          m-distr          m-distr          m-distr          m-distr
   1  3.8116e-18    2  9.6511e-12    3  5.7070e-09    4  8.1363e-07    5  2.7037e-05
   6  3.0151e-04    7  1.1964e-03    8  7.6284e-04    9  9.3264e-05   10  6.3285e-06
  11  1.7402e-07   12  1.3540e-09   13  1.9548e-12
Airborne number distribution...
        n-distr          n-distr          n-distr          n-distr          n-distr
   1  9.5289e+05    2  2.4128e+08    3  1.4267e+10    4  2.0341e+11    5  6.7592e+11
   6  7.5378e+11    7  2.9910e+11    8  1.9071e+10    9  2.3316e+08   10  1.5821e+06
  11  4.3504e+03   12  3.3851e+00   13  4.8870e-04

**********************************************************************************

**********************************************************
step no. =    7   time =  2.1000e+03   mass check = -3.2025e-07
**********************************************************

Mass budget...
   air-borne    flor dep.    wall dep.    clng dep.     source       leaked
  1.1525e+03  5.9927e+00  8.0150e+00  0.0000e+00  1.1667e+03  1.4068e-01
Airborne aerosol moments...
     sigma        rad50      mdensity     ndensity     geommean      mass50
  2.0855e+00  9.1087e-07  6.4029e-03  4.4977e+12  1.6848e-18  8.8639e-18
Source moments...
     sigma        rad50      mdensity     ndensity     geommean      mass50
  2.0000e+00  5.0000e-07  3.0864e-06  1.8292e+10  1.9419e-17  1.4661e-18

**********************************************************************************

**********************************************************
step no. =    8   time =  2.4000e+03   mass check = -1.1730e-07
**********************************************************

Mass budget...
   air-borne    flor dep.    wall dep.    clng dep.     source       leaked
  1.3141e+03  8.7487e+00  1.0255e+01  0.0000e+00  1.3333e+03  1.8351e-01
Airborne aerosol moments...
     sigma        rad50      mdensity     ndensity     geommean      mass50
```

```
  2.0910e+00  9.9484e-07  7.3008e-03  4.4698e+12  1.7523e-16  1.1548e-14
Source moments...
     sigma        rad50      mdensity     ndensity     geommean      mass50
  2.0000e+00  5.0000e-07  3.0864e-06  1.8292e+10  1.9419e-17  1.4661e-15

**********************************************************************

**********************************************************
step no. =   9   time =  2.7000e+03   mass check =  3.6089e-06
**********************************************************

Mass budget...
   air-borne    flor dep.    wall dep.    clng dep.     source       leaked
  1.4745e+03  1.2539e+01  1.2723e+01  0.0000e+00  1.5000e+03  2.3192e-01
Airborne aerosol moments...
     sigma        rad50      mdensity     ndensity     geommean      mass50
  2.1113e+00  1.0831e-06  8.1917e-03  4.4438e+12  1.7960e-16  1.4902e-14
Source moments...
     sigma        rad50      mdensity     ndensity     geommean      mass50
  2.0000e+00  5.0000e-07  3.0864e-06  1.8292e+10  1.9419e-17  1.4661e-15
Airborne mass distribution...
        m-distr          m-distr          m-distr          m-distr          m-distr
   1  3.9987e-15    2  9.1927e-12    3  5.4551e-09    4  7.8430e-07    5  2.6150e-05
   6  2.7587e-04    7  1.3492e-03    8  1.5376e-03    9  3.1734e-04   10  4.7160e-05
  11  3.3584e-06   12  7.4937e-08   13  3.1724e-10
Airborne number distribution...
        n-distr          n-distr          n-distr          n-distr          n-distr
   1  9.9918e+05    2  2.2982e+08    3  1.3638e+10    4  1.9608e+11    5  6.5376e+11
   6  6.8966e+11    7  3.3731e+11    8  3.8441e+10    9  7.9335e+08   10  1.1790e+07
  11  8.3959e+04   12  1.8734e+02   13  7.9309e-02

**********************************************************************

**********************************************************
step no. =  10   time =  3.0000e+03   mass check = -3.6441e-06
**********************************************************

Mass budget...
   air-borne    flor dep.    wall dep.    clng dep.     source       leaked
  1.6332e+03  1.7802e+01  1.8404e+01  0.0000e+00  1.6667e+03  2.8588e-01
Airborne aerosol moments...
     sigma        rad50      mdensity     ndensity     geommean      mass50
  2.1271e+00  1.1716e-06  9.0732e-03  4.4908e+12  1.8205e-16  1.7961e-14
Source moments...
     sigma        rad50      mdensity     ndensity     geommean      mass50
  2.0000e+00  5.0000e-07  3.0864e-06  1.8292e+10  1.9419e-17  1.4661e-15

**********************************************************************

**********************************************************
step no. =  11   time =  3.3000e+03   mass check = -3.8971e-06
**********************************************************

Mass budget...
   air-borne    flor dep.    wall dep.    clng dep.     source       leaked
  1.7895e+03  2.5216e+01  1.8236e+01  0.0000e+00  1.8333e+03  3.4531e-01
Airborne aerosol moments...
     sigma        rad50      mdensity     ndensity     geommean      mass50
  2.1390e+00  1.2583e-06  9.9416e-03  4.4010e+12  1.8299e-16  2.3369e-14
Source moments...
     sigma        rad50      mdensity     ndensity     geommean      mass50
  2.0000e+00  5.0000e-07  3.0864e-06  1.8292e+10  1.9419e-17  1.4661e-15

**********************************************************************

**********************************************************
step no. =  12   time =  3.6000e+03   mass check = -3.8432e-06
**********************************************************
```

```
Mass budget...
   air-borne    flor dep.    wall dep.    clng dep.     source       leaked
 1.9424e+03  3.5838e-01  2.1354e+01  0.0000e+00  2.0000e+03  4.1011e-01
Airborne aerosol moments...
     sigma        rad50      mdensity     ndensity     geommean      mass50
 2.1478e+00  1.3436e-06  1.0791e-02  4.3845e+12  1.8276e-16  2.8451e-14
Source moments...
     sigma        rad50      mdensity     ndensity     geommean      mass50
 2.0000e+00  5.0000e-07  3.0864e-06  1.8292e+10  1.9419e-17  1.4681e-15
Airborne mass distribution...
        m-distr          m-distr          m-distr          m-distr          m-distr
  1  6.8464e-15   2  9.0981e-12   3  5.4043e-09   4  7.7890e-07   5  2.6041e-05
  6  2.6713e-04   7  1.2857e-03   8  2.1426e-03   9  7.2295e-04  10  2.0660e-04
 11  3.2863e-05  12  1.9091e-06  13  2.3576e-08
Airborne number distribution...
        n-distr          n-distr          n-distr          n-distr          n-distr
  1  1.7116e+06   2  2.2745e+08   3  1.3511e+10   4  1.9472e+11   5  8.5103e+11
  6  6.6782e+11   7  3.2142e+11   8  5.3586e+10   9  1.8074e+09  10  5.1649e+07
 11  8.2158e+05  12  4.7728e+03  13  5.8941e+00

*****************************************************************************

*************************************************
step no. =  13   time =  7.2000e+03   mass check = -3.1102e-05
*************************************************

Mass budget...
   air-borne    flor dep.    wall dep.    clng dep.     source       leaked
 2.5552e+03  1.3783e+03  6.5045e+01  0.0000e+00  4.0000e+03  1.4441e+00
Airborne aerosol moments...
     sigma        rad50      mdensity     ndensity     geommean      mass50
 2.1272e+00  1.9690e-06  1.4196e-02  4.3777e+12  1.7032e-16  8.9533e-14
Source moments...
     sigma        rad50      mdensity     ndensity     geommean      mass50
 2.0000e+00  5.0000e-07  3.0864e-06  1.8292e+10  1.9419e-17  1.4681e-15

*****************************************************************************

*************************************************
step no. =  14   time =  1.0800e+04   mass check =  5.0609e-05
*************************************************

Mass budget...
   air-borne    flor dep.    wall dep.    clng dep.     source       leaked
 2.5168e+03  3.3722e+03  1.0845e+02  0.0000e+00  6.0000e+03  2.4908e+00
Airborne aerosol moments...
     sigma        rad50      mdensity     ndensity     geommean      mass50
 2.1308e+00  1.8995e-06  1.3982e-02  4.3757e+12  1.7167e-16  8.0385e-14
Source moments...
     sigma        rad50      mdensity     ndensity     geommean      mass50
 2.0000e+00  5.0000e-07  3.0864e-06  1.8292e+10  1.9419e-17  1.4681e-15

*****************************************************************************

*************************************************
step no. =  15   time =  1.4400e+04   mass check =  4.7371e-05
*************************************************

Mass budget...
   air-borne    flor dep.    wall dep.    clng dep.     source       leaked
 2.5167e+03  5.3278e+03  1.5197e+02  0.0000e+00  8.0000e+03  3.5397e+00
Airborne aerosol moments...
     sigma        rad50      mdensity     ndensity     geommean      mass50
 2.1303e+00  1.9043e-06  1.3981e-02  4.3762e+12  1.7159e-16  8.0989e-14
Source moments...
     sigma        rad50      mdensity     ndensity     geommean      mass50
 2.0000e+00  5.0000e-07  3.0864e-06  1.8292e+10  1.9419e-17  1.4681e-15
Airborne mass distribution...
        m-distr          m-distr          m-distr          m-distr          m-distr
```

```
   1  4.1983e-15    2  9.3044e-12    3  5.5188e-09    4  7.9288e-07    5  2.8447e-05
   6  2.7280e-04    7  1.1894e-03    8  1.9934e-03    9  1.1238e-03   10  7.2144e-04
  11  5.7548e-04   12  1.7187e-04   13  1.8658e-05
Airborne number distribution...
        n-distr          n-distr          n-distr          n-distr          n-distr
   1  1.0498e+08    2  2.3281e+08    3  1.3797e+10    4  1.9818e+11    5  8.8118e+11
   6  8.8199e+11    7  2.9235e+11    8  4.9835e+10    9  2.8098e+09   10  1.8038e+08
  11  1.4388e+07   12  4.2987e+05   13  4.1840e+03

*****************************************************************************

*************************************************************
step no. =  16   time =  1.8000e+04   mass check =  4.8542e-05
*************************************************************

Mass budget...
   air-borne    flor dep.    wall dep.    clng dep.     source       leaked
  2.5164e+03  7.2835e+03  1.9547e+02  0.0000e+00  1.0000e+04  4.5883e+00
Airborne aerosol moments...
     sigma         rad50      mdensity     ndensity     geommean      mass50
  2.1303e+00  1.9040e-08  1.3980e-02  4.3782e+12  1.7180e-18  8.0957e-14
Source moments...
     sigma         rad50      mdensity     ndensity     geommean      mass50
  2.0000e+00  5.0000e-07  3.0884e-08  1.8292e+10  1.9419e-17  1.4381e-15

*****************************************************************************

*************************************************************
step no. =  17   time =  2.1600e+04   mass check =  4.5810e-05
*************************************************************

Mass budget...
   air-borne    flor dep.    wall dep.    clng dep.     source       leaked
  2.5164e+03  9.2390e+03  2.3897e+02  0.0000e+00  1.2000e+04  5.6388e+00
Airborne aerosol moments...
     sigma         rad50      mdensity     ndensity     geommean      mass50
  2.1303e+00  1.9040e-08  1.3980e-02  4.3782e+12  1.7180e-18  8.0958e-14
Source moments...
     sigma         rad50      mdensity     ndensity     geommean      mass50
  2.0000e+00  5.0000e-07  3.0884e-08  1.8292e+10  1.9419e-17  1.4381e-15

*****************************************************************************

*************************************************************
step no. =  18   time =  2.5200e+04   mass check =  4.5708e-05
*************************************************************

Mass budget...
   air-borne    flor dep.    wall dep.    clng dep.     source       leaked
  2.5164e+03  1.1194e+04  2.8248e+02  0.0000e+00  1.4000e+04  6.6853e+00
Airborne aerosol moments...
     sigma         rad50      mdensity     ndensity     geommean      mass50
  2.1303e+00  1.9040e-08  1.3980e-02  4.3782e+12  1.7180e-18  8.0957e-14
Source moments...
     sigma         rad50      mdensity     ndensity     geommean      mass50
  2.0000e+00  5.0000e-07  3.0884e-08  1.8292e+10  1.9419e-17  1.4381e-15
Airborne mass distribution...
        m-distr          m-distr          m-distr          m-distr          m-distr
   1  4.1982e-15    2  9.3042e-12    3  5.5187e-09    4  7.9285e-07    5  2.8447e-05
   6  2.7278e-04    7  1.1898e-03    8  1.9932e-03    9  1.1239e-03   10  7.2132e-04
  11  5.7521e-04   12  1.7170e-04   13  1.8630e-05
Airborne number distribution...
        n-distr          n-distr          n-distr          n-distr          n-distr
   1  1.0495e+08    2  2.3280e+08    3  1.3797e+10    4  1.9818e+11    5  8.8117e+11
   6  8.8198e+11    7  2.9240e+11    8  4.9829e+10    9  2.8097e+09   10  1.8033e+08
  11  1.4380e+07   12  4.2924e+05   13  4.1576e+03

*****************************************************************************
```

```
**********************************************************
step no. =  19   time =  2.8800e+04   mass check =  4.7011e-05
**********************************************************

Mass budget...
   air-borne    flor dep.    wall dep.    clng dep.     source       leaked
  2.5184e+03  1.3150e+04  3.2598e+02  0.0000e+00  1.8000e+04  7.7338e+00
Airborne aerosol moments...
     sigma        rad50      mdensity     ndensity     geommean      mass50
  2.1303e+00  1.9040e-06  1.3980e-02  4.3762e+12  1.7160e-16  8.0957e-14
Source moments...
     sigma        rad50      mdensity     ndensity     geommean      mass50
  2.0000e+00  5.0000e-07  3.0864e-08  1.8292e+10  1.9419e-17  1.4661e-15

**********************************************************************

**********************************************************
step no. =  20   time =  3.2400e+04   mass check =  4.7252e-05
**********************************************************

Mass budget...
   air-borne    flor dep.    wall dep.    clng dep.     source       leaked
  2.5184e+03  1.5105e+04  3.6940e+02  0.0000e+00  1.8000e+04  8.7823e+00
Airborne aerosol moments...
     sigma        rad50      mdensity     ndensity     geommean      mass50
  2.1303e+00  1.9040e-06  1.3980e-02  4.3762e+12  1.7160e-16  8.0957e-14
Source moments...
     sigma        rad50      mdensity     ndensity     geommean      mass50
  2.0000e+00  5.0000e-07  3.0864e-08  1.8292e+10  1.9419e-17  1.4661e-15

**********************************************************************

**********************************************************
step no. =  21   time =  3.6000e+04   mass check =  4.6498e-05
**********************************************************

Mass budget...
   air-borne    flor dep.    wall dep.    clng dep.     source       leaked
  2.5184e+03  1.7061e+04  4.1295e+02  0.0000e+00  2.0000e+04  9.8308e+00
Airborne aerosol moments...
     sigma        rad50      mdensity     ndensity     geommean      mass50
  2.1303e+00  1.9040e-06  1.3980e-02  4.3762e+12  1.7160e-16  8.0957e-14
Source moments...
     sigma        rad50      mdensity     ndensity     geommean      mass50
  2.0000e+00  5.0000e-07  3.0864e-08  1.8292e+10  1.9419e-17  1.4661e-15
Airborne mass distribution...
        m-distr          m-distr          m-distr          m-distr          m-distr
   1  4.1983e-15    2  9.3042e-12    3  5.5187e-09    4  7.9265e-07    5  2.6447e-05
   6  2.7278e-04    7  1.1696e-03    8  1.9932e-03    9  1.1239e-03   10  7.2132e-04
  11  5.7621e-04   12  1.7170e-04   13  1.6630e-05
Airborne number distribution...
        n-distr          n-distr          n-distr          n-distr          n-distr
   1  1.0495e+06    2  2.3260e+08    3  1.3797e+10    4  1.9816e+11    5  6.6117e+11
   6  6.8196e-11    7  2.9240e+11    8  4.9829e+10    9  2.8097e+09   10  1.8033e+08
  11  1.4380e+07   12  4.2924e+05   13  4.1576e+03

**********************************************************************

**********************************************************
step no. =  22   time =  3.6300e+04   mass check =  4.1648e-05
**********************************************************

Mass budget...
   air-borne    flor dep.    wall dep.    clng dep.     source       leaked
  2.3503e+03  1.7223e+04  4.1640e+02  0.0000e+00  2.0000e+04  9.9153e+00
Airborne aerosol moments...
     sigma        rad50      mdensity     ndensity     geommean      mass50
  1.9358e+00  2.0510e-06  1.3057e-02  2.1450e+12  5.5522e-16  1.0119e-13
Source moments...
```

```
    sigma        rad50      mdensity      ndensity     geommean      mass50
  2.0000e+00   5.0000e-07   0.0000e+00   0.0000e+00   1.9419e-17   1.4661e-15

*****************************************************************************

*************************************************************
step no. =  23   time =  3.6600e+04   mass check =  4.1020e-05
*************************************************************

Mass budget...
   air-borne    flor dep.    wall dep.    clng dep.     source       leaked
  2.1866e+03   1.7384e+04   4.1954e+02   0.0000e+00   2.0000e+04   9.9941e+00
Airborne aerosol moments...
    sigma        rad50      mdensity      ndensity     geommean      mass50
  1.8922e+00   2.1800e-08   1.2148e-02   1.4572e+12   8.8039e-16   1.2151e-13
Source moments...
    sigma        rad50      mdensity      ndensity     geommean      mass50
  2.0000e+00   5.0000e-07   0.0000e+00   0.0000e+00   1.9419e-17   1.4661e-15

*****************************************************************************

*************************************************************
step no. =  24   time =  3.6900e+04   mass check =  4.0848e-05
*************************************************************

Mass budget...
   air-borne    flor dep.    wall dep.    clng dep.     source       leaked
  2.0281e+03   1.7539e+04   4.2239e+02   0.0000e+00   2.0000e+04   1.0067e+01
Airborne aerosol moments...
    sigma        rad50      mdensity      ndensity     geommean      mass50
  1.8883e+00   2.2903e-08   1.1267e-02   1.0997e+12   1.1661e-15   1.4091e-13
Source moments...
    sigma        rad50      mdensity      ndensity     geommean      mass50
  2.0000e+00   5.0000e-07   0.0000e+00   0.0000e+00   1.9419e-17   1.4661e-15
Airborne mass distribution...
       m-distr          m-distr          m-distr          m-distr          m-distr
  1 -5.1760e-23   2 -4.2209e-20   3  1.0480e-18   4  7.9735e-10   5  1.9142e-06
  6  8.2878e-05   7  7.1373e-04   8  1.8665e-03   9  1.0224e-03  10  8.7864e-04
 11  5.5185e-04  12  1.6064e-04  13  1.4939e-05
Airborne number distribution...
       n-distr          n-distr          n-distr          n-distr          n-distr
  1 -1.2940e-02   2 -1.0552e+00   3  2.6189e+00   4  1.9034e+08   5  9.7854e+10
  6  2.0670e+11   7  1.7843e+11   8  4.1664e+10   9  2.5561e+09  10  1.6286e+08
 11  1.3738e+07  12  4.0159e+05  13  3.7348e+03

*****************************************************************************

*************************************************************
step no. =  25   time =  3.7200e+04   mass check =  4.1042e-05
*************************************************************

Mass budget...
   air-borne    flor dep.    wall dep.    clng dep.     source       leaked
  1.8778e+03   1.7687e+04   4.2500e+02   0.0000e+00   2.0000e+04   1.0135e+01
Airborne aerosol moments...
    sigma        rad50      mdensity      ndensity     geommean      mass50
  1.8517e+00   2.3817e-08   1.0431e-02   8.7799e+11   1.4184e-15   1.5846e-13
Source moments...
    sigma        rad50      mdensity      ndensity     geommean      mass50
  2.0000e+00   5.0000e-07   0.0000e+00   0.0000e+00   1.9419e-17   1.4661e-15

*****************************************************************************

*************************************************************
step no. =  26   time =  3.7500e+04   mass check =  4.1395e-05
*************************************************************

Mass budget...
   air-borne    flor dep.    wall dep.    clng dep.     source       leaked
```

```
  1.7372e+03  1.7825e+04  4.2739e-02  0.0000e+00  2.0000e+04  1.01[illegible]e+01
Airborne aerosol moments...
     sigma        rad50      mdensity     ndensity     geommean      mass50
  1.8395e+00  2.4552e-08  9.6513e-03  7.2683e+11  1.6347e-15  1.7359e-13
Source moments...
     sigma        rad50      mdensity     ndensity     geommean      mass50
  2.0000e+00  5.0000e-07  0.0000e+00  0.0000e+00  1.9419e-17  1.4681e-15

*******************************************************************************

*************************************************
step no. =  27   time =  3.7800e+04   mass check =  4.2587e-05
*************************************************

Mass budget...
   air-borne    flor dep.    wall dep.    clng dep.     source       leaked
  1.8080e+03  1.7952e+04  4.2959e-02  0.0000e+00  2.0000e+04  1.0256e+01
Airborne aerosol moments...
     sigma        rad50      mdensity     ndensity     geommean      mass50
  1.8303e+00  2.5128e-08  8.9335e-03  6.1730e+11  1.8253e-15  1.8610e-13
Source moments...
     sigma        rad50      mdensity     ndensity     geommean      mass50
  2.0000e+00  5.0000e-07  0.[illegible]e+00  0.0000e+00  1.9419e-17  1.4681e-15
Airborne mass distribution...
       m-distr          m-distr          m-distr          m-distr          m-distr
   1 -3.4222e-24    2  3.4494e-23    3  2.8243e-21    4  2.2042e-11    5  5.1163e-07
   6  4.1328e-05    7  4.6751e-04    8  1.3118e-03    9  8.5784e-04   10  5.8347e-04
  11  4.8218e-04   12  1.2524e-04   13  1.0070e-05
Airborne number distribution...
       n-distr          n-distr          n-distr          n-distr          n-distr
   1 -8.5555e-04    2  8.[illegible]236e-04    3  7.0808e-03    4  5.5106e+08    5  1.2791e+10
   6  1.0332e+11    7  1.16[illegible]9e+11    8  3.2795e+10    9  2.1441e+09   10  1.4587e+08
  11  1.2055e+07   12  3.1310[illegible]+05   13  2.5176e+03

*******************************************************************************

*************************************************
step no. =  28   time =  4.1400e+04   mass check =  4.5807e-05
*************************************************

Mass budget...
   air-borne    flor dep.    wall dep.    clng dep.     source       leaked
  7.4924e+02  1.8794e+04  4.4654e-02  0.0000e+00  2.0000e+04  1.0712e+01
Airborne aerosol moments...
     sigma        rad50      mdensity     ndensity     geommean      mass50
  1.7954e+00  2.6488e-08  4.1625e-03  2.0080e+11  2.9991e-15  2.1749e-13
Source moments...
     sigma        rad50      mdensity     ndensity     geommean      mass50
  2.0000e+00  5.0000e-07  0.0000e+00  0.0000e+00  1.9419e-17  1.4681e-15

*******************************************************************************

*************************************************
step no. =  29   time =  4.5000e+04   mass check =  4.5840e-05
*************************************************

Mass budget...
   air-borne    flor dep.    wall dep.    clng dep.     source       leaked
  4.5430e+02  1.9079e+04  4.5551e-02  0.0000e+00  2.0000e+04  1.0954e+01
Airborne aerosol moments...
     sigma        rad50      mdensity     ndensity     geommean      mass50
  1.7886e+00  2.5873e-08  2.5239e-03  1.1307e+11  3.4472e-15  2.0313e-13
Source moments...
     sigma        rad50      mdensity     ndensity     geommean      mass50
  2.0000e+00  5.0000e-07  0.0000e+00  0.0000e+00  1.9419e-17  1.4681e-15

*******************************************************************************

*************************************************
```

```
step no. =  30   time =  4.8600e+04   mass check =  4.5914e-05
*************************************************************

Mass budget...
   air-borne    flor dep.    wall dep.    clng dep.     source       leaked
  3.1588e+02  1.9212e+04  4.8139e+02  0.0000e+00  2.0000e+04  1.1111e+01
Airborne aerosol moments...
     sigma        rad50      mdensity     ndensity     geommean      mass50
  1.7848e+00  2.5272e-08  1.7549e-03  7.6737e+10  3.6880e-15  1.8931e-13
Source moments...
     sigma        rad50      mdensity     ndensity     geommean      mass50
  2.0000e+00  5.0000e-07  0.0000e+00  0.0000e+00  1.9419e-17  1.4661e-15
Airborne mass distribution...
        m-distr          m-distr          m-distr          m-distr          m-distr
   1  1.3458e-23    2 -1.7186e-24    3 -1.1105e-25    4  1.2102e-16    5  6.2822e-09
   6  3.5897e-08    7  6.6789e-05    8  2.7712e-04    9  2.1144e-04   10  1.6397e-04
  11  3.7088e-05   12  2.1145e-06   13  2.0952e-08
Airborne number distribution...
        n-distr          n-distr          n-distr          n-distr          n-distr
   1  3.3646e-03    2 -4.296[illegible]-05    3 -2.7762e-07    4  3.0255e+01    5  1.5656e+08
   6  8.9741e+09    7  1.6697e+10    8  8.9280e+09    9  5.2860e+08   10  4.0992e+07
  11  9.2715e+05   12  5.2863e+03   13  5.2381e+00

*************************************************************

*************************************************************
step no. =  31   time =  5.2200e+04   mass check =  4.6001e-05
*************************************************************

Mass budget...
   air-borne    flor dep.    wall dep.    clng dep.     source       leaked
  2.3584e+02  1.9287e+04  4.6565e+02  0.0000e+00  2.0000e+04  1.1225e+01
Airborne aerosol moments...
     sigma        rad50      mdensity     ndensity     geommean      mass50
  1.7815e+00  2.4498e-08  1.3102e-03  5.7265e+10  3.8448e-15  1.7245e-13
Source moments...
     sigma        rad50      mdensity     ndensity     geommean      mass50
  2.0000e+00  5.0000e-07  0.0000e+00  0.0000e+00  1.9419e-17  1.4661e-15

*************************************************************

*************************************************************
step no. =  32   time =  5.5800e+04   mass check =  4.5963e-05
*************************************************************

Mass budget...
   air-borne    flor dep.    wall dep.    clng dep.     source       leaked
  1.5400e+02  1.9238e+04  4.6894e+02  0.0000e+00  2.0000e+04  1.1311e+01
Airborne aerosol moments...
     sigma        rad50      mdensity     ndensity     geommean      mass50
  1.7786e+00  2.3601e-08  1.0228e-03  4.5253e+10  3.9823e-15  1.5419e-13
Source moments...
     sigma        rad50      mdensity     ndensity     geommean      mass50
  2.0000e+00  5.0000e-07  0.0000e+00  0.0000e+00  1.9419e-17  1.4661e-15

*************************************************************

*************************************************************
step no. =  33   time =  5.9400e+04   mass check =  4.5963e-05
*************************************************************

Mass budget...
   air-borne    flor dep.    wall dep.    clng dep.     source       leaked
  1.4838e+02  1.9369e+04  4.7157e+02  0.0000e+00  2.0000e+04  1.1380e+01
Airborne aerosol moments...
     sigma        rad50      mdensity     ndensity     geommean      mass50
  1.7762e+00  2.2717e-08  8.2436e-04  3.7150e+10  4.0598e-15  1.3750e-13
Source moments...
     sigma        rad50      mdensity     ndensity     geommean      mass50
```

```
  2.0000e+00  5.0000e-07  0.0000e+00  0.0000e+00  1.9419e-17  1.4681e-15
Airborne mass distribution...
        m-distr          m-distr          m-distr          m-distr          m-distr
   1  2.1814e-21    2  9.8148e-23    3  1.7230e-25    4  2.4821e-18    5  1.5792e-09
   6  1.6108e-08    7  3.2682e-05    8  1.4450e-04    9  1.0968e-04   10  6.3448e-05
  11  5.9710e-06   12  1.3989e-07   13  4.4123e-10
Airborne number distribution...
        n-distr          n-distr          n-distr          n-distr          n-distr
   1  5.4535e-01    2  2.4037e-03    3  4.3075e-07    4  6.2052e-01    5  3.9479e+07
   6  4.0284e+09    7  8.1654e+09    8  3.8125e+09    9  2.7420e+08   10  1.5862e+07
  11  1.4928e+05   12  3.4923e+02   13  1.1031e-01

*******************************************************************************

*************************************************************
step no. =  34   time =  8.3000e+04   mass check =  4.5948e-05
*************************************************************

Mass budget...
   air-borne    flor dep.    wall dep.    clng dep.     source       leaked
  1.2278e+02  1.9392e+04  4.7374e+02  0.0000e+00  2.0000e+04  1.1438e+01
Airborne aerosol moments...
     sigma         rad50       mdensity     ndensity     geommean      mass50
  1.7742e+00  2.1933e-08  8.8199e-04  3.1329e+10  4.1442e-15  1.2374e-13
Source moments...
     sigma         rad50       mdensity     ndensity     geommean      mass50
  2.0000e+00  5.0000e-07  0.0000e+00  0.0000e+00  1.9419e-17  1.4681e-15

*******************************************************************************

*************************************************************
step no. =  35   time =  8.6600e+04   mass check =  4.8273e-05
*************************************************************

Mass budget...
   air-borne    flor dep.    wall dep.    clng dep.     source       leaked
  1.0372e+02  1.9409e+04  4.7657e+02  0.0000e+00  2.0000e+04  1.1483e+01
Airborne aerosol moments...
     sigma         rad50       mdensity     ndensity     geommean      mass50
  1.7724e+00  2.1273e-08  8.7624e-04  2.8950e+10  4.2192e-15  1.1292e-13
Source moments...
     sigma         rad50       mdensity     ndensity     geommean      mass50
  2.0000e+00  5.0000e-07  0.0000e+00  0.0000e+00  1.9419e-17  1.4681e-15

*******************************************************************************

*************************************************************
step no. =  98   time =  7.0200e+04   mass check =  4.8034e-05
*************************************************************

Mass budget...
   air-borne    flor dep.    wall dep.    clng dep.     source       leaked
  8.9185e+01  1.0422e+04  4.7714e+02  0.0000e+00  2.0000e+04  1.1523e+01
Airborne aerosol moments...
     sigma         rad50       mdensity     ndensity     geommean      mass50
  1.7709e+00  2.0732e-08  4.9536e-04  2.3538e+10  4.2880e-15  1.0451e-13
Source moments...
     sigma         rad50       mdensity     ndensity     geommean      mass50
  2.0000e+00  5.0000e-07  0.0000e+00  0.0000e+00  1.9419e-17  1.4681e-15
Airborne mass distribution...
        m-distr          m-distr          m-distr          m-distr          m-distr
   1  3.6072e-20    2  1.0058e-22    3  4.0920e-26    4  2.7196e-19    5  6.8296e-10
   6  9.7388e-07    7  2.0752e-05    8  9.5830e-05    9  7.2320e-05   10  2.3889e-05
  11  1.3728e-06   12  1.5245e-08   13 -3.8987e-10
Airborne number distribution...
        n-distr          n-distr          n-distr          n-distr          n-distr
   1  9.0180e+00    2  2.5145e-03    3  1.0230e-07    4  6.7990e-02    5  1.7074e+07
   6  2.4347e+09    7  5.1880e+09    8  2.3957e+09    9  1.8080e+08   10  5.9671e+07
  11  3.4321e+04   12  3.8111e+01   13 -9.7468e-02
```

```
**********************************************************************

**********************************************************
step no. =  37   time =  7.3800e+04   mass check =  4.5466e-05
**********************************************************

Mass budget...
   air-borne    flor dep.    wall dep.    clng dep.     source       leaked
  7.7731e-01  1.9432e+04  4.7852e+02  0.0000e+00  2.0000e+04  1.1558e+01
Airborne aerosol moments...
     sigma        rad50      mdensity     ndensity     geommean      mass50
  1.7693e+00  2.0286e-06  4.3184e-04  2.0805e+10  4.3453e-15  9.7912e-14
Source moments...
     sigma        rad50      mdensity     ndensity     geommean      mass50
  2.0000e+00  5.0000e-07  0.0000e+00  0.0000e+00  1.9419e-17  1.4681e-15

**********************************************************************

**********************************************************
step no. =  38   time =  7.7400e+04   mass check =  4.5340e-05
**********************************************************

Mass budget...
   air-borne    flor dep.    wall dep.    clng dep.     source       leaked
  6.8530e+01  1.9440e+04  4.7973e+02  0.0000e+00  2.0000e+04  1.1589e+01
Airborne aerosol moments...
     sigma        rad50      mdensity     ndensity     geommean      mass50
  1.7678e+00  1.9912e-06  3.8072e-04  1.8568e+10  4.3979e-15  9.2602e-14
Source moments...
     sigma        rad50      mdensity     ndensity     geommean      mass50
  2.0000e+00  5.0000e-07  0.0000e+00  0.0000e+00  1.9419e-17  1.4681e-15

**********************************************************************

**********************************************************
step no. =  39   time =  8.1000e+04   mass check =  4.5453e-05
**********************************************************

Mass budget...
   air-borne    flor dep.    wall dep.    clng dep.     source       leaked
  6.0966e+01  1.9447e+04  4.8080e+02  0.0000e+00  2.0000e+04  1.1616e+01
Airborne aerosol moments...
     sigma        rad50      mdensity     ndensity     geommean      mass50
  1.7661e+00  1.9590e-06  3.3870e-04  1.6705e+10  4.4442e-15  8.8179e-14
Source moments...
     sigma        rad50      mdensity     ndensity     geommean      mass50
  2.0000e+00  5.0000e-07  0.0000e+00  0.0000e+00  1.9419e-17  1.4681e-15
Airborne mass distribution...
       m-distr          m-distr          m-distr          m-distr          m-distr
  1 -2.6526e-20   2  2.9815e-23   3 -9.5324e-28   4  6.9910e-20   5  3.7088e-10
  6  6.6798e-07   7  1.4755e-05   8  7.0241e-05   9  5.1397e-06  10  9.6505e-06
 11  3.8124e-07  12  2.3619e-09  13  2.0986e-12
Airborne number distribution...
       n-distr          n-distr          n-distr          n-distr          n-distr
  1 -6.6316e+00   2  7.4538e-04   3 -2.3831e-09   4  1.4978e-02   5  9.2720e+06
  6  1.6700e+09   7  3.6888e+09   8  1.7560e+09   9  1.2849e+08  10  2.4126e+06
 11  9.5309e+03  12  5.9048e+00  13  5.2466e-04

**********************************************************************

**********************************************************
step no. =  40   time =  8.4600e+04   mass check =  4.5437e-05
**********************************************************

Mass budget...
   air-borne    flor dep.    wall dep.    clng dep.     source       leaked
  5.4631e+01  1.9452e+04  4.8177e+02  0.0000e+00  2.0000e+04  1.1640e+01
Airborne aerosol moments...
```

```
     sigma        rad50      mdensity     ndensity     geommean      mass50
  1.7642e+00  1.9303e-06  3.0351e-04  1.5130e+10  4.4848e-15  8.4355e-14
Source moments...
     sigma        rad50      mdensity     ndensity     geommean      mass50
  2.0000e+00  5.0000e-07  0.0000e+00  0.0000e+00  1.9419e-17  1.4661e-15

*****************************************************************************

*************************************************
step no. =  41   time =  8.8200e+04   mass check =  4.5415e-05
*************************************************

Mass budget...
   air-borne    flor dep.    wall dep.    clng dep.     source       leaked
  4.9243e+01  1.9458e+04  4.8264e+02  0.0000e+00  2.0000e+04  1.1661e+01
Airborne aerosol moments...
     sigma        rad50      mdensity     ndensity     geommean      mass50
  1.7622e+00  1.9038e-06  2.7357e-04  1.3783e+10  4.5203e-15  8.0935e-14
Source moments...
     sigma        rad50      mdensity     ndensity     geommean      mass50
  2.0000e+00  5.0000e-07  0.0000e+00  0.0000e+00  1.9419e-17  1.4661e-15

*****************************************************************************

*************************************************
step no. =  42   time =  9.1800e+04   mass check =  4.5375e-05
*************************************************

Mass budget...
   air-borne    flor dep.    wall dep.    clng dep.     source       leaked
  4.4600e+01  1.9460e+04  4.8344e+02  0.0000e+00  2.0000e+04  1.1681e+01
Airborne aerosol moments...
     sigma        rad50      mdensity     ndensity     geommean      mass50
  1.7601e+00  1.8789e-06  2.4778e-04  1.2818e+10  4.5512e-15  7.7792e-14
Source moments...
     sigma        rad50      mdensity     ndensity     geommean      mass50
  2.0000e+00  5.0000e-07  0.0000e+00  0.0000e+00  1.9419e-17  1.4661e-15
Airborne mass distribution...
      m-distr          m-distr          m-distr          m-distr          m-distr
  1 -3.5820e-20    2 -1.0489e-23    3 -1.4801e-28    4  1.8478e-20    5  2.2854e-10
  6  4.9135e-07    7  1.1174e-05    8  5.4326e-05    9  3.7118e-05   10  4.3750e-06
 11  1.2350e-07   12  4.8239e-10   13  9.3030e-11
Airborne number distribution...
      n-distr          n-distr          n-distr          n-distr          n-distr
  1 -8.9051e+00    2 -2.8223e-04    3 -3.7002e-10    4  4.8195e-03    5  5.7138e+06
  6  1.2284e+09    7  2.7936e+09    8  1.3581e+09    9  9.2794e+07   10  1.0938e+06
 11  3.0875e+03   12  1.2080e+00   13  1.5737e-02

*****************************************************************************

*************************************************
step no. =  43   time =  9.5400e+04   mass check =  4.5587e-05
*************************************************

Mass budget...
   air-borne    flor dep.    wall dep.    clng dep.     source       leaked
  4.0557e+01  1.9464e+04  4.8416e+02  0.0000e+00  2.0000e+04  1.1698e+01
Airborne aerosol moments...
     sigma        rad50      mdensity     ndensity     geommean      mass50
  1.7578e+00  1.8549e-06  2.2532e-04  1.1601e+10  4.5781e-15  7.4847e-14
Source moments...
     sigma        rad50      mdensity     ndensity     geommean      mass50
  2.0000e+00  5.0000e-07  0.0000e+00  0.0000e+00  1.9419e-17  1.4661e-15

*****************************************************************************

*************************************************
step no. =  44   time =  9.9000e+04   mass check =  4.5619e-05
*************************************************
```

```
Mass budget...
   air-borne    flor dep.    wall dep.    clng dep.     source       leaked
  3.7007e+01   1.9486e+04   4.8482e+02   0.0000e+00   2.0000e+04   1.1715e+01
Airborne aerosol moments...
     sigma        rad50      mdensity     ndensity     geommean      mass50
  1.7554e+00   1.8315e-06   2.0559e-04   1.0707e+10   4.6016e-15   7.2055e-14
Source moments...
     sigma        rad50      mdensity     ndensity     geommean      mass50
  2.0000e+00   5.0000e-07   0.0000e+00   0.0000e+00   1.9419e-17   1.4661e-15

*****************************************************************************

*************************************************************
step no. =  45    time =  1.0280e+05    mass check =  4.5690e-05
*************************************************************

Mass budget...
   air-borne    flor dep.    wall dep.    clng dep.     source       leaked
  3.3887e+01   1.9469e+04   4.8542e+02   0.0000e+00   2.0000e+04   1.1729e+01
Airborne aerosol moments...
     sigma        rad50      mdensity     ndensity     geommean      mass50
  1.7529e+00   1.8086e-06   1.8815e-04   9.9161e+09   4.6220e-15   6.9391e-14
Source moments...
     sigma        rad50      mdensity     ndensity     geommean      mass50
  2.0000e+00   5.0000e-07   0.0000e+00   0.0000e+00   1.9419e-17   1.4661e-15
Airborne mass distribution...
        m-distr              m-distr              m-distr              m-distr              m-distr
   1  3.1743e-20    2 -2.3109e-23    3  1.7541e-29    4  7.2626e-21    5  1.5264e-10
   6  3.7787e-07    7  8.8140e-06    8  4.3480e-05    9  2.6792e-05   10  2.2040e-06
  11  4.5370e-08   12  1.2082e-10   13 -1.6069e-10
Airborne number distribution...
        n-distr              n-distr              n-distr              n-distr              n-distr
   1  7.9357e+00    2 -5.7772e-04    3  4.3853e-11    4  1.8157e-03    5  3.8159e+06
   6  9.4467e+08    7  2.2035e+09    8  1.0870e+09    9  8.6980e+07   10  5.5100e+05
  11  1.1342e+03   12  3.0206e-01   13 -4.0172e-02

*****************************************************************************

*************************************************************
step no. =  46    time =  1.0620e+05    mass check =  4.5870e-05
*************************************************************

Mass budget...
   air-borne    flor dep.    wall dep.    clng dep.     source       leaked
  3.1075e+01   1.9471e+04   4.8598e+02   0.0000e+00   2.0000e+04   1.1743e+01
Airborne aerosol moments...
     sigma        rad50      mdensity     ndensity     geommean      mass50
  1.7503e+00   1.7863e-06   1.7264e-04   9.2111e+09   4.6397e-15   6.6849e-14
Source moments...
     sigma        rad50      mdensity     ndensity     geommean      mass50
  2.0000e+00   5.0000e-07   0.0000e+00   0.0000e+00   1.9419e-17   1.4661e-15

*****************************************************************************

*************************************************************
step no. =  47    time =  1.0980e+05    mass check =  4.5813e-05
*************************************************************

Mass budget...
   air-borne    flor dep.    wall dep.    clng dep.     source       leaked
  2.8581e+01   1.9473e+04   4.8649e+02   0.0000e+00   2.0000e+04   1.1755e+01
Airborne aerosol moments...
     sigma        rad50      mdensity     ndensity     geommean      mass50
  1.7476e+00   1.7644e-06   1.5878e-04   8.5794e+09   4.6552e-15   6.4423e-14
Source moments...
     sigma        rad50      mdensity     ndensity     geommean      mass50
  2.0000e+00   5.0000e-07   0.0000e+00   0.0000e+00   1.9419e-17   1.4661e-15
```

```
**********************************************************************************

**********************************************************
step no. =  48    time =  1.1340e+05    mass check =  4.5813e-05
**********************************************************

Mass budget...
   air-borne    flor dep.    wall dep.    clng dep.     source       leaked
  2.6343e+01   1.9475e+04   4.8696e+02   0.0000e+00   2.0000e+04   1.1767e+01
Airborne aerosol moments...
     sigma         rad50       mdensity     ndensity     geommean      mass50
  1.7449e+00   1.7431e-06   1.4635e-04   8.0106e+09   4.8688e-15   6.2118e-14
Source moments...
     sigma         rad50       mdensity     ndensity     geommean      mass50
  2.0000e+00   5.0000e-07   0.0000e+00   0.0000e+00   1.9419e-17   1.4681e-15
Airborne mass distribution...
        m-distr             m-distr             m-distr             m-distr             m-distr
   1 -8.9422e-22    2 -6.4798e-25    3  1.3112e-30    4  3.3483e-21    5  1.0775e-10
   6  2.9983e-07    7  7.1518e-06    8  3.5831e-05    9  1.9281e-05   10  1.1959e-06
  11  1.8224e-08   12  3.4217e-11   13 -1.4100e-13
Airborne number distribution...
        n-distr             n-distr             n-distr             n-distr             n-distr
   1 -2.2355e-01    2 -1.8199e-05    3  3.2781e-12    4  8.3658e-04    5  2.8937e+06
   6  7.4908e+08    7  1.7879e+09    8  8.9079e+08    9  4.8154e+07   10  2.9897e+05
  11  4.5580e+02   12  8.5544e-02   13 -3.5250e-05

**********************************************************************************

**********************************************************
step no. =  49    time =  1.1700e+05    mass check =  4.5813e-05
**********************************************************

Mass budget...
   air-borne    flor dep.    wall dep.    clng dep.     source       leaked
  2.4329e+01   1.9478e+04   4.8740e+02   0.0000e+00   2.0000e+04   1.1777e+01
Airborne aerosol moments...
     sigma         rad50       mdensity     ndensity     geommean      mass50
  1.7422e+00   1.7224e-06   1.3516e-04   7.4980e+09   4.8802e-15   5.9934e-14
Source moments...
     sigma         rad50       mdensity     ndensity     geommean      mass50
  2.0000e+00   5.0000e-07   0.0000e+00   0.0000e+00   1.9419e-17   1.4681e-15

**********************************************************************************

**********************************************************
step no. =  50    time =  1.2060e+05    mass check =  4.5813e-05
**********************************************************

Mass budget...
   air-borne    flor dep.    wall dep.    clng dep.     source       leaked
  2.2511e+01   1.9478e+04   4.8781e+02   0.0000e+00   2.0000e+04   1.1787e+01
Airborne aerosol moments...
     sigma         rad50       mdensity     ndensity     geommean      mass50
  1.7394e+00   1.7025e-06   1.2506e-04   7.0286e+09   4.8903e-15   5.7873e-14
Source moments...
     sigma         rad50       mdensity     ndensity     geommean      mass50
  2.0000e+00   5.0000e-07   0.0000e+00   0.0000e+00   1.9419e-17   1.4681e-15

**********************************************************************************

**********************************************************
step no. =  51    time =  1.2240e+05    mass check =  4.5814e-05
**********************************************************

Mass budget...
   air-borne    flor dep.    wall dep.    clng dep.     source       leaked
  2.1887e+01   1.9479e+04   4.8800e+02   0.0000e+00   2.0000e+04   1.1792e+01
Airborne aerosol moments...
     sigma         rad50       mdensity     ndensity     geommean      mass50
```

```
  1.7381e+00  1.6928e-08  1.2037e-04  6.8107e+09  4.6948e-15  5.6890e-14
Source moments...
     sigma        rad50      mdensity     ndensity     geommean      mass50
  2.0000e+00  5.0000e-07  0.0000e+00  0.0000e+00  1.9419e-17  1.4881e-15
Airborne mass distribution...
        m-distr          m-distr          m-distr          m-distr          m-distr
   1  9.4432e-23    2 -9.3458e-27    3  2.3197e-31    4  1.9197e-21    5  8.3180e-11
   6  2.5126e-07    7  6.1057e-08    8  3.0582e-05    9  1.4585e-05   10  7.4486e-07
  11  8.9202e-09   12  1.2785e-11   13  2.0536e-15
Airborne number distribution...
        n-distr          n-distr          n-distr          n-distr          n-distr
   1  2.3608e-02    2 -2.3384e-07    3  5.7991e-13    4  4.7993e-04    5  2.0790e+06
   6  6.2814e+08    7  1.5284e+09    8  7.6454e+08    9  3.6484e+07   10  1.8821e+05
  11  2.2301e+02   12  3.1982e-02   13  5.1340e-07
```

APPENDIX B - SUBROUTINE DESCRIPTION

We will summarize here the purpose of each subroutine in order of its appearance in the figure in Appendix C, reading left to right then top to bottom. Generally, this will only be a statement of what each subroutine does, since details can be obtained from the source listing in Appendix D. However, a more detailed description will be given when it is thought that some aspects of operation of a subroutine may not be immediately apparent.

CHARM

This is the main program. It organizes the sequence in which the other subroutines are called, the aim being to make sure that the common block variables are updated in the right order as the calculation progresses.

The values the time dependent variables take, when they are not updated continuously, correspond to the mid-point of the current time and the next time they are to be be updated. ITHHY is a counter which determines which set of interpolation formulae are to be used.

The ODE solver is allowed to integrate beyond the value of TIME and interpolate backwards as needed. Therefore, the subroutines CHARMUTH to CHARMSLN could have been called by CHARMRHS at a time not equal to TIME when the time dependent variables are updated continuously and so, in this circumstance, they are called again immediately prior to calling CHARMOUT.

When the argument to CHARMDIF equals RESET it tells the ODE solver to reinitialize its variables. This is done whenever the time dependent variables or their derivatives may change discontinuously and ensures that the solver is not required to integrate across a discontinuity.

CHARMBLO

This is a block data subroutine which sets default values for all input variables, assigns values to π, g, k and R, and defines the Fortran unit numbers of the input and output streams.

CHARMIN

This subroutine reads the input data file.

CHARMCOL

This subroutine calculates m_i for i = 1, ...n, h, e^h and $\log_e(m_1 e^{-h})$.

CHARMIND

This subroutine calculates the indexing which is later used to determine when P^i_{jk} for given i, j and k is non-zero and where it is stored.

CHARMFUN

This function subroutine calculates the function $\log_e(1-e^{-x})$. Series expansions and compiler directives to switch-off vectorization during addition of the terms in the series are used to reduce rounding errors.

CHARMCOE

This subroutine calculates all non-zero values of P^i_{jk}. A change of variable has been made in Eq. (56) $y' = y - j$ to avoid adding the result obtained from CHARMFUN, which can be small compared to unity, to j. Gauss-Legendre integration does not work when the integrand is non-smooth and so the integration range is divided into sub-ranges over which the integrand is smooth and the integral is calculated as the sum of the integrals over the sub-ranges.

SSORT

This subroutine sorts the contents of an array into ascending order. Refer to the CLAMS compendium (Ford, 1984) for instructions on how to get an abstract, documentation and a compiler listing for this subroutine.

GAUS8

This subroutine calculates the definite integral of the supplied function using Gauss-Legendre integration. Refer to the CLAMS compendium (Ford, 1984) for instructions on how to get an abstract, documentation and a compiler listing for this subroutine.

CHARMPJK

This function subroutine calculates the integrand of the production coefficient integral. The addition is done in double precision to avoid rounding errors in CHARMPJK when CHARMFE is small compared to unity.

CHARMFE

This function subroutine calculates the basic finite element. The calculations are done in double precision to avoid rounding errors in CHARMFE when it is small compared to unity.

CHARMNOR

This subroutine calculates n_{jk} for j,k = 1, ...n.

CHARMRAD

This subroutine calculates r_i for i = 1, ...n.

CHARMZLN

This subroutine calculates the initial values of Y_i for i = 1, ...n assuming the initial aerosol number distribution to be log-normal. The discretized distribution is renormalized so the airborne mass exactly equals that specified.

CHARMITH

This subroutine sets-up (Initializes) the interpolation formulae for the time dependent variables. It first calculates the time up to which each formula applies and the number of formulae and then calculates the interpolation formulae coefficients for all the time dependent variables.

CHARMWTH

This subroutine calculates a set of interpolation formulae coefficients for a given table of data values.

CHARMUTH

This subroutine calculates (Updates) the time dependent variables at the specified time using the interpolation formulae previously calculated.

CHARMGAS

This subroutine calculates ρ_g, η_g, 1, D_v, c_c, c_w and c_f. Correlations are used for η_g and D_v assuming the gas in the bulk of the cell to be pure air and the vapor in gas mixture adjacent to surfaces to be steam in air.

CHARMMOB

This subroutine calculates B_i for i = 1, ...n.

CHARMFLO

This subroutine calculates u_*, ϵ_*, δ_* and δ_{Di} for i = 1, ...n. The Fanning friction factor, which enters the equation for u_*, is determined by a transcendental equation which is solved with CO5WHE.

CO5WHE

This subroutine finds a root of the supplied function within the specified range.

CHARMFAN

The zero of this function determines the Fanning friction factor in a cylindrical pipe of any roughness at any Reynolds number. Pipes of other cross-sectional shape are dealt-with by defining a hydraulic diameter equal to the diameter of the equivalent cylindrical pipe.

CHARMAGG

This subroutine calculates K_{jk} for j,k = 1, ...n.

CHARMDEP

This subroutine calculates λ_{ci}, λ_{wi} and λ_{fi} for i = 1, ...n.

CHARMSLN

This subroutine calculates m_iS_i for i = 1, ...n. The number distribution of the source is assumed to be log-normal. The m_iS_i are renormalized so that the mass generation rate of the descretized distribution exactly equals that specified.

CHARMDIF

This subroutine sets-up the input required by DEBDF, calls DEBDF to integrate the governing equations to the specified time and checks whether the call was successful.

DEBDF

This subroutine integrates a set of coupled non-linear ODE's using the variable order backward differentiation method due to Hindmarsh (Shampine and Watts, 1979). Refer to the CLAMS compendium (Ford, 1984) for instructions on how to get an abstract, documentation and a compiler listing for this subroutine.

CHARMRHS

This subroutine calculates the time derivatives of: Y_i for i = 1, ..., the mass deposited on the floor, wall and ceiling, the mass released from the source and the leaked mass.

CHARMMOM

This subroutine calculates the moments σ, $r_{5\emptyset}$, ρ, N, m_g and $m_{5\emptyset}$ of the discretized distribution using the trapezium rule. $m_{5\emptyset}$ is determined by a transcendental equation whose root is found with CO5WHE.

CHARMM5O

This function subroutine calculates $\int_{\emptyset}^{m} mY(m,t)\ d\log_e m - \rho/2$. The trapezium rule is used to estimate the integral. The zero of CHARMM5O is the natural logarithm of $m_{5\emptyset}$.

CHARMFEO

This function subroutine calculates $\int_{\emptyset}^{m} g_i(\log_e m)\ d\log_e m$ for arbitrary m. Analytic formulae are used.

CHARMOUT

This subroutine writes results on the two output files.

APPENDIX C - SUBROUTINE HIERARCHY AND CALLING SEQUENCE

The overall organization of the computations in CHARM is discussed here to assist those who would like to modify CHARM to suit their own purposes.

CHARM is modular in the sense that the calculations are broken down into a sequence of tasks and each task is dealt-with within a subroutine devoted to that task only and nowhere else. The purpose of each subroutine has been described in the previous section. Here, we describe how the subroutines fit together.

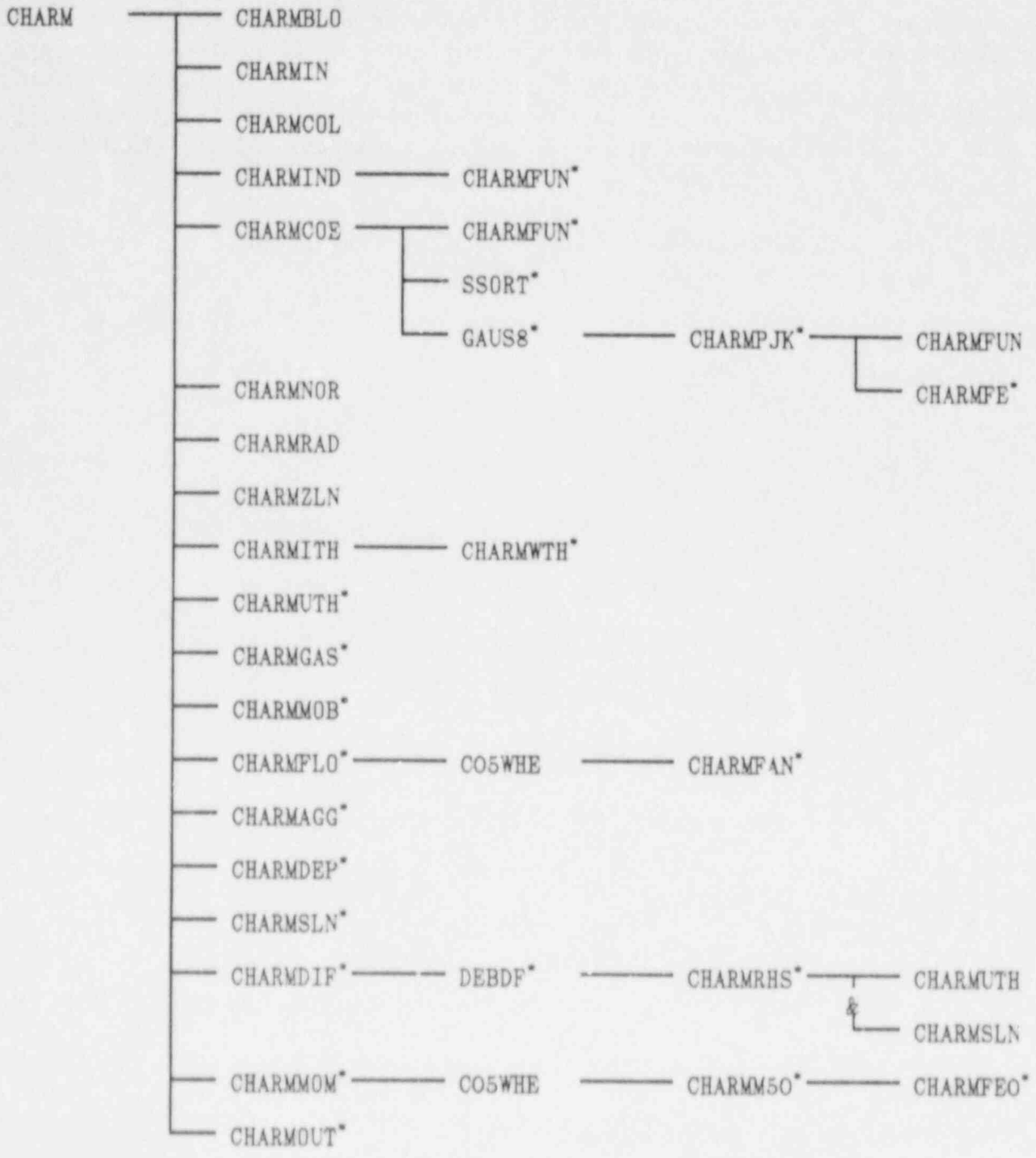
This is illustrated in the figure on the next page which shows what calls a subroutine makes to others and in what order. The logic for when and how often specific calls are made is outlined below (we indicate on the figure with an asterisk adjacent to a subroutine when it can be called more than once by its caller - the reader is referred to the source listing in Appendix D for exact details of when a subroutine is called). The callees for CHARMCOE and CHARMPJK are shown in order of their first encounter in the sequence of calls made by the callers.

Data is communicated largely by means of named common blocks; in some cases it is communicated through subroutine arguments when this is convenient or dictated by externally supplied subroutines. Each common block is designed to hold data for particular purposes so that subroutines as far as possible do not have long lists of common block variables.

Referring now to the sequence of calls shown in the figure, CHARMBLO is a block data subroutine and is shown on the sequence for completeness.

The subroutines CHARMIN to CHARMITH are called once only to calculate variables associated with the discretized form of the governing equations, the initial density distribution and coefficients in the interpolation formulae for the time dependent data.

The subroutines CHARMUTH to CHARMSLN calculate the source distribution and the agglomeration and deposition rates. These calculations depend on the external time dependent variables and therefore have to be repeated according to the specifications chosen by the user. If the user requires these quantities to be updated continuously with time then they are recalculated by CHARMRHS each time it is called. Otherwise they are recalculated by CHARM whenever the current time exceeds the previous time these variables were updated by a specified time period or equals a time when a new set of interpolation formulae for the time dependent variables apply, whichever occurs first.



* The subroutine is called more than once by the caller.
& The sequence here is the same as that between CHARMUTH and CHARMSLN called from CHARM.

The subroutine hierarchy and calling sequence

CHARMDIF is called to integrate the governing equations to the next time output is required or the next time the time dependent variables have to be updated

(when they are not updated continuously) or the next time a new set of interpolation formulae apply, whichever occurs first.

CHARMMOM is called initially (i.e. at time equals zero) and following every call to CHARMDIF, since some moments are used in the succeeding call to CHARMDIF to set tolerances.

CHARMOUT is called initially and whenever the governing equations have been integrated to a time when output is desired.

APPENDIX D - SOURCE LISTING

The subroutines are listed in order of appearance in the calling sequence shown in the figure in Appendix C, reading left to right then top to bottom.

```
      program charm(charmdat,tape4=charmdat,tty,tape5=tty                charm  2
     +                                      ,out,tape6=out)             charm  3
c                                                                        charm  4
c     This program solves the aerosol equation taking account of         charm  5
c     agglomeration, deposition and leakage of the aerosol.              charm  6
c     Collocation is used to discretize the equation.                    charm  7
c     The agglomeration kernel is calculated on a 2D mesh only           charm  8
c     so it can be updated as the thermal hydraulic conditions           charm  9
c     change with time without undue computational labor.                charm 10
c     The agglomeration and deposition rate formulae are based on those  charm 11
c     in MAEROS but have been extended to treat diffusiophoresis         charm 12
c     and turbulent deposition.                                          charm 13
c                                                                        charm 14
      common /timings/  time,istep,thhystep                              timings2
     +                 ,itime,ntime,timestep(20),timeend(20)             timings3
     +                 ,ithhy,nthhy,timethhy(20)                         timings4
     +                 ,idata,ndata,timedata(20)                         timings5
c                                                                        charm 16
      logical reset,noreset,flag                                         charm 17
      data reset,noreset/.true.,.false./                                 charm 18
      external charmblo                                                  charm 19
c                                                                        charm 20
c     Calculate time-independent variables                               charm 21
c     ************************************                               charm 22
c                                                                        charm 23
      call charmin                                                       charm 24
      call charmcol                                                      charm 25
      call charmind                                                      charm 26
      call charmcoe                                                      charm 27
      call charmnor                                                      charm 28
      call charmrad                                                      charm 29
      call charmzln                                                      charm 30
c                                                                        charm 31
c     Initialize the time dependent data                                 charm 32
c     **********************************                                 charm 33
c                                                                        charm 34
      call charmith                                                      charm 35
      ithhy=1                                                            charm 36
      timemean=min(thhystep,timethhy(ithhy))/2.e0                        charm 37
      call charmuth(timemean)                                            charm 38
      call charmgas                                                      charm 39
      call charmmob                                                      charm 40
      call charmflo                                                      charm 41
      call charmagg                                                      charm 42
      call charmdep                                                      charm 43
      call charmsln                                                      charm 44
c                                                                        charm 45
c     Calculate initial moments and print initial values                 charm 46
c     **************************************************                 charm 47
c                                                                        charm 48
      call charmmom                                                      charm 49
      istep=0                                                            charm 50
      time=0.e0                                                          charm 51
      call charmout                                                      charm 52
      istep=istep+1                                                      charm 53
c                                                                        charm 54
c     Calculate upper bound for number of calls to CHARMDIF              charm 55
c     *****************************************************              charm 56
c                                                                        charm 57
      icalls=ntime+nthhy+timeend(1)/timestep(1)                          charm 58
         if(thhystep.ne.0.e0)then                                        charm 59
         icalls=icalls+timeend(ntime)/thhystep                           charm 60
         endif                                                           charm 61
         if(ntime.gt.1)then                                              charm 62
         do 5 itime=2,ntime                                              charm 63
         icalls=icalls+(timeend(itime)-timeend(itime-1))/timestep(itime)charm 64
5        continue                                                        charm 65
         endif                                                           charm 66
      ncalls=icalls                                                      charm 67
```

```
c                                                                       charm 68
c     Loop over calls to CHARMDIF                                       charm 69
c     ***************************                                       charm 70
c                                                                       charm 71
      itime=1                                                           charm 72
      flag=reset                                                        charm 73
      do 10 icalls=1,ncalls                                             charm 74
c                                                                       charm 75
c     Find the next time to which the equations are to be integrated    charm 76
c     **************************************************************    charm 77
c                                                                       charm 78
      if(time.lt.timeend(ntime))then                                    charm 79
c                                                                       charm 80
c        Print results at this time                                     charm 81
c        **************************                                     charm 82
c                                                                       charm 83
         if(itime.eq.1)then                                             charm 84
         timelow=0.e0                                                   charm 85
         else                                                           charm 86
         timelow=timeend(itime-1)                                       charm 87
         endif                                                          charm 88
      time1=timelow                                                     charm 89
     +    +  (int((time-timelow)/timestep(itime))+1)*timestep(itime)    charm 90
      if(time1.le.time)time1=time1+timestep(itime)                      charm 91
      if(time1.gt.timeend(itime))time1=timeend(itime)                   charm 92
c                                                                       charm 93
c        Update the variables at this time                              charm 94
c        *********************************                              charm 95
c                                                                       charm 96
         if(thhystep.ne.0.e0)then                                       charm 97
            if(ithhy.eq.1)then                                          charm 98
            timelow=0.e0                                                charm 99
            else                                                        charm100
            timelow=timethhy(ithhy-1)                                   charm101
            endif                                                       charm102
         time2=timelow+(int((time-timelow)/thhystep)+1)*thhystep        charm103
         if(time2.le.time)time2=time2+thhystep                          charm104
         if(time2.gt.timethhy(ithhy))time2=timethhy(ithhy)              charm105
         else                                                           charm106
         time2=timethhy(ithhy)                                          charm107
         endif                                                          charm108
c                                                                       charm109
c     Integrate the equations to the new time                           charm110
c     ***************************************                           charm111
c                                                                       charm112
c     CHARMMOM is always called after CHARMDIF because the following    charm113
c     call to CHARMDIF needs the moments to set tolerances.             charm114
c                                                                       charm115
      time=min(time1,time2)                                             charm116
      call charmdif(flag)                                               charm117
      call charmmom                                                     charm118
c                                                                       charm119
c        Print results                                                  charm120
c        *************                                                  charm121
c                                                                       charm122
      if(time.eq.timeend(itime))itime=min(itime+1,ntime)                charm123
         if(time.eq.time1)then                                          charm124
c                                                                       charm125
c           Make sure time dependent variables                          charm126
c           are evaluated at the current time.                          charm127
c           **********************************                          charm128
c                                                                       charm129
            if(thhystep.eq.0.e0)then                                    charm130
            call charmuth(time)                                         charm131
            call charmgas                                               charm132
            call charmmob                                               charm133
            call charmflo                                               charm134
            call charmagg                                               charm135
            call charmdep                                               charm136
```

```
            call charmsln                                                    charm137
            endif                                                            charm138
         call charmout                                                       charm139
         istep=istep+1                                                       charm140
         endif                                                               charm141
c                                                                            charm142
c        Update the time dependent data if necessary                         charm143
c        *******************************************                         charm144
c                                                                            charm145
      if(time.eq.timethhy(ithhy))ithhy=min(ithhy+1,nthhy)                    charm146
         if(   time.eq.time2                                                 charm147
     +   .and. thhystep.ne.0.e0                                              charm148
     +   .and. time.lt.timeend(ntime)   )then                                charm149
         timemean=time+min(thhystep,timethhy(ithhy)-time)/2.e0               charm150
         call charmuth(timemean)                                             charm151
         call charmgas                                                       charm152
         call charmmob                                                       charm153
         call charmflo                                                       charm154
         call charmagg                                                       charm155
         call charmdep                                                       charm156
         call charmsln                                                       charm157
         endif                                                               charm158
c                                                                            charm159
c        Reset the ODE solver when the var[illegible]tinuously               charm160
c        *****************************[illegible]**********                 charm161
c                                                                            charm162
         if(time.eq.time2)then                                               charm163
         flag=reset                                                          charm164
         else                                                                charm165
         flag=noreset                                                        charm166
         endif                                                               charm167
      endif                                                                  charm168
10    continue                                                               charm169
      stop                                                                   charm170
      end                                                                    charm171
```

```
      block data charmblo                                                    chmblo 2
c                                                                            chmblo 3
c     This subroutine sets-up default data values. Phenomenological         chmblo 4
c     constants are those used in MAEROS. The data values can be            chmblo 5
c     altered by CHARMIN.                                                    chmblo 6
c                                                                            chmblo 7
      common /collpts/  ncoll,mlower,mupper,spacing,dlogem,logem0            collpts2
     +                 ,radius(100),mass(100),mobility(100)                 collpts3
      real              logem0,mass,mobility,mlower,mupper                   collpts4
c                                                                            chmblo 9
      common /aerslcon/ cshpfctr,dshpfctr,stickeff                           aerslco2
     +                 ,aknudweb,qknudweb,bknudweb                          aerslco3
     +                 ,pdensity,pthrmcon                                   aerslco4
     +                 ,kbrock,cmbrock,ctbrock                              aerslco5
      real              kbrock                                               aerslco6
c                                                                            chmblo11
      common /gasprops/ temp,press,gdensity,dynvisc,molwt,mnfrpath           gasprop2
     +                 ,gthrmcon,velocity,molwtv,diffusv                    gasprop3
     +                 ,vmfrclng,vmfrwall,vmfrflor                          gasprop4
     +                 ,vcgrclng,vcgrwall,vcgrflor                          gasprop5
     +                 ,vconclng,vconwall,vconflor                          gasprop6
      real              molwt,mnfrpath,molwtv                                gasprop7
c                                                                            chmblo13
      common /fundcon/  pi,boltzmnn,gravitat,gasconst                        fundcon2
c                                                                            chmblo15
      common /indexcoe/ nelement,hwidth,jbarmin,index(-2:100)                indexco2
     +                 ,kbarmin(-2:100),kbarmax(-2:100),nkbar(-2:100)       indexco3
c                                                                            chmblo17
      common /lognorms/ sigmazln,rad50zln,mdenzln                            lognorm2
     +                 ,ndenzln,geomzln,mas50zln                            lognorm3
      real              mdenzln,ndenzln,mas50zln                             lognorm4
c                                                                            chmblo19
      common /thrmhydr/ tgasa0(20),tgasa1(20),tgasdata(20)                   thrmhyd2
     +                 ,pgasa0(20),pgasa1(20),pgasdata(20)                  thrmhyd3
     +                 ,vgasa0(20),vgasa1(20),vgasdata(20)                  thrmhyd4
     +                 ,tclga0(20),tclga1(20),tclgdata(20)                  thrmhyd5
     +                 ,twala0(20),twala1(20),twaldata(20)                  thrmhyd6
     +                 ,tflra0(20),tflra1(20),tflrdata(20)                  thrmhyd7
     +                 ,sigsa0(20),sigsa1(20),sigsdata(20)                  thrmhyd8
     +                 ,radsa0(20),radsa1(20),radsdata(20)                  thrmhyd9
     +                 ,mdesa0(20),mdesa1(20),mdesdata(20)                  thrmhy10
      real              mdesa0,mdesa1,mdesdata                               thrmhy11
c                                                                            chmblo21
      common /timings/  time,istep,thhystep                                  timings2
     +                 ,itime,ntime,timestep(20),timeend(20)                timings3
     +                 ,ithhy,nthhy,timethhy(20)                            timings4
     +                 ,idata,ndata,timedata(20)                            timings5
c                                                                            chmblo23
      common /toleranc/ eps,eta,zeta,maxcalls,maxtrys                        toleran2
c                                                                            chmblo25
      common /ioflags/  iondis,iozrom,iocoef,ionorm                          ioflags2
     +                 ,iodepo,iomass,ioradi,iomobi                         ioflags3
     +                 ,ioaggl,iomdis,iosdis,iombal                         ioflags4
     +                 ,ioindx,iosmom,iocell,iogasp                         ioflags5
     +                 ,iotole,ioacon,ioflow                                ioflags6
c                                                                            chmblo27
      common /iotapes/  columns,ntape4,ntape5,ntape6                         iotapes2
      integer           columns                                              iotapes3
c                                                                            chmblo29
      common /cell/     areaclng,areawall,areaflor                           cell   2
     +                 ,tempclng,tempwall,tempflor                          cell   3
     +                 ,volume,leakrate                                     cell   4
     +                 ,hydrdiam,eqvrough                                   cell   5
      real              leakrate                                             cell   6
c                                                                            chmblo31
      common /flow/     blflag,vblthick,dblthick(100)                        flow   2
     +                 ,eddydiss,ustar,reynolds                             flow   3
      integer           blflag                                               flow   4
c                                                                            chmblo33
```

```
c                                                                       chmblo34
c                                                                       chmblo35
      data mlower,mupper,spacing/4.e-21,4.e-9,10.e0/                    chmblo36
      data nelement,ncoll/2,0/                                          chmblo37
c                                                                       chmblo38
      data cshpfctr,dshpfctr,stickeff/1.e0,1.e0,1.e0/                   chmblo39
      data aknudweb,qknudweb,bknudweb/1.37e0,0.4e0,1.1e0/               chmblo40
      data pdensity,pthrmcon/2.8e3,.8375e0/                             chmblo41
      data kbrock,cmbrock,ctbrock/1.e0,1.37e0,1.e0/                     chmblo42
c                                                                       chmblo43
      data molwt,gthrmcon,molwtv                                        chmblo44
     +     /28.96e0,.0255e0,18.015e0/                                   chmblo45
      data tgasdata,pgasdata,vgasdata                                   chmblo46
     +     /293.15e0,19*-1.e0,1.01325e5,19*-1.e0,0.e0,19*-1.e0/         chmblo47
      data vmfrclng,vmfrwall,vmfrflor/0.e0,0.e0,0.e0/                   chmblo48
      data vcgrclng,vcgrwall,vcgrglor/0.e0,0.e0,0.e0/                   chmblo49
c                                                                       chmblo50
      data blflag,vblthick,dblthick/0,0.e0,100*0.e0/                    chmblo51
c                                                                       chmblo53
      data pi,boltzmnn,gravitat,gasconst                                chmblo53
     +     /3.141592653e0,1.38054e-23,9.81e0,8.3143e0/                  chmblo54
c                                                                       chmblo55
      data sigmazln,rad50zln,mdenzln/2.e0,5.e-7,0.e-2/                  chmblo56
c                                                                       chmblo57
      data sigsdata,radsdata,mdesdata                                   chmblo58
     +   /2.e0,19*-1.e0,5.e-7,19*-1.e0,0.e0,19*-1.e0/                   chmblo59
c                                                                       chmblo60
      data ntime,timestep(1),timeend(1)/1,10.e0,10.e0/                  chmblo61
c                                                                       chmblo62
      data eps,eta,zeta,maxcalls,maxtrys                                chmblo63
     +     /1.e-6,0.e0,0.5e0,30,10/                                     chmblo64
c                                                                       chmblo65
      data iocell,iogasp,icacon/1,1,1/                                  chmblo66
      data iosmom,iosdis,ioflow/1,0,1/                                  chmblo67
      data iomass,ioradi,iomobi/1,1,0/                                  chmblo68
      data iodepo,ioaggl/0,0/                                           chmblo69
      data ioindx,iocoef,ionorm/0,0,0/                                  chmblo70
      data iombal,iotcle/1,0/                                           chmblo71
      data iozmom,iondis,iomdis/1,0,0/                                  chmblo72
c                                                                       chmblo73
      data columns/80/                                                  chmblo74
      data ntape4,ntape5,ntape6/4,5,6/                                  chmblo75
c                                                                       chmblo76
      data volume,leakrate/1.e5,0.e0/                                   chmblo77
      data hydrdian,eqvrough/0.e0,4.5e-5/                               chmblo78
      data areaclng,areawall,areaflor/0.e0,0.e0,0.e0/                   chmblo79
      data tclgdata,twaldata,tflrdata                                   chmblo80
     +     /293.15e0,19*-1.e0,293.15e0,19*-1.e0,293.15e0,19*-1.e0/      chmblo81
c                                                                       chmblo82
      data ndata/1/                                                     chmblo83
      data timedata/20*0.e0/                                            chmblo84
c                                                                       chmblo85
      data thhystep/1.e10/                                              chmblo86
c                                                                       chmblo87
      end                                                               chmblo88
```

```
      subroutine charmin                                                    chmin  2
c                                                                           chmin  3
c     This subroutine reads in data from the file CHARMDAT.                 chmin  4
c     List directed read is used so data items can be in free format.       chmin  5
c     Since character variables are not in the input lists, a character     chmin  6
c     placed in a line will cause it to be read as a comment card.          chmin  7
c     The back slash can be used to skip reading a line so that             chmin  8
c     default values are used. Otherwise, values read-in will               chmin  9
c     over-ride values preset in the blockdata subroutine CHARMBLO.         chmin 10
c                                                                           chmin 11
      common /collpts/  ncoll,mlower,mupper,spacing,dlogem,logem0           collpts2
     +                 ,radius(100),mass(100),mobility(100)                collpts3
      real              logem0,mass,mobility,mlower,mupper                  collpts4
c                                                                           chmin 13
      common /aerslcon/ cshpfctr,dshpfctr,stickeff                          aerslco2
     +                 ,aknudweb,qknudweb,bknudweb                         aerslco3
     +                 ,pdensity,pthrmcon                                  aerslco4
     +                 ,kbrock,cmbrock,ctbrock                             aerslco5
      real              kbrock                                              aerslco6
c                                                                           chmin 15
      common /gasprops/ temp,press,gdensity,dynvisc,molwt,mnfrpath          gasprop2
     +                 ,gthrmcon,velocity,molwtv,diffusv                   gasprop3
     +                 ,vmfrclng,vmfrwall,vmfrflor                         gasprop4
     +                 ,vcgrclng,vcgrwall,vcgrflor                         gasprop5
     +                 ,vconclng,vconwall,vconflor                         gasprop6
      real              molwt,mnfrpath,molwtv                               gasprop7
c                                                                           chmin 17
      common /indexcoe/ nelement,hwidth,jbarmin,index(-2:100)               indexco2
     +                 ,kbarmin(-2:100),kbarmax(-2:100),nkbar(-2:100)      indexco3
c                                                                           chmin 19
      common /timings/  time,istep,thhystep                                 timings2
     +                 ,itime,ntime,timestep(20),timeend(20)               timings3
     +                 ,ithhy,nthhy,timethhy(20)                           timings4
     +                 ,idata,ndata,timedata(20)                           timings5
c                                                                           chmin 21
      common /toleranc/ eps,eta,zeta,maxcalls,maxtrys                       toleran2
c                                                                           chmin 23
      common /lognorms/ sigmazln,rad50zln,mdenzln                           lognorm2
     +                 ,ndenzln,geomzln,mas50zln                           lognorm3
      real              mdenzln,ndenzln,mas50zln                            lognorm4
c                                                                           chmin 25
      common /thrmhydr/ tgasa0(20),tgasa1(20),tgasdata(20)                  thrmhyd2
     +                 ,pgasa0(20),pgasa1(20),pgasdata(20)                 thrmhyd3
     +                 ,vgasa0(20),vgasa1(20),vgasdata(20)                 thrmhyd4
     +                 ,tclga0(20),tclga1(20),tclgdata(20)                 thrmhyd5
     +                 ,twala0(20),twala1(20),twaldata(20)                 thrmhyd6
     +                 ,tflra0(20),tflra1(20),tflrdata(20)                 thrmhyd7
     +                 ,sigsa0(20),sigsa1(20),sigsdata(20)                 thrmhyd8
     +                 ,radsa0(20),radsa1(20),radsdata(20)                 thrmhyd9
     +                 ,mdesa0(20),mdesa1(20),mdesdata(20)                 thrmhy10
      real              mdesa0,mdesa1,mdesdata                              thrmhy11
c                                                                           chmin 27
      common /ioflags/  iondis,iozmom,iocoef,ionorm                         ioflags2
     +                 ,iodepo,iomass,ioradi,iomobi                        ioflags3
     +                 ,ioaggl,iomdis,iosdis,iombal                        ioflags4
     +                 ,ioindx,iosmom,iocell,iogasp                        ioflags5
     +                 ,iotole,ioacon,ioflow                               ioflags6
c                                                                           chmin 29
      common /iotapes/  columns,ntape4,ntape5,ntape6                        iotapes2
      integer           columns                                             iotapes3
c                                                                           chmin 31
      common /cell/     areaclng,areawall,areaflor                          cell   2
     +                 ,tempclng,tempwall,tempflor                         cell   3
     +                 ,volume,leakrate                                    cell   4
     +                 ,hydrdiam,eqvrough                                  cell   5
      real              leakrate                                            cell   6
c                                                                           chmin 33
      common /flow/     blflag,vblthick,dblthick(100)                       flow   2
     +                 ,eddydiss,ustar,reynolds                            flow   3
```

```
      integer           blflag                                          flow   4
c                                                                       chmin 35
      character*1 input(72)                                             chmin 36
c                                                                       chmin 37
c     Write input file to tapes NTAPE5 & NTAPE6                         chmin 38
c     *****************************************                         chmin 39
c                                                                       chmin 40
c     Tape5 is the terminal and tape6 is the output file.               chmin 41
c                                                                       chmin 42
50    read(ntape4,1000,end=60)(input(i),i=1,72)                         chmin 43
      write(ntape5,1000)(input(i),i=1,72)                               chmin 44
      write(ntape6,2000)(input(i),i=1,72)                               chmin 45
1000  format(72a1)                                                      chmin 46
2000  format(4x,72a1)                                                  chmin 47
      goto 50                                                           chmin 48
60    rewind(ntape4)                                                    chmin 49
c                                                                       chmin 50
c     Read output flags                                                 chmin 51
c     *****************                                                 chmin 52
c                                                                       chmin 53
c     A zero value for a flag means no info. is printed.                chmin 54
c     A non-zero value for a flag means print the information.          chmin 55
c     It will be printed every IOflag times (defined by ISTEP)          chmin 56
c     when the requested quantity is time dependent.                    chmin 57
c                                                                       chmin 58
1     read(ntape4,*,err= 1,end=100)iocell,ioga_p,ioacon                 chmin 59
2     read(ntape4,*,err= 2,end=100)iosmom,iosdis,ioflow                 chmin 60
3     read(ntape4,*,err= 3,end=100)iomass,ioradi,iomobi                 chmin 61
23    read(ntape4,*,err=23,end=100)iodepo,ioaggl                        chmin 62
4     read(ntape4,*,err= 4,end=100)ioindx,iocoef,ionorm                 chmin 63
5     read(ntape4,*,err= 5,end=100)iombal,iotole                        chmin 64
6     read(ntape4,*,err= 6,end=100)iozmom,iondis,iomdis                 chmin 65
c                                                                       chmin 66
c     Read step information.                                            chmin 67
c     **********************                                            chmin 68
c                                                                       chmin 69
c     This defines the time steps at which information is printed       chmin 70
c     on the output file. This allows info. to be printed more          chmin 71
c     frequently when things get interesting.                           chmin 72
c                                                                       chmin 73
7     read(ntape4,*,err= 7,end=100)ntime                                chmin 74
         if(ntime.gt.20 .or. ntime.le.0)then                            chmin 75
         write(ntape5,2001)                                             chmin 76
         write(ntape6,2001)                                             chmin 77
         stop                                                           chmin 78
2001     format(4x,'*** CHARMIN fails: NTIME is le 0 or gt 20 ***')     chmin 79
         endif                                                          chmin 80
8     read(ntape4,*,err= 8,end=100)                                     chmin 81
     +   (timestep(itime),timeend(itime),itime=1,ntime)                 chmin 82
c                                                                       chmin 83
c     Number of columns on output file.                                 chmin 84
c     *********************************                                 chmin 85
c                                                                       chmin 86
9     read(ntape4,*,err= 9,end=100)columns                              chmin 87
      if(columns.ne.80)columns=132                                      chmin 88
c                                                                       chmin 89
c     Read times at which thermal-hydraulic data is provided.           chmin 90
c     *******************************************************           chmin 91
c                                                                       chmin 92
10    read(ntape4,*,err=10,end=100)ndata                                chmin 93
         if(ndata.gt.20 .or. ndata.le.0)then                            chmin 94
         write(ntape5,2002)                                             chmin 95
         write(ntape6,2002)                                             chmin 96
         stop                                                           chmin 97
2002     format(4x,'*** CHARMIN fails: NDATA is le 0 or gt 20 ***')     chmin 98
         endif                                                          chmin 99
11    read(ntape4,*,err=11,end=100)                                     chmin100
     +   (timedata(idata),idata=1,ndata)                                chmin101
c                                                                       chmin102
```

```
c     Read how often the thermal hydraulic variables are to be updated. chmin103
c     ***************************************************************** chmin104
c                                                                         chmin105
c     If THHYSTEP = 0 then the thermal-hydraulic variables are updated  chmin106
c     during the integration of the ODE's. Otherwise they are updated   chmin107
c     every THHYSTEP seconds.                                           chmin108
c                                                                         chmin109
24    read(ntape4,*,err=24,end=100)thhystep                             chmin110
      if(thhystep.lt.0.e0)thhystep=0.e0                                 chmin111
c                                                                         chmin112
c     Read cell information.                                            chmin113
c     **********************                                            chmin114
c                                                                         chmin115
12    read(ntape4,*,err=12,end=100)areaclng,areawall,areaflor           chmin116
13    read(ntape4,*,err=13,end=100)                                     chmin117
     +   (tclgdata(idata),twaldata(idata),tflrdata(idata),idata=1,ndata)chmin118
14    read(ntape4,*,err=14,end=100)volume,leakrate                      chmin119
33    read(ntape4,*,err=33,end=100)hydrdiam,eqvrough                    chmin120
c                                                                         chmin121
c     Read information about the gas.                                   chmin122
c     *******************************                                   chmin123
c                                                                         chmin124
15    read(ntape4,*,err=15,end=100)                                     chmin125
     +   (tgasdata(idata),pgasdata(idata),vgasdata(idata),idata=1,ndata)chmin126
16    read(ntape4,*,err=16,end=100)molwt,gthrmcon,molwtv                chmin127
31    read(ntape4,*,err=31,end=100)vmfrclng,vmfrwall,vmfrflor           chmin128
32    read(ntape4,*,err=32,end=100)vcgrclng,vcgrwall,vcgrflor           chmin129
c                                                                         chmin130
c     Read information about the boundary layer thicknesses             chmin131
c     *****************************************************             chmin132
c                                                                         chmin133
30    read(ntape4,*,err=30,end=100)blflag,vblthick,dblthick(1)          chmin134
c                                                                         chmin135
c     Read information about the initial aerosol.                       chmin136
c     *******************************************                       chmin137
c                                                                         chmin138
c     It is assumed to be log-normal in C(m,t)...                       chmin139
c                                                                         chmin140
25    read(ntape4,*,err=25,end=100)sigmazln,radzOzln,mdenzln            chmin141
c                                                                         chmin142
c     Read information about the aerosol source.                        chmin143
c     ******************************************                        chmin144
c                                                                         chmin145
c     This is treated in the same way as the thermal-hydraulic data.    chmin146
c                                                                         chmin147
17    read(ntape4,*,err=17,end=100)                                     chmin148
     +   (sigsdata(idata),radsdata(idata),mdesdata(idata),idata=1,ndata)chmin149
c                                                                         chmin150
c     Read information about the collocation points.                    chmin151
c     **********************************************                    chmin152
c                                                                         chmin153
c     If NCOLL is zero then it is calculated from SPACING etc.          chmin154
c     otherwise the input value of SPACING is ignored.                  chmin155
c                                                                         chmin156
18    read(ntape4,*,err=18,end=100)ncoll,spacing                        chmin157
19    read(ntape4,*,err=19,end=100)mlower,mupper                        chmin158
20    read(ntape4,*,err=20,end=100)nelement                             chmin159
c                                                                         chmin160
c     Read tolerance information.                                       chmin161
c     ***************************                                       chmin162
c                                                                         chmin163
21    read(ntape4,*,err=21,end=100)eps,maxcalls,maxtrys                 chmin164
22    read(ntape4,*,err=22,end=100)eta,zeta                             chmin165
c                                                                         chmin166
c     Read new aerosol physics data.                                    chmin167
c     ******************************                                    chmin168
c                                                                         chmin169
26    read(ntape4,*,err=26,end=100)pdensity,pthrmcon                    chmin170
27    read(ntape4,*,err=27,end=100)cshpfctr,dshpfctr,stickeff           chmin171
```

```
28    read(ntape4,*,err=28,end=100)aknudweb,qknudweb,bknudweb          chmin172
29    read(ntape4,*,err=29,end=100)kbrock,cmbrock,ctbrock              chmin173
      return                                                           chmin174
c                                                                      chmin175
c     End of file read - give a warning and carry on                   chmin176
c     ******************************************                       chmin177
c                                                                      chmin178
100   write(ntape5,2003)                                               chmin179
      write(ntape6,2003)                                               chmin180
2003  format(4x,'*** CHARMIN warning: end-of-file read - could be'     chmin181
     +,' an error in the data ***')                                    chmin182
      return                                                           chmin183
      end                                                              chmin184
```

```
      subroutine charmcol                                                   chmcol 2
c                                                                           chmcol 3
c     This subroutine calculates the aerosol particle masses which          chmcol 4
c     will serve as collocation points in the subsequent calculations.      chmcol 5
c                                                                           chmcol 6
      common /collpts/  ncoll,mlower,mupper,spacing,dlogem,logem0           collpts2
     +                 ,radius(100),mass(100),mobility(100)                collpts3
      real              logem0,mass,mobility,mlower,mupper                  collpts4
c                                                                           chmcol 8
c     Set-up parameters which define the collocation points                 chmcol 9
c     *****************************************************                 chmcol10
c                                                                           chmcol11
      if(ncoll.eq.0)then                                                    chmcol12
         dummy=alog(mupper/mlower)/alog(spacing)+1.e0                       chmcol13
         ncoll=int(dummy)                                                   chmcol14
         if(float(ncoll).lt.dummy)ncoll=ncoll+1                             chmcol15
         if(ncoll.gt.100)ncoll=100                                          chmcol16
      else                                                                  chmcol17
         if(ncoll.gt.100)ncoll=100                                          chmcol18
         spacing=(mupper/mlower)**(1.e0/float(ncoll-1))                     chmcol19
      endif                                                                 chmcol20
      mass(1)=mlower                                                        chmcol21
      dlogem = alog( spacing )                                              chmcol22
      logem0 = alog( mass(1) ) - dlogem                                     chmcol23
c                                                                           chmcol24
c     Calculate the collocation points                                      chmcol25
c     ********************************                                      chmcol26
c                                                                           chmcol27
      do 100 icoll=2,ncoll                                                  chmcol28
      mass(icoll)=mass(icoll-1)*spacing                                     chmcol29
100   continue                                                              chmcol30
      mupper=mass(ncoll)                                                    chmcol31
      return                                                                chmcol32
      end                                                                   chmcol33
```

```
      subroutine charmind                                                chmind 2
c                                                                        chmind 3
c     This subroutine sets-up the indexing to the terms on the rhs.      chmind 4
c     For given i and j with 1 <= i <= ncoll and 1 <= j <= i then        chmind 5
c     values of k are found such that gk(mi-m) * gj(m) for 0 < m < mi    chmind 6
c     is non-zero.                                                       chmind 7
c     gj and gk are the j-th and k-th finite elements and mi is the      chmind 8
c     mass at the i-th collocation point.                                chmind 9
c                                                                        chmind10
c     Note that the indexing depends only on i-j (jbar) and i-k (kbar).  chmind11
c                                                                        chmind12
      common /collpts/  ncoll,mlower,mupper,spacing,dlogem,logem0        collpts2
     +                 ,radius(100),mass(100),mobility(100)             collpts3
      real              logem0,mass,mobility,mlower,mupper               collpts4
c                                                                        chmind14
      common /indexcoe/ nelement,hwidth,jbarmin,index(-2:100)            indexco2
     +                 ,kbarmin(-2:100),kbarmax(-2:100),nkbar(-2:100)   indexco3
c                                                                        chmind16
      common /iotapes/  columns,ntape4,ntape5,ntape6                     iotapes2
      integer           columns                                          iotapes3
c                                                                        chmind18
c     Calculate the element half-width                                   chmind19
c     ********************************                                   chmind20
c                                                                        chmind21
      if(nelement.eq.1)hwidth=0.5e0                                      chmind22
      if(nelement.eq.2)hwidth=1.e0                                       chmind23
      if(nelement.eq.3 .or. nelement.eq.4)hwidth=2.e0                     chmind24
      if(nelement.eq.5 .or. nelement.eq.6)hwidth=3.e0                     chmind25
c                                                                        chmind26
c     Calculate the minimum value of jbar                                chmind27
c     ***********************************                                chmind28
c                                                                        chmind29
      jbarmin = -int(hwidth)                                             chmind30
      if(float(jbarmin) .le. -hwidth)jbarmin=jbarmin+1                   chmind31
c                                                                        chmind32
c     Calculate indexing for i=ncoll                                     chmind33
c     ******************************                                     chmind34
c                                                                        chmind35
      if(jbarmin.le.ncoll-1)then                                         chmind36
      do 100 jbar=jbarmin,ncoll-1                                        chmind37
      sjbar=charmfun((jbar+hwidth)*dlogem)/dlogem                        chmind38
      kbarmin(jbar) = - int(sjbar+hwidth)                                chmind39
      if( kbarmin(jbar)+hwidth .le. -sjbar )                             chmind40
     +          kbarmin(jbar) = kbarmin(jbar) + 1                        chmind41
      if(kbarmin(jbar).lt.jbarmin)kbarmin(jbar)=jbarmin                  chmind42
         if(float(jbar).le.hwidth)then                                   chmind43
         kbarmax(jbar)=ncoll-1                                           chmind44
         else                                                            chmind45
         sjbar=charmfun((jbar-hwidth)*dlogem)/dlogem                     chmind46
         kbarmax(jbar) = - int(sjbar-hwidth)                             chmind47
         if( kbarmax(jbar)-hwidth .ge. -sjbar )                          chmind48
     +          kbarmax(jbar) = kbarmax(jbar) - 1                        chmind49
         if(kbarmax(jbar).gt.ncoll-1)kbarmax(jbar)=ncoll-1               chmind50
         endif                                                           chmind51
      nkbar(jbar) = kbarmax(jbar) - kbarmin(jbar) + 1                    chmind52
      if(nkbar(jbar).lt.0)nkbar(jbar)=0                                  chmind53
      if(jbar.eq.jbarmin)index(jbar)=1                                   chmind54
      if(jbar.gt.jbarmin)index(jbar)=index(jbar-1) + nkbar(jbar-1)       chmind55
100   continue                                                           chmind56
c                                                                        chmind57
c     Check number of non-zero values                                    chmind58
c     *******************************                                    chmind59
c                                                                        chmind60
      nonzero = index(ncoll-1) + nkbar(ncoll-1) - 1                      chmind61
      index(ncoll)=nonzero                                               chmind62
         if(nonzero.gt.300)then                                          chmind63
         write(ntape5,1000)nonzero                                       chmind64
         write(ntape6,1000)nonzero                                       chmind65
1000     format(4x,'*** CHARMIND fails: NONZERO is ',i3,' ***')          chmind66
```

```
            stop                                                        chmind67
            endif                                                       chmind68
c                                                                       chmind69
c       Translate index with respect to kbarmin                         chmind70
c       ***************************************                         chmind71
c                                                                       chmind72
        do 110 jbar=jbarmin,ncoll-1                                     chmind73
        index(jbar)=index(jbar)-kbarmin(jbar)                           chmind74
110     continue                                                        chmind75
        endif                                                           chmind76
        return                                                          chmind77
        end                                                             chmind78
```

```
      function charmfun(x)                                                  chmfun 2
c                                                                           chmfun 3
c     This function subroutine calculates the function alog(1.-exp(-x)).chmfun 4
c     A series expansion is used when x is >= alog(10.) or x <= .1        chmfun 5
c     to avoid rounding errors.                                           chmfun 6
c                                                                         chmfun 7
      dimension term(14)                                                  chmfun 8
      data cutoff/0.1e0/                                                  chmfun 9
c                                                                         chmfun10
c     Switch-off vectorization and expand the exponential if x is < .1    chmfun11
c     ****************************************************************  chmfun12
c                                                                         chmfun13
c$dir novector                                                            chmfun14
      if(x.lt.cutoff)then                                                 chmfun15
         term(1)=x                                                        chmfun16
         do 30 i=2,14                                                     chmfun17
         term(i)=-term(i-1)*x/float(i)                                    chmfun18
30       continue                                                         chmfun19
         sum=0.e0                                                         chmfun20
         do 40 i=14,1,-1                                                  chmfun21
         sum=sum+term(i)                                                  chmfun22
40       continue                                                         chmfun23
         charmfun=alog(sum)                                               chmfun24
      else                                                                chmfun25
c                                                                         chmfun26
c        Expand the logarithm if x is large enough ie if y = exp(-x) < .chmfun27
c        ****************************************************************chmfun28
c                                                                         chmfun29
         y=exp(-x)                                                        chmfun30
         if(y.lt.cutoff)then                                              chmfun31
            term(1)=y                                                     chmfun32
            do 10 i=2,14                                                  chmfun33
            term(i)=term(i-1)*y                                           chmfun34
10          continue                                                      chmfun35
            sum=0.e0                                                      chmfun36
            do 20 i=14,1,-1                                               chmfun37
            sum=sum+term(i)/float(i)                                      chmfun38
20          continue                                                      chmfun39
            charmfun=-sum                                                 chmfun40
         else                                                             chmfun41
c                                                                         chmfun42
c           Use standard functions and switch-on vectorization            chmfun43
c           ************************************************              chmfun44
c                                                                         chmfun45
            charmfun=alog(1.e0-y)                                         chmfun46
         endif                                                            chmfun47
      endif                                                               chmfun48
c$dir vector                                                              chmfun49
      return                                                              chmfun50
      end                                                                 chmfun51
```

```
      subroutine charmcoe                                                chmcoe 2
c                                                                        chmcoe 3
c     This subroutine calculates all non-zero values of the integral     chmcoe 4
c     of gk(mi-m) * gj(m) w.r.t. loge(m) for the range 0 < m < mi.       chmcoe 5
c     These depend only on i-j and i-k and are stored using the          chmcoe 6
c     indexing developed in charmind.                                    chmcoe 7
c                                                                        chmcoe 8
      common /collpts/  ncoll,mlower,mupper,spacing,dlogem,logem0        collpts2
     +                  ,radius(100),mass(100),mobility(100)            collpts3
      real              logem0,mass,mobility,mlower,mupper               collpts4
c                                                                        chmcoe10
      common /indexcoe/ nelement,hwidth,jbarmin,index(-2:100)            indexco2
     +                 ,kbarmin(-2:100),kbarmax(-2:100),nkbar(-2:100)   indexco3
c                                                                        chmcoe12
      common /coef/     pijk(300),njk(100,100)                           coef   2
      real              njk                                              coef   3
c                                                                        chmcoe14
      common /toleranc/ eps,eta,zeta,maxcalls,maxtrys                    toleran2
c                                                                        chmcoe16
      common /jandkbar/ jbar,kbar                                        jandkba2
c                                                                        chmcoe18
      common /iotapes/  columns,ntape4,ntape5,ntape6                     iotapes2
      integer           columns                                          iotapes3
c                                                                        chmcoe20
      dimension xlimit(14),dummy(14)                                     chmcoe21
      external charmpjk                                                  chmcoe22
c                                                                        chmcoe23
c     Set all pijk to zero                                               chmcoe24
c     ********************                                               chmcoe25
c                                                                        chmcoe26
      do 150 ind=1,300                                                   chmcoe27
      pijk(ind)=0.e0                                                     chmcoe28
150   continue                                                           chmcoe29
c                                                                        chmcoe30
c     Calculate all non-zero values of pijk                              chmcoe31
c     *************************************                              chmcoe32
c                                                                        chmcoe33
      i=ncoll                                                            chmcoe34
      if(i-jbarmin.ge.1)then                                             chmcoe35
      do 210 j=1,i-jbarmin                                               chmcoe36
      jbar=i-j                                                           chmcoe37
         if(nkbar(jbar).ne.0)then                                        chmcoe38
         do 220 kbar=kbarmin(jbar),kbarmax(jbar)                         chmcoe39
         k=i-kbar                                                        chmcoe40
c                                                                        chmcoe41
c        Find integration range                                          chmcoe42
c        **********************                                          chmcoe43
c                                                                        chmcoe44
         if(kbar-hwidth.le.0.e0)then                                     chmcoe45
         ylower=-hwidth-jbar                                             chmcoe46
         else                                                            chmcoe47
         ylower=max(-hwidth-jbar,charmfun((kbar-hwidth)*dlogem)/dlogem)  chmcoe48
         endif                                                           chmcoe49
         yupper=min( hwidth-jbar,charmfun((kbar+hwidth)*dlogem)/dlogem   chmcoe50
     +              ,0.e0 )                                              chmcoe51
            if(ylower.ge.yupper)then                                     chmcoe52
            write(ntape5,1000)                                           chmcoe53
            write(ntape6,1000)                                           chmcoe54
1000        format(4x,'*** CHARMCOE fails: YLOWER is ge YUPPER ***')     chmcoe55
            stop                                                         chmcoe56
            endif                                                        chmcoe57
c                                                                        chmcoe58
c        No need for sub-ranges if NELEMENT = 1                          chmcoe59
c        **************************************                          chmcoe60
c                                                                        chmcoe61
         if(nelement.eq.1)then                                           chmcoe62
         call gaus8(charmpjk,ylower,yupper,eps,answer,ierror)            chmcoe63
            if(ierror.ne.1)then                                          chmcoe64
            write(ntape5,1010)ierror                                     chmcoe65
```

```
            write(ntape6,1010)ierror                                    chmcoe66
1010        format(4x,'*** CHARMCOE warning: IERROR is ',i2,' ***')     chmcoe67
            endif                                                       chmcoe68
          pijk(index(jbar)+kbar)=answer                                 chmcoe69
          else                                                          chmcoe70
c                                                                       chmcoe71
c         Split integration range into sub-ranges                       chmcoe72
c         ***************************************                       chmcoe73
c                                                                       chmcoe74
          nlimit=2*int(hwidth)+1                                        chmcoe75
          do 80 klimit=1,nlimit                                         chmcoe76
          ylimit=-hwidth+float(klimit-1)                                chmcoe77
          xlimit(nlimit+klimit)=ylimit-jbar                             chmcoe78
            if(kbar+ylimit.gt.C.e0)then                                 chmcoe79
            xlimit(klimit)=charmfun((kbar+ylimit)*dlogem)/dlogem        chmcoe80
            else                                                        chmcoe81
            xlimit(klimit)=-hwidth-jbar                                 chmcoe82
            endif                                                       chmcoe83
80        continue                                                      chmcoe84
          call ssort(xlimit,dummy,2*nlimit,1)                           chmcoe85
c                                                                       chmcoe86
c         Integrate between successive limits                           chmcoe87
c         ***********************************                           chmcoe88
c                                                                       chmcoe89
          sum=0.e0                                                      chmcoe90
          do 100 klimit=1,2*nlimit-1                                    chmcoe91
          xlower=xlimit(klimit)                                         chmcoe92
          xupper=xlimit(klimit+1)                                       chmcoe93
          if(    xlower.lt.xupper                                       chmcoe94
     +    .and. xlower.ge.ylower                                        chmcoe95
     +    .and. xupper.le.yupper )then                                  chmcoe96
          call gaus8(charmpjk,xlower,xupper,eps,answer,ierror)          chmcoe97
            if(ierror.ne.1)then                                         chmcoe98
            write(ntape5,1010)ierror                                    chmcoe99
            write(ntape6,1010)ierror                                    chmco100
            endif                                                       chmco101
          sum = sum + answer                                            chmco102
          endif                                                         chmco103
100       continue                                                      chmco104
          pijk(index(jbar)+kbar)=sum                                    chmco105
          endif                                                         chmco106
220       continue                                                      chmco107
          endif                                                         chmco108
210     continue                                                        chmco109
        endif                                                           chmco110
        return                                                          chmco111
        end                                                             chmco112
```

```
      function charmpjk(arg)                                                    chmpjk 2
c                                                                               chmpjk 3
c     This function subroutine calculates the integrand in the                  chmpjk 4
c     integral required for the production coefficient.                         chmpjk 5
c                                                                               chmpjk 6
      common /collpts/  ncoll,mlower,mupper,spacing,dlogem,logem0               collpts2
     +                 ,radius(100),mass(100),mobility(100)                     collpts3
      real              logem0,mass,mobility,mlower,mupper                      collpts4
c                                                                               chmpjk 8
      common /jandkbar/ jbar,kbar                                               jandkba2
c                                                                               chmpjk10
      charmpjk = charmfe(dble(arg)+jbar,0)                                      chmpjk11
     +    *charmfe(dble(charmfun(-arg*dlogem)/dlogem)+kbar,0)                   chmpjk12
      return                                                                    chmpjk13
      end                                                                       chmpjk14
```

```
      function charmfe(arg,k)                                                   chmfe  2
c                                                                               chmfe  3
c     This function subroutine calculates the value of one from a               chmfe  4
c     choice of several finite elements at ARG. The elements                    chmfe  5
c     herein calculated are centered with respect to ARG=0, are                 chmfe  6
c     symmetric and have unit spacing. ARG is therefore scaled                  chmfe  7
c     by the collocation interval (the points are equally spaced)               chmfe  8
c     and translated, assuming the argument to be mass.                         chmfe  9
c     i.e. for the k-th element the following transformation is                 chmfe 10
c     performed: x = (loge(m) - loge(m0))/dlogem - k.                           chmfe 11
c     The transformation is not done when k = 0.                                chmfe 12
c                                                                               chmfe 13
c     The choice of element is determined by NELEMENT                           chmfe 14
c                                                                               chmfe 15
      double precision arg,x,y,z                                                chmfe 16
c                                                                               chmfe 17
      common /collpts/  ncoll,mlower,mupper,spacing,dlogem,logem0               collpts2
     +                 ,radius(100),mass(100),mobility(100)                     collpts3
      real              logem0,mass,mobility,mlower,mupper                      collpts4
c                                                                               chmfe 19
      common /indexcoe/ nelement,hwidth,jbarmin,index(-2:100)                   indexco2
     +                 ,kbarmin(-2:100),kbarmax(-2:100),nkbar(-2:100)           indexco3
c                                                                               chmfe 21
c     Transform the argument and select an element                              chmfe 22
c     ********************************************                              chmfe 23
c                                                                               chmfe 24
      charmfe = 0.e0                                                            chmfe 25
      if(k.eq.0) x = abs( arg )                                                 chmfe 26
      if(k.ne.0) x = abs( (log(arg) - logem0)/dlogem - k )                      chmfe 27
      goto(1,2,3,4,5,6)nelement                                                 chmfe 28
c                                                                               chmfe 29
c     First order element                                                       chmfe 30
c     *******************                                                       chmfe 31
c                                                                               chmfe 32
1     continue                                                                  chmfe 33
      if(x.le.0.5d0)charmfe=1.e0                                                chmfe 34
      return                                                                    chmfe 35
c                                                                               chmfe 36
c     Second order element                                                      chmfe 37
c     ********************                                                      chmfe 38
c                                                                               chmfe 39
2     continue                                                                  chmfe 40
      if(x.lt.1.d0)charmfe=1.d0-x                                               chmfe 41
      return                                                                    chmfe 42
c                                                                               chmfe 43
c     Third order element                                                       chmfe 44
c     *******************                                                       chmfe 45
c                                                                               chmfe 46
3     continue                                                                  chmfe 47
      y=2.d0-x                                                                  chmfe 48
      if(x.lt.2.d0 .and. x.gt.1.d0)                                             chmfe 49
     +      charmfe=(1.d0-x)*y*y/2.d0                                           chmfe 50
      if(x.lt.1.d0)                                                             chmfe 51
     +      charmfe=(2.d0-5.d0*x*x+3.d0*x*x*x)/2.d0                             chmfe 52
      return                                                                    chmfe 53
c                                                                               chmfe 54
c     Fourth order element                                                      chmfe 55
c     ********************                                                      chmfe 56
c                                                                               chmfe 57
4     continue                                                                  chmfe 58
      y.2.d0-x                                                                  chmfe 59
      if(x.lt.2.d0 .and. x.gt.1.d0)                                             chmfe 60
     +      charmfe=-(1.d0-x)*y*y*y*(1.d0-2.d0*x)/2.d0                          chmfe 61
      if(x.lt.1.d0)                                                             chmfe 62
     +      charmfe=(1.d0-x)*(1.d0+x-4.5e0*x*x*x+3.d0*x*x*x*x)                  chmfe 63
      return                                                                    chmfe 64
c                                                                               chmfe 65
c     Fifth order element                                                       chmfe 66
c     *******************                                                       chmfe 67
```

```
c                                                                               chmfe 68
5     continue                                                                  chmfe 69
      y=2.d0-x                                                                  chmfe 70
      z=3.d0-x                                                                  chmfe 71
      if(x.lt.3.d0 .and. x.gt.2.d0)                                             chmfe 72
     +      charmfe=y*y*y*z*z*z*z/12.d0                                        chmfe 73
      if(x.lt.2.d0 .and. x.gt.1.d0)                                             chmfe 74
     +      charmfe=-y*y*y*(1.d0-x)                                            chmfe 75
     +      *(1.d0-61.d0*y+89.d0*y*y-35.d0*y*y*y)/12.d0                        chmfe 76
      if(x.lt.1.d0)                                                             chmfe 77
     +      charmfe=1.d0-x*x-91.d0*x*x*x*x/8.d0                                chmfe 78
     +      +38.d0*x*x*x*x*x-32.d0*x*x*x*x*x*x+55.d0*x*x*x*x*x*x*x/8.d0        chmfe 79
      return                                                                    chmfe 80
c                                                                               chmfe 81
c     Sixth order element                                                       chmfe 82
c     *******************                                                       chmfe 83
c                                                                               chmfe 84
6     continue                                                                  chmfe 85
      y=2.d0-x                                                                  chmfe 86
      z=3.d0-x                                                                  chmfe 87
      if(x.lt.3.d0 .and. x.gt.2.d0)                                             chmfe 88
     +      charmfe=-z*z*z*z*z*y*y*y*(16.d0-9.d0*x)/24.d0                      chmfe 89
      if(x.lt.2.d0 .and. x.gt.1.d0)                                             chmfe 90
     +      charmfe=-y*y*y*(1.d0-x)*(2.d0+3.d0*y-432.d0*y*y                    chmfe 91
     +        +1010.d0*y*y*y-830.d0*y*y*y*y+235.d0*y*y*y*y*y)/24.d0            chmfe 92
      if(x.lt.1.d0)                                                             chmfe 93
     +      charmfe=(12.d0-12.d0*x*x+3.d0*x*x*x*x-659.d0*x*x*x*x*x             chmfe 94
     +      +2226.d0*x*x*x*x*x*x-2880.d0*x*x*x*x*x*x*x                         chmfe 95
     +      +1885.d0*x*x*x*x*x*x*x*x-375.d0*x*x*x*x*x*x*x*x*x)/12.d0           chmfe 96
      return                                                                    chmfe 97
      end                                                                       chmfe 98
```

```
      subroutine charmnor                                                    chmnor 2
c                                                                            chmnor 3
c     This subroutine calculates a normalization for the agglomeration       chmnor 4
c     terms which guarantees mass conservation.                              chmnor 5
c     The production term must be multiplied by the normalization if         chmnor 6
c     it is non-zero, otherwise the corresponding destruction term           chmnor 7
c     must be set to zero.                                                   chmnor 8
c                                                                            chmnor 9
      common /collpts/  ncoll,mlower,mupper,spacing,dlogem,logem0            collpts2
     +                 ,radius(100),mass(100),mobility(100)                  collpts3
      real              logem0,mass,mobility,mlower,mupper                   collpts4
c                                                                            chmnor11
      common /indexcoe/ nelement,hwidth,jbarmin,index(-2:100)                indexco2
     +                 ,kbarmin(-2:100),kbarmax(-2:100),nkbar(-2:100)        indexco3
c                                                                            chmnor13
      common /coef/     pijk(300),njk(100,100)                               coef   2
      real              njk                                                  coef   3
c                                                                            chmnor15
c     Loop over j and k: the normalization is symmetric so k <= j.           chmnor16
c     ************************************************************           chmnor17
c                                                                            chmnor18
      do 100 j=1,ncoll                                                       chmnor19
      do 200 k=1,j                                                           chmnor20
      sum=0.e0                                                               chmnor21
      do 300 i=1,ncoll                                                       chmnor22
      jbar=i-j                                                               chmnor23
      kbar=i-k                                                               chmnor24
c                                                                            chmnor25
c     Extract Pijk                                                           chmnor26
c     ************                                                           chmnor27
c                                                                            chmnor28
      cijk=0.e0                                                              chmnor29
         if(jbar.ge.jbarmin)then                                             chmnor30
         if(kbar.ge.kbarmin(jbar) .and. kbar.le.kbarmax(jbar))               chmnor31
     +   cijk=pijk(index(jbar)+kbar)                                         chmnor32
         endif                                                               chmnor33
c                                                                            chmnor34
c     Extract Pikj                                                           chmnor35
c     ************                                                           chmnor36
c                                                                            chmnor37
      cikj=0.e0                                                              chmnor38
         if(kbar.ge.jbarmin)then                                             chmnor39
         if(jbar.ge.kbarmin(kbar) .and. jbar.le.kbarmax(kbar))               chmnor40
     +   cikj=pijk(index(kbar)+jbar)                                         chmnor41
         endif                                                               chmnor42
c                                                                            chmnor43
c     Add to the normalization sum                                           chmnor44
c     ****************************                                           chmnor45
c                                                                            chmnor46
      sum = sum + mass(i) * ( cijk + cikj )                                  chmnor47
300   continue                                                               chmnor48
c                                                                            chmnor49
c     Calculate the normalization                                            chmnor50
c     ***************************                                            chmnor51
c                                                                            chmnor52
      if(sum.eq.0.e0)njk(j,k)=0.e0                                           chmnor53
      if(sum.ne.0.e0)njk(j,k)=(mass(j)+mass(k))/sum                          chmnor54
      njk(k,j)=njk(j,k)                                                      chmnor55
200   continue                                                               chmnor56
100   continue                                                               chmnor57
      return                                                                 chmnor58
      end                                                                    chmnor59
```

```
      subroutine charmrad                                                   chmrad 2
c                                                                           chmrad 3
c     This subroutine calculates particle radii at the collocation          chmrad 4
c     points.                                                               chmrad 5
c                                                                           chmrad 6
      common /collpts/  ncoll,mlower,mupper,mspacing,dlogem,logem0          collpts2
     +                 ,radius(100),mass(100),mobility(100)                collpts3
      real              logem0,mass,mobility,mlower,mupper                  collpts4
c                                                                           chmrad 8
      common /aerslcon/ cshpfctr,dshpfctr,stickeff                          aerslco2
     +                 ,aknudweb,qknudweb,bknudweb                         aerslco3
     +                 ,pdensity,pthrmcon                                  aerslco4
     +                 ,kbrock,cmbrock,etbrock                             aerslco5
      real              kbrock                                              aerslco6
c                                                                           chmrad10
      common /fundcon/  pi,boltzmnn,gravitat,gasconst                       fundcon2
c                                                                           chmrad12
c     The radius defined here is an equivalent radius, which is the         chmrad13
c     radius of a spherical particle of equal mass. Non-sphericity is       chmrad14
c     taken into account via correction factors (called shape factors).     chmrad15
c                                                                           chmrad16
      factor = 4.e0 * pi * pdensity / 3.e0                                  chmrad17
      do 100 icoll=1,ncoll                                                  chmrad18
      radius(icoll) = ( mass(icoll) / factor )**(1.e0/3.e0)                 chmrad19
100   continue                                                              chmrad20
      return                                                                chmrad21
      end                                                                   chmrad22
```

```
      subroutine charmzln                                                   chmzln 2
c                                                                           chmzln 3
c     This subroutine initializes the number density distribution.          chmzln 4
c                                                                           chmzln 5
c     It takes the initial distribution to be log-normal. The three         chmzln 6
c     input parameters are the cube root of the geometric mass              chmzln 7
c     standard deviation (in keeping with the normal convention),           chmzln 8
c     sigmazln (no units), the mass median radius, rad50zln (m),            chmzln 9
c     and the total mass density, mdenzln (kg m**-3).                       chmzln10
c                                                                           chmzln11
c     The three parameters of the log-normal distribution are the total     chmzln12
c     number density, ndenzln (m**-3), the geometric mean mass,             chmzln13
c     geomzln (kg), and the logarithm of the geometric mass standard        chmzln14
c     deviation, logsigma (no units).                                       chmzln15
c                                                                           chmzln16
c     The discretized distribution is stored as the number density          chmzln17
c     times mass, since this is the chosen dependent variable of            chmzln18
c     the aerosol equation.                                                 chmzln19
c                                                                           chmzln20
      common /fundcon/  pi,boltzmnn,gravitat,gasconst                       fundcon2
c                                                                           chmzln22
      common /aerslcon/ cshpfctr,dshpfctr,stickeff                          aerslco2
     +                 ,aknudweb,qknudweb,bknudweb                        aerslco3
     +                 ,pdensity,pthrmcon                                 aerslco4
     +                 ,kbrock,cmbrock,ctbrock                            aerslco5
      real              kbrock                                              aerslco6
c                                                                           chmzln24
      common /collpts/  ncoll,mlower,mupper,spacing,dlogem,logem0           collpts2
     +                 ,radius(100),mass(100),mobility(100)                collpts3
      real              logem0,mass,mobility,mlower,mupper                  collpts4
c                                                                           chmzln26
      common /distrib/  zstore(105)                                         distrib2
c                                                                           chmzln28
      common /lognormz/ sigmazln,rad50zln,mdenzln                           lognorm2
     +                 ,ndenzln,geomzln,mas50zln                          lognorm3
      real              mdenzln,ndenzln,mas50zln                            lognorm4
c                                                                           chmzln30
      real logsigma,mdensity                                                chmzln31
c                                                                           chmzln32
c     Convert the input parameters to the log-normal parameters             chmzln33
c     ********************************************************              chmzln34
c                                                                           chmzln35
      logsigma = 3.e0*alog( sigmazln )                                      chmzln36
      mas50zln=4.e0*pi*pdensity*rad50zln*rad50zln*rad50zln/3.e0             chmzln37
      geomzln=mas50zln*exp(-logsigma*logsigma)                              chmzln38
      ndenzln=mdenzln*exp(-logsigma*logsigma/2.e0)/geomzln                  chmzln39
c                                                                           chmzln40
c     Calculate the distribution at the collocation points and the          chmzln41
c     total mass concentration density of the discretized distribution      chmzln42
c     ****************************************************************      chmzln43
c                                                                           chmzln44
      mdensity=0.e0                                                         chmzln45
      const=ndenzln/( sqrt(2.e0*pi) * logsigma )                            chmzln46
      do 100 icoll=1,ncoll                                                  chmzln47
      exponent=alog( mass(icoll)/geomzln ) / logsigma                       chmzln48
      zstore(icoll)=const*exp(-exponent*exponent/2.e0)                      chmzln49
      mdensity=mdensity+dlogem*zstore(icoll)*mass(icoll)                    chmzln50
100   continue                                                              chmzln51
c                                                                           chmzln52
c     Renormalize so that no mass is lost                                   chmzln53
c     ***********************************                                   chmzln54
c                                                                           chmzln55
      if(mdensity.ne.0.e0)then                                              chmzln56
      renorm=mdenzln/mdensity                                               chmzln57
      do 200 icoll=1,ncoll                                                  chmzln58
      zstore(icoll)=zstore(icoll)*renorm                                    chmzln59
200   continue                                                              chmzln60
      endif                                                                 chmzln61
      return                                                                chmzln62
```

```
end                                                    .hmsln63
```

```
      subroutine charmith                                                chmith 2
c                                                                        chmith 3
c     This subroutine translates the data for the time dependent         chmith 4
c     variables into a form which can be used by CHARMUTH.               chmith 5
c     It is assumed that the times for the data points are in            chmith 6
c     non-decreasing order. Consecutive times can be equal to allow the  chmith 7
c     variables to change discontinuously.                               chmith 8
c     The interpolation formulae are of the form: a0 + a1 * time.        chmith 9
c     a1 is zero (i.e. constant extrapolation is used) when TIME lies    chmith10
c     outside the range of the times of the data.                        chmith11
c                                                                        chmith12
      common /timings/  time,istep,thhystep                              timings2
     +                 ,itime,ntime,timestep(20),timeend(20)             timings3
     +                 ,ithhy,nthhy,timethhy(20)                         timings4
     +                 ,idata,ndata,timedata(20)                         timings5
c                                                                        chmith14
      common /thrmhydr/ tgasa0(20),tgasa1(20),tgasdata(20)               thrmhyd2
     +                 ,pgasa0(20),pgasa1(20),pgasdata(20)               thrmhyd3
     +                 ,vgasa0(20),vgasa1(20),vgasdata(20)               thrmhyd4
     +                 ,tclga0(20),tclga1(20),tclgdata(20)               thrmhyd5
     +                 ,twala0(20),twala1(20),twaldata(20)               thrmhyd6
     +                 ,tflra0(20),tflra1(20),tflrdata(20)               thrmhyd7
     +                 ,sigsa0(20),sigsa1(20),sigsdata(20)               thrmhyd8
     +                 ,radsa0(20),radsa1(20),radsdata(20)               thrmhyd9
     +                 ,mdesa0(20),mdesa1(20),mdesdata(20)               thrmhy10
      real               mdesa0,mdesa1,mdesdata                          thrmhy11
c                                                                        chmith16
c     Find the end-point times for each set of formulae                  chmith17
c     *************************************************                  chmith18
c                                                                        chmith19
      ithhy=0                                                            chmith20
      do 10 idata=1,ndata                                                chmith21
         if(   ithhy.ne.0                                                chmith22
     +   .and. timedata(idata).gt.timethhy(ithhy)                        chmith23
     +   .and. timethhy(ithhy).lt.timeend(ntime))then                    chmith24
         ithhy=ithhy+1                                                   chmith25
         timethhy(ithhy)=timedata(idata)                                 chmith26
         endif                                                           chmith27
         if(ithhy.eq.0 .and. timedata(idata).gt.0.e0)then                chmith28
         ithhy=1                                                         chmith29
         timethhy(1)=timedata(idata)                                     chmith30
         endif                                                           chmith31
10    continue                                                           chmith32
         if(ithhy.eq.0 .or. ndata.eq.1)then                              chmith33
         nthhy=1                                                         chmith34
         timethhy(1)=timeend(ntime)                                      chmith35
         else                                                            chmith36
            if(timethhy(ithhy).ge.timeend(ntime))then                    chmith37
            nthhy=ithhy                                                  chmith38
            else                                                         chmith39
            nthhy=ithhy+1                                                chmith40
            timethhy(nthhy)=timeend(ntime)                               chmith41
            endif                                                        chmith42
         endif                                                           chmith43
c                                                                        chmith44
c     Calculate a0 and a1 for the time dependent variables               chmith45
c     ****************************************************               chmith46
c                                                                        chmith47
      call charmwth(tgasa0,tgasa1,tgasdata)                              chmith48
      call charmwth(pgasa0,pgasa1,pgasdata)                              chmith49
      call charmwth(vgasa0,vgasa1,vgasdata)                              chmith50
      call charmwth(tclga0,tclga1,tclgdata)                              chmith51
      call charmwth(twala0,twala1,twaldata)                              chmith52
      call charmwth(tflra0,tflra1,tflrdata)                              chmith53
      call charmwth(sigsa0,sigsa1,sigsdata)                              chmith54
      call charmwth(radsa0,radsa1,radsdata)                              chmith55
      call charmwth(mdesa0,mdesa1,mdesdata)                              chmith56
      return                                                             chmith57
      end                                                                chmith58
```

```
      subroutine charmwth(a0,a1,xdata)                                  chmwth 2
c                                                                       chmwth 3
c     This subroutine calculates the interpolation coefficients         chmwth 4
c     a0 and a1 from the data table XDATA.                              chmwth 5
c                                                                       chmwth 6
      common /timings/  time,istep,thhystep                             timings2
     +                 ,itime,ntime,timestep(20),timeend(20)            timings3
     +                 ,ithhy,nthhy,timethhy(20)                        timings4
     +                 ,idata,ndata,timedata(20)                        timings5
c                                                                       chmwth 8
      dimension a0(100),a1(100),xdata(100)                              chmwth 9
c                                                                       chmwth10
c     Special case when NDATA = 1                                       chmwth11
c     ***************************                                       chmwth12
c                                                                       chmwth13
      if(ndata.eq.1)then                                                chmwth14
      a0(1)=xdata(1)                                                    chmwth15
      a1(1)=0.e0                                                        chmwth16
      return                                                            chmwth17
      endif                                                             chmwth18
c                                                                       chmwth19
c     Copy data points when insufficient supplied                       chmwth20
c     *******************************************                       chmwth21
c                                                                       chmwth22
      if(ndata.gt.1)then                                                chmwth23
      do 5 idata=2,ndata                                                chmwth24
      if(xdata(idata).lt.0.e0)xdata(idata)=xdata(idata-1)                chmwth25
5     continue                                                          chmwth26
      endif                                                             chmwth27
c                                                                       chmwth28
c     Set-up interpolation formulae when NDATA > 1                      chmwth29
c     ********************************************                      chmwth30
c                                                                       chmwth31
      ithhy=1                                                           chmwth32
      do 10 idata=1,ndata                                               chmwth33
      if(ithhy.le.nthhy)then                                            chmwth34
      if(timedata(idata).eq.timethhy(ithhy))then                        chmwth35
         if(idata.eq.1)then                                             chmwth36
         a0(1)=xdata(1)                                                 chmwth37
         a1(1)=0.e0                                                     chmwth38
         else                                                           chmwth39
         a0(ithhy)=( xdata(idata-1)*timedata(idata)                     chmwth40
     +             - xdata(idata)*timedata(idata-1) )                   chmwth41
     +           / ( timedata(idata) - timedata(idata-1) )              chmwth42
         a1(ithhy)=( xdata(idata) - xdata(idata-1) )                    chmwth43
     +           / ( timedata(idata) - timedata(idata-1) )              chmwth44
         endif                                                          chmwth45
      ithhy=ithhy+1                                                     chmwth46
      endif                                                             chmwth47
      endif                                                             chmwth48
10    continue                                                          chmwth49
      if(timedata(ndata).lt.timethhy(nthhy))then                        chmwth50
      a0(nthhy)=xdata(ndata)                                            chmwth51
      a1(nthhy)=0.e0                                                    chmwth52
      endif                                                             chmwth53
      return                                                            chmwth54
      end                                                               chmwth55
```

```
      subroutine charmuth(timemean)                                          chmuth 2
c                                                                            chmuth 3
c     This subroutine updates the thermal-hydraulic data at time             chmuth 4
c     TIMEMEAN using the interpolation formulae calculated previously.       chmuth 5
c                                                                            chmuth 6
      common /timings/  time,istep,thhystep                                  timings2
     +                 ,itime,ntime,timestep(20),timeend(20)                 timings3
     +                 ,ithhy,nthhy,timethhy(20)                             timings4
     +                 ,idata,ndata,timedata(20)                             timings5
c                                                                            chmuth 8
      common /thrmhydr/ tgasa0(20),tgasa1(20),tgasdata(20)                   thrmhyd2
     +                 ,pgasa0(20),pgasa1(20),pgasdata(20)                   thrmhyd3
     +                 ,vgasa0(20),vgasa1(20),vgasdata(20)                   thrmhyd4
     +                 ,tclga0(20),tclga1(20),tclgdata(20)                   thrmhyd5
     +                 ,twala0(20),twala1(20),twaldata(20)                   thrmhyd6
     +                 ,tflra0(20),tflra1(20),tflrdata(20)                   thrmhyd7
     +                 ,sigsa0(20),sigsa1(20),sigsdata(20)                   thrmhyd8
     +                 ,radsa0(20),radsa1(20),radsdata(20)                   thrmhyd9
     +                 ,mdesa0(20),mdesa1(20),mdesdata(20)                   thrmhy10
      real               mdesa0,mdesa1,mdesdata                              thrmhy11
c                                                                            chmuth10
      common /gasprops/ temp,press,gdensity,dynvisc,molwt,mnfrpath           gasprop2
     +                 ,gthrmcon,velocity,molwtv,diffusv                     gasprop3
     +                 ,vmfrclng,vmfrwall,vmfrflor                           gasprop4
     +                 ,vcgrclng,vcgrwall,vcgrflor                           gasprop5
     +                 ,vconclng,vconwall,vconflor                           gasprop6
      real               molwt,mnfrpath,molwtv                               gasprop7
c                                                                            chmuth12
      common /cell/     areaclng,areawall,areaflor                           cell   2
     +                 ,tempclng,tempwall,tempflor                           cell   3
     +                 ,volume,leakrate                                      cell   4
     +                 ,hydrdiam,eqvrough                                    cell   5
      real               leakrate                                            cell   6
c                                                                            chmuth14
      common /lognorms/ sigmasln,rad50sln,mdensln                            lognorm2
     +                 ,ndensln,geomsln,mas50sln                             lognorm3
      real               mdensln,ndensln,mas50sln                            lognorm4
c                                                                            chmuth16
      temp     = tgasa0(ithhy) + tgasa1(ithhy) * timemean                    chmuth17
      press    = pgasa0(ithhy) + pgasa1(ithhy) * timemean                    chmuth18
      velocity = vgasa0(ithhy) + vgasa1(ithhy) * timemean                    chmuth19
      tempclng = tclga0(ithhy) + tclga1(ithhy) * timemean                    chmuth20
      tempwall = twala0(ithhy) + twala1(ithhy) * timemean                    chmuth21
      tempflor = tflra0(ithhy) + tflra1(ithhy) * timemean                    chmuth22
      sigmasln = sigsa0(ithhy) + sigsa1(ithhy) * timemean                    chmuth23
      rad50sln = radsa0(ithhy) + radsa1(ithhy) * timemean                    chmuth24
      mdensln  = mdesa0(ithhy) + mdesa1(ithhy) * timemean                    chmuth25
      return                                                                 chmuth26
      end                                                                    chmuth27
```

```
      subroutine charmgas                                                    chmgas 2
c                                                                            chmgas 3
c     This subroutine calculates properties of the gas from its              chmgas 4
c     temperature, pressure and molecular weight.                            chmgas 5
c     The formulae are those used in MAEROS.                                 chmgas 6
c                                                                            chmgas 7
      common /gasprops/ temp,press,gdensity,dynvisc,molwt,mnfrpath           gasprop2
     +                 ,gthrmcon,velocity,molwtv,diffusv                     gasprop3
     +                 ,vmfrclng,vmfrwall,vmfrflor                           gasprop4
     +                 ,vcgrclng,vcgrwall,vcgrflor                           gasprop5
     +                 ,vconclng,vconwall,vconflor                           gasprop6
      real               molwt,mnfrpath,molwtv                               gasprop7
c                                                                            chmgas 9
      common /cell/      areaclng,areawall,areaflor                          cell   2
     +                 ,tempclng,tempwall,tempflor                           cell   3
     +                 ,volume,leakrate                                      cell   4
     +                 ,hydrdiam,eqvrough                                    cell   5
      real               leakrate                                            cell   6
c                                                                            chmgas11
      common /fundcon/  pi,boltzmnn,gravitat,gasconst                        fundcon2
c                                                                            chmgas13
      data dynviscr,tempr1/15.65e-6,114.e0/                                  chmgas14
      data diffusvr,tempr2,pressr/2.11e-5,273.15e0,1.01325e5/                chmgas15
c                                                                            chmgas16
c     Density: the molecular weight has units = kgm/kmol.                    chmgas17
c     *******                                                                chmgas18
c                                                                            chmgas19
      gdensity = press * molwt / ( 1.e3 * gasconst * temp )                  chmgas20
c                                                                            chmgas21
c     Dynamic viscosity                                                      chmgas22
c     *****************                                                      chmgas23
c                                                                            chmgas24
      dynvisc = dynviscr * (temp/tempr1)**1.5e0 *1.e0/(1.e0+temp/tempr1)     chmgas25
c                                                                            chmgas26
c     Mean free path                                                         chmgas27
c     **************                                                         chmgas28
c                                                                            chmgas29
      mnfrpath = dynvisc * sqrt(pi/(2.e0*press*gdensity))                    chmgas30
c                                                                            chmgas31
c     Diffusivity of water vapor in air                                      chmgas32
c     *********************************                                      chmgas33
c                                                                            chmgas34
      diffusv = diffusvr * (pressr/press) * (temp/tempr2)**1.94              chmgas35
c                                                                            chmgas36
c     Vapor density near surfaces                                            chmgas37
c     ***************************                                            chmgas38
c                                                                            chmgas39
c     The molecular weight has units = kgm/kmol.                             chmgas40
c                                                                            chmgas41
      vconclng = press*molwtv*vmfrclng / ( 1.e3*gasconst*tempclng )          chmgas42
      vconwall = press*molwtv*vmfrwall / ( 1.e3*gasconst*tempwall )          chmgas43
      vconflor = press*molwtv*vmfrflor / ( 1.e3*gasconst*tempflor )          chmgas44
      return                                                                 chmgas45
      end                                                                    chmgas46
```

```
      subroutine charmmob                                                    chmmob 2
c                                                                            chmmob 3
c     This subroutine calculates particle mobilities at the collocation      chmmob 4
c     points using the MAEROS formula.                                       chmmob 5
c                                                                            chmmob 6
      common /collpts/  ncoll,mlower,mupper,spacing,dlogem,logem0            collpts2
     +                 ,radius(100),mass(100),mobility(100)                  collpts3
      real              logem0,mass,mobility,mlower,mupper                   collpts4
c                                                                            chmmob 8
      common /aerslcon/ cshpfctr,dshpfctr,stickeff                           aerslco2
     +                 ,aknudweb,qknudweb,bknudweb                           aerslco3
     +                 ,pdensity,pthrmcon                                    aerslco4
     +                 ,kbrock,cmbrock,ctbrock                               aerslco5
      real              kbrock                                               aerslco6
c                                                                            chmmob10
      common /gasprops/ temp,press,gdensity,dynvisc,molwt,mnfrpath           gasprop2
     +                 ,gthrmcon,velocity,molwtv,diffusv                     gasprop3
     +                 ,vmfrclng,vmfrwall,vmfrflor                           gasprop4
     +                 ,vcgrclng,vcgrwall,vcgrflor                           gasprop5
     +                 ,vconclng,vconwall,vconflor                           gasprop6
      real              molwt,mnfrpath,molwtv                                gasprop7
c                                                                            chmmob12
      common /fundcon/  pi,boltzmnn,gravitat,gasconst                        fundcon2
c                                                                            chmmob14
c                                                                            chmmob15
c                                                                            chmmob16
      do 100 icoll=1,ncoll                                                   chmmob17
c                                                                            chmmob18
c     Stoke's law mobility                                                   chmmob19
c     ********************                                                   chmmob20
c                                                                            chmmob21
      stokes = 1.e0/(6.e0*pi*dshpfctr*dynvisc*radius(icoll))                 chmmob22
c                                                                            chmmob23
c     Cunningham slip correction                                             chmmob24
c     **************************                                             chmmob25
c                                                                            chmmob26
      rl = radius(icoll)/mnfrpath                                            chmmob27
      cunning = 1.e0 + aknudweb/rl                                           chmmob28
     +        + qknudweb/rl*exp(-min(bknudweb*rl,100.e0))                    chmmob29
c                                                                            chmmob30
c     Combine the two factors                                                chmmob31
c     ***********************                                                chmmob32
c                                                                            chmmob33
      mobility(icoll) = stokes * cunning                                     chmmob34
100   continue                                                               chmmob35
      return                                                                 chmmob36
      end                                                                    chmmob37
```

```
      subroutine charmflo                                                    chmflo 2
c                                                                            chmflo 3
c     This subroutine calculates flow properties which are needed            chmflo 4
c     for the calculation of deposition velocities and                       chmflo 5
c     agglomeration rates.                                                   chmflo 6
c                                                                            chmflo 7
      common /gasprops/ temp,press,gdensity,dynvisc,molwt,mnfrpath           gasprop2
     +                 ,gthrmcon,velocity,molwtv,diffusv                     gasprop3
     +                 ,vmfrclng,vmfrwall,vmfrflor                           gasprop4
     +                 ,vcgrclng,vcgrwall,vcgrflor                           gasprc
     +                 ,vconclng,vconwall,vconflor                           gaspro
      real               molwt,mnfrpath,molwtv                               gasprop7
c                                                                            chmflo 9
      common /fundcon/  pi,boltzmnn,gravitat,gasconst                        fundcon2
c                                                                            chmflo11
      common /collpts/  ncoll,mlower,mupper,spacing,dlogem,logem0            collpts2
     +                 ,radius(100),mass(100),mobility(100)                  collpts3
      real               logem0,mass,mobility,mlower,mupper                  collpts4
c                                                                            chmflo13
      common /cell/     areaclng,areawall,areaflor                           cell   2
     +                 ,tempclng,tempwall,tempflor                          cell   3
     +                 ,volume,leakrate                                     cell   4
     +                 ,hydrdiam,eqvrough                                   cell   5
      real               leakrate                                            cell   6
c                                                                            chmflo15
      common /flow/     blflag,vblthick,dblthick(100)                        flow   2
     +                 ,eddydiss,ustar,reynolds                             flow   3
      integer            blflag                                              flow   4
c                                                                            chmflo17
      common /toleranc/ eps,eta,zeta,maxcalls,maxtrys                        toleran2
c                                                                            chmflo19
      common /iotapes/  columns,ntape4,ntape5,ntape6                         iotapes2
      integer           columns                                              iotapes3
c                                                                            chmflo21
      real kt                                                                chmflo22
      real kinvisc                                                           chmflo23
      logical firstgo                                                        chmflo24
      data firstgo/.true./                                                   chmflo25
      data reyncut/2300.e0/                                                  chmflo26
      external charmfan                                                      chmflo27
c                                                                            chmflo28
      kt = boltzmnn*temp                                                     chmflo29
c                                                                            chmflo30
c     Friction velocity                                                      chmflo31
c     *****************                                                      chmflo32
c                                                                            chmflo33
c     The formulae are from Schlichting. They are for a pipe of any          chmflo34
c     roughness - the roughness required by the correlation is the           chmflo35
c     equivalent sand roughness as first used by Nikuradse. The              chmflo36
c     hydraulic diameter is 4 * the flow area / the wetted perimeter.        chmflo37
c     This equals the pipe diameter for cylindrical pipes but enables        chmflo38
c     the correlation to be used for other cross-sectional shapes.           chmflo39
c                                                                            chmflo40
      kinvisc=dynvisc/gdensity                                               chmflo41
      reynolds=velocity*hydrdiam/kinvisc                                     chmflo42
      if(reynolds.lt.reyncut)then                                            chmflo43
      ustar=0.e0                                                             chmflo44
      else                                                                   chmflo45
      ifail = 1                                                              chmflo46
      flower = 1.e-10                                                        chmflo47
      fupper = 1.e0                                                          chmflo48
      rtol = eps                                                             chmflo49
      atol = eta                                                             chmflo50
      ztol = zeta                                                            chmflo51
      ncalls = maxcalls                                                      chmflo52
      call c05whe                                                            chmflo53
     +   (flower,fupper,rtol,atol,charmfan,ncalls,ztol,fanning,ifail)        chmflo54
         if(ifail.ne.0)then                                                  chmflo55
         write(ntape5,1000)ifail                                             chmflo56
```

```
            write(ntape6,1000)ifail                                          chmflo57
1000        format(4x,'*** CHARMFLO fails: IFAIL is ',i2,' ***')             chmflo58
            stop                                                             chmflo59
            endif                                                            chmflo60
         ustar = velocity * sqrt(fanning/2.e0)                               chmflo61
         endif                                                               chmflo62
c                                                                            chmflo63
c        Eddy dissipation rate                                               chmflo64
c        *********************                                               chmflo65
c                                                                            chmflo66
c        The eddy dissipation rate is calculated by assuming all the energychmflo67
c        consumed by shear stresses at the walls is dissipated by turbulencchmflo68
c                                                                            chmflo69
         if(hydrdiam.eq.0.e0)then                                            chmflo70
         eddydiss=0.e0                                                       chmflo71
         else                                                                chmflo72
         eddydiss=4.e0*velocity*ustar*ustar/hydrdiam                         chmflo73
         endif                                                               chmflo74
c                                                                            chmflo75
c        Viscous boundary layer thickness                                    chmflo76
c        ********************************                                    chmflo77
c                                                                            chmflo78
         if(blflag.eq.0)then                                                 chmflo79
            if(ustar.eq.0.e0)then                                            chmflo80
            vblthick=0.e0                                                    chmflo81
            else                                                             chmflo82
            vblthick=kinvisc/ustar                                           chmflo83
            endif                                                            chmflo84
         endif                                                               chmflo85
c                                                                            chmflo86
c        Diffusion boundary layer thickness                                  chmflo87
c        **********************************                                  chmflo88
c                                                                            chmflo89
c        The diffusion boundary layer thickness is an input data item        chmflo90
c        or else defaults to 1.e-5 in MAEROS.                                chmflo91
c                                                                            chmflo92
c        Here it is calculated from the viscous boundary layer               chmflo93
c        thickness using the Keller (1973) formula.                          chmflo94
c                                                                            chmflo95
         if(blflag.eq.0)then                                                 chmflo96
            do 10 icoll=1,ncoll                                              chmflo97
            schmidt = kinvisc/(kt*mobility(icoll))                           chmflo98
            dblthick(icoll) = vblthick / schmidt**(1.e0/3.e0)                chmflo99
10          continue                                                         chmfl100
         else                                                                chmfl101
            if(firstgo)then                                                  chmfl102
               firstgo = .false.                                             chmfl103
               do 20 icoll=1,ncoll                                           chmfl104
               dblthick(icoll)=dblthick(1)                                   chmfl105
20             continue                                                      chmfl106
            endif                                                            chmfl107
         endif                                                               chmfl108
         return                                                              chmfl109
         end                                                                 chmfl110
```

```
      subroutine c05whe(x1,xu,eps,eta,func,maxcalls,zeta,x,ifail)       c05whe 2
c                                                                       c05whe 3
c     This subroutine locates zeros of a function of one variable. The  c05whe 4
c     method employs inverse quadratic interpolation and bisection.     c05whe 5
c                                                                       c05whe 6
c     ifail = 1 on entry: soft fail.    ifail = 0 on entry: hard fail.  c05whe 7
c     ifail = 0 on return: success.     ifail = 1 on return: no zero.   c05whe 8
c     ifail = 2 on return: maxcalls reached.                            c05whe 9
c                                                                       c05whe10
c     written by C.J.Wheatley October 1985.                             c05whe11
c                                                                       c05whe12
      common /iotapes/  columns,ntape4,ntape5,ntape6                    iotapes2
      integer           columns                                         iotapes3
c                                                                       c05whe14
      eps=amax1(1.e-14,eps)                                             c05whe15
      eta=amax1(0.e0,eta)                                               c05whe16
      zeta=amax1(0.5e0,zeta)                                            c05whe17
      maxcalls=max0(3,maxcalls)                                         c05whe18
c                                                                       c05whe19
c     check if zero present and start iteration.                        c05whe20
c     ******************************************                        c05whe21
c                                                                       c05whe22
      x1=x1                                                             c05whe23
      x3=xu                                                             c05whe24
      y1 = func(x1)                                                     c05whe25
      y3 = func(x3)                                                     c05whe26
      icalls=2                                                          c05whe27
      if(y1*y3.ge.0.e0)goto 1000                                        c05whe28
      x2=(x1+x3)/2.e0                                                   c05whe29
c                                                                       c05whe30
c           start of iteration loop.                                    c05whe31
c           ************************                                    c05whe32
c                                                                       c05whe33
100         y2 = func(x2)                                             c05whe34
            icalls=icalls+1                                           c05whe35
c                                                                       c05whe36
c           test for last iteration step.                               c05whe37
c           *****************************                               c05whe38
c                                                                       c05whe39
            if(abs(y2).le.eta)goto 500                                c05whe40
            if((y1*y2.lt.0.e0).and.(abs(x2-x1).le.eps))goto 600       c05whe41
            if((y2*y3.le.0.e0).and.(abs(x3-x2).le.eps))goto 700       c05whe42
            if(icalls.eq.maxcalls)goto 2000                           c05whe43
c                                                                       c05whe44
c           test for inverse quadratic interpolation step.              c05whe45
c           **********************************************              c05whe46
c                                                                       c05whe47
            denom=x1*(y2-y3)+x2*(y3-y1)+x3*(y1-y2)                    c05whe48
            if(denom.eq.0.e0)goto 200                                 c05whe49
            dy=0.5e0*(y1-y2)*(y2-y3)*(x3-x1)/denom                    c05whe50
            if(abs(dy).ge.zeta*abs(y3-y1))goto 200                    c05whe51
c                                                                       c05whe52
c           interpolate.                                                c05whe53
c           ************                                                c05whe54
c                                                                       c05whe55
            if(y1*y2.lt.0.e0)x=0.5e0*(x1+x2)                          c05whe56
            if(y2*y3.lt.0.e0)x=0.5e0*(x2+x3)                          c05whe57
            goto 300                                                  c05whe58
200         x=x2-y2*(y3*((x2-x1)/(y2-y1))-y1*((x2-x3)/(y2-y3)))/(y3-y1) c05whe59
c                                                                       c05whe60
c           revise range.                                               c05whe61
c           *************                                               c05whe62
c                                                                       c05whe63
300         if(y1*y2.lt.0.e0)goto 310                                 c05whe64
            x1=x2                                                     c05whe65
            y1=y2                                                     c05whe66
            goto 400                                                  c05whe67
310         x3=x2                                                     c05whe68
            y3=y2                                                     c05whe69
```

```
c                                                                       c05whe70
c           revise intermediate point.                                  c05whe71
c           **************************                                  c05whe72
c                                                                       c05whe73
400         x2=x                                                        c05whe74
            if(abs(x3-x1).le.2.e0*eps)x2=0.5e0*(x1+x3)                  c05whe75
            if(abs(x3-x1).le.2.e0*eps)goto 100                          c05whe76
            if(abs(x2-x1).lt.eps)x2=x1+sign(eps,x3-x1)                  c05whe77
            if(abs(x3-x2).lt.eps)x2=x3-sign(eps,x3-x1)                  c05whe78
            goto 100                                                    c05whe79
c                                                                       c05whe80
c           end of iteration loop.                                      c05whe81
c           **********************                                      c05whe82
c                                                                       c05whe83
500   x=x2                                                              c05whe84
      goto 710                                                          c05whe85
600   x=x2                                                              c05whe86
      if(abs(y1).lt.abs(y2))x=x1                                        c05whe87
      goto 710                                                          c05whe88
700   x=x2                                                              c05whe89
      if(abs(y3).lt.abs(y2))x=x3                                        c05whe90
710   ifail=0                                                           c05whe91
      return                                                            c05whe92
c                                                                       c05whe93
c           no zero found.                                              c05whe94
c           **************                                              c05whe95
c                                                                       c05whe96
1000        if(ifail.eq.1)return                                        c05whe97
            write(ntape5,9000)                                          c05whe98
            write(ntape5,9001)x1,y1,x3,y3                               c05whe99
            write(ntape6,9000)                                          c05wh100
            write(ntape6,9001)x1,y1,x3,y3                               c05wh101
            stop                                                        c05wh102
c                                                                       c05wh103
c           maxcalls reached.                                           c05wh104
c           *****************                                           c05wh105
c                                                                       c05wh106
2000        if(ifail.eq.1)ifail=2                                       c05wh107
            if(ifail.eq.2)return                                        c05wh108
            write(ntape5,9002)maxcalls                                  c05wh109
            write(ntape5,9001)x1,y1,x3,y3                               c05wh110
            write(ntape6,9002)maxcalls                                  c05wh111
            write(ntape6,9001)x1,y1,x3,y3                               c05wh112
            stop                                                        c05wh113
9000  format(                                                           c05wh114
     +4x,'*** c05whe fails: interval does not contain a zero ***')      c05wh115
9001  format(                                                           c05wh116
     +4x,'***      x1           y1           x3           y3      ***',/ c05wh117
     +4x,'***',1p4e12.4,' ***')                                         c05wh118
9002  format(                                                           c05wh119
     +4x,'*** c05whe fails: zero not found after ',i3,' steps ***')     c05wh120
      end                                                               c05wh121
```

```
      function charmfan(f)                                                  chmfan 2
c                                                                           chmfan 3
c     This function subroutine is used to calculate the fanning             chmfan 4
c     friction factor for a cylindrical pipe with arbitrary roughness.      chmfan 5
c                                                                           chmfan 6
      common /cell/     areaclng,areawall,areaflor                          cell   2
     +                 ,tempclng,tempwall,tempflor                         cell   3
     +                 ,volume,leakrate                                    cell   4
     +                 ,hydrdiam,eqvrough                                  cell   5
      real               leakrate                                          cell   6
c                                                                           chmfan 8
      common /flow/     blflag,vblthick,dblthick(100)                       flow   2
     +                 ,eddydiss,ustar,reynolds                            flow   3
      integer            blflag                                            flow   4
c                                                                           chmfan10
      sqrtfric = 2.e0 * sqrt(f)                                           chmfan11
      charmfan = 1.e0 - sqrtfric * ( 1.74e0 - 2.e0 *                      chmfan12
     +   alog10( 2.e0*eqvrough/hydrdiam + 18.7e0/(reynolds*sqrtfric) ) )  chmfan13
      return                                                              chmfan14
      end                                                                 chmfan15
```

```
      subroutine charmagg                                                    chmagg 2
c                                                                            chmagg 3
c     This subroutine calculates the agglomeration rates of the              chmagg 4
c     particles at all combinations of the collocation points. This          chmagg 5
c     is done in a nested DO loop which could be written as a single         chmagg 6
c     loop to enable full vectorization if this program unit turns-out       chmagg 7
c     to be computationally costly. The agglomeration rates are              chmagg 8
c     identical to those used in MAEROS to facilitate comparisons. The       chmagg 9
c     particle radius, mass and mobility at the collocation points are       chmagg10
c     assumed to have been previously calculated.                            chmagg11
c                                                                            chmagg12
      common /collpts/  ncoll,mlower,mupper,spacing,dlogem,logem0            collpts2
     +                 ,radius(100),mass(100),mobility(100)                 collpts3
      real              logem0,mass,mobility,mlower,mupper                   collpts4
c                                                                            chmagg14
      common /aerslcon/ cshpfctr,dshpfctr,stickeff                           aerslco2
     +                 ,aknudweb,qknudweb,bknudweb                           aerslco3
     +                 ,pdensity,pthrmcon                                    aerslco4
     +                 ,kbrock,cmbrock,ctbrock                               aerslco5
      real              kbrock                                               aerslco6
c                                                                            chmagg16
      common /gasprops/ temp,press,gdensity,dynvisc,molwt,mnfrpath           gasprop2
     +                 ,gthrmcon,velocity,molwtv,diffusv                     gasprop3
     +                 ,vmfrclng,vmfrwall,vmfrflor                           gasprop4
     +                 ,vcgrclng,vcgrwall,vcgrflor                           gasprop5
     +                 ,vconclng,vconwall,vconflor                           gasprop6
      real              molwt,mnfrpath,molwtv                                gasprop7
c                                                                            chmagg18
      common /fundcon/  pi,boltzmnn,gravitat,gasconst                        fundcon2
c                                                                            chmagg20
      common /agglom/   agglomrt(100,100),deposrtf(100)                     agglom 2
     +                 ,deposrtw(100),deposrtc(100)                          agglom 3
c                                                                            chmagg22
      real ma1,ma2,mo1,mo2,kt                                                chmagg23
      kt = boltzmnn*temp                                                     chmagg24
c                                                                            chmagg25
c     Nested DO loops to calculate agglomeration rates.                      chmagg26
c     *************************************************                      chmagg27
c                                                                            chmagg28
      do 100 icoll1=1,ncoll                                                  chmagg29
      do 200 icoll2=1,icoll1                                                 chmagg30
      r1 = radius(icoll1)                                                    chmagg31
      r2 = radius(icoll2)                                                    chmagg32
      ma1 = mass(icoll1)                                                     chmagg33
      ma2 = mass(icoll2)                                                     chmagg34
      mo1 = mobility(icoll1)                                                 chmagg35
      mo2 = mobility(icoll2)                                                 chmagg36
      cr1=r1*cshpfctr                                                        chmagg37
      cr2=r2*cshpfctr                                                        chmagg38
c                                                                            chmagg39
c     Brownian agglomeration                                                 chmagg40
c     **********************                                                 chmagg41
c                                                                            chmagg42
      vbar = sqrt( 8.e0*kt*(1.e0/ma1 + 1.e0/ma2)/pi )                        chmagg43
      fuchs1 = kt*(mo1 + mo2) / (stickeff*vbar*(r1 + r2))                    chmagg44
c                                                                            chmagg45
      a1 = mo1*sqrt( 2.e0*kt*ma1/pi )                                        chmagg46
      a2 = mo2*sqrt( 2.e0*kt*ma2/pi )                                        chmagg47
      g1 = ((r1+a1)**3-(r1*r1+a1*a1)**1.5e0)/(3.e0*r1*a1) - r1               chmagg48
      g2 = ((r2+a2)**3-(r2*r2+a2*a2)**1.5e0)/(3.e0*r2*a2) - r2               chmagg49
      gbar = sqrt( g1*g1 + g2*g2 )                                           chmagg50
      fuchs2 = 1.e0 / (1.e0 + 2.e0*gbar/(r1 + r2))                           chmagg51
c                                                                            chmagg52
c     Note that there is no sticking efficiency factor.                      chmagg53
c     FUCHS2 equals unity in Dunbar et al., EUR 9172.                        chmagg54
c                                                                            chmagg55
      fuchscor = 1.e0 / (fuchs1 + fuchs2)                                    chmagg56
      aggb = 4.e0 * pi * kt * (mo1 + mo2) * (cr1 + cr2) * fuchscor           chmagg57
c                                                                            chmagg58
```

```
c     Gravitational agglomeration                                       chmagg59
c     ***************************                                       chmagg60
c                                                                       chmagg61
c     The Fuchs collision efficiency...                                 chmagg62
c                                                                       chmagg63
      colleff = 1.5e0 * ( min(r1,r2)/(r1+r2) )**2                       chmagg64
c                                                                       chmagg65
c     Dunbar et al. has pdensity - gdensity / pdensity  here...         chmagg66
c                                                                       chmagg67
      relvel = gravitat * ( ma1*mo1 - ma2*mo2 )                         chmagg68
c                                                                       chmagg69
c                                                                       chmagg70
      aggg = colleff * stickeff * abs(relvel) * pi * (cr1+cr2)**2       chmagg71
c                                                                       chmagg72
c     Turbulent agglomeration                                           chmagg73
c     ***********************                                           chmagg74
c                                                                       chmagg75
      tfactor = 8.e0*pi*gdensity*eddydiss/(15.e0*dynvisc)               chmagg76
      aggts = stickeff * (cr1+cr2)**3 * sqrt( tfactor )                 chmagg77
      aggti = stickeff * (cr1+cr2)**2 * sqrt( 8.e0*pi*eddydiss )        chmagg78
     +              * ( tfactor )**.25e0 * abs(relvel) / gravitat       chmagg79
c                                                                       chmagg80
c     Add the shear and inertial contributions in quadrature - note     chmagg81
c     the collision efficiency factor is missing from both terms.       chmagg82
c                                                                       chmagg83
      aggt = sqrt(aggts*aggts + aggti*aggti)                            chmagg84
c                                                                       chmagg85
c     Combine the agglomeration contributions                           chmagg86
c     ***************************************                           chmagg87
c                                                                       chmagg88
      agglomrt(icoll1,icoll2) = aggb + aggg + aggt                      chmagg89
      agglomrt(icoll2,icoll1) = agglomrt(icoll1,icoll2)                 chmagg90
200   continue                                                          chmagg91
100   continue                                                          chmagg92
      return                                                            chmagg93
      end                                                               chmagg94
```

```
      subroutine charmdep                                                    chmdep 2
c                                                                            chmdep 3
c     This subroutine calculates the particle deposition rates               chmdep 4
c     at the collocation points. The deposition rates for                    chmdep 5
c     gravitation, Brownian diffusion and thermophoresis are                 chmdep 6
c     identical to those used in MAEROS to facilitate comparisons.           chmdep 7
c     Deposition due to turbulent diffusion, turbulent impaction and         chmdep 8
c     diffusiophoresis have also been included. The diffusiophoresis         chmdep 9
c     formula is taken from a later version of MAEROS which includes         chmdep10
c     this mechanism. It was an input data item in early versions.           chmdep11
c                                                                            chmdep12
c     The particle radius, mass and mobility at the collocation points       chmdep13
c     are assumed to have been previously calculated.                        chmdep14
c                                                                            chmdep15
c     Deposition rates onto floors, walls and ceilings are kept              chmdep16
c     separate to enable the deposited mass onto the three                   chmdep17
c     surfaces to be separately integrated.                                  chmdep18
c                                                                            chmdep19
      common /collpts/  ncoll,mlower,mupper,spacing,dlogem,logem0            collpts2
     +                  radius(100),mass(100),mobility(100)                  collpts3
      real              logem0,mass,mobility,mlower,mupper                   collpts4
c                                                                            chmdep21
      common /aerslcon/ cshpfctr,dshpfctr,stickeff                           aerslco2
     +                 ,aknudweb,qknudweb,bknudweb                           aerslco3
     +                 ,pdensity,pthrmcon                                    aerslco4
     +                 ,kbrock,cmbrock,ctbrock                               aerslco5
      real              kbrock                                               aerslco6
c                                                                            chmdep23
      common /gasprops/ temp,press,gdensity,dynvisc,molwt,mnfrpath           gasprop2
     +                 ,gthrmcon,velocity,molwtv,diffusv                     gasprop3
     +                 ,vmfrclng,vmfrwall,vmfrflor                           gasprop4
     +                 ,vcgrclng,vcgrwall,vcgrflor                           gasprop5
     +                 ,vconclng,vconwall,vconflor                           gasprop6
      real              molwt,mnfrpath,molwtv                                gasprop7
c                                                                            chmdep25
      common /fundcon/  pi,boltzmnn,gravitat,gasconst                        fundcon2
c                                                                            chmdep27
      common /agglom/   agglomrt(100,100),deposrtf(100)                     agglom 2
     +                 ,deposrtw(100),deposrtc(100)                          agglom 3
c                                                                            chmdep29
      common /cell/     areaclng,areawall,areaflor                           cell   2
     +                 ,tempclng,tempwall,tempflor                           cell   3
     +                 ,volume,leakrate                                      cell   4
     +                 ,hydrdiam,eqvrough                                    cell   5
      real              leakrate                                             cell   6
c                                                                            chmdep31
      common /flow/     blflag,vblthick,dblthick(100)                        flow   2
     +                 ,eddydiss,ustar,reynolds                              flow   3
      integer           blflag                                               flow   4
c                                                                            chmdep33
      real kt,mo                                                             chmdep34
c                                                                            chmdep35
c     In MAEROS, ratiocon defaults to 0.05.                                  chmdep36
c                                                                            chmdep37
      kt = boltzmnn*temp                                                     chmdep38
      ratiocon = gthrmcon/pthrmcon                                           chmdep39
      deltclng = 1.e0 - tempclng/temp                                        chmdep40
      deltwall = 1.e0 - tempwall/temp                                        chmdep41
      deltflor = 1.e0 - tempflor/temp                                        chmdep42
c                                                                            chmdep43
c     DO loop to calculate deposition rates                                  chmdep44
c     *************************************                                  chmdep45
c                                                                            chmdep46
      do 100 icoll=1,ncoll                                                   chmdep47
      r = radius(icoll)                                                      chmdep48
      rl = r/mnfrpath                                                        chmdep49
      mo = mobility(icoll)                                                   chmdep50
c                                                                            chmdep51
c     Gravitational deposition velocity                                      chmdep52
```

```
c     ********************************                                  chmdep53
c                                                                       chmdep54
c     Dunbar et al. have pdensity - gdensity / pdensity here...         chmdep55
c                                                                       chmdep56
      vgravity = gravitat*mass(icoll)*mo                                chmdep57
c                                                                       chmdep58
c     Brownian diffusion deposition velocity                            chmdep59
c     **************************************                            chmdep60
c                                                                       chmdep61
      if(dblthick(icoll).eq.0.e0)then                                   chmdep62
      vbrown = 0.e0                                                     chmdep63
      else                                                              chmdep64
      vbrown = 0.0594e0 * kt*mo/dblthick(icoll)                         chmdep65
      endif                                                             chmdep66
c                                                                       chmdep67
c     Turbulence deposition velocity                                    chmdep68
c     ******************************                                    chmdep69
c                                                                       chmdep70
c     The correlation used here is derived from the Liu and Agarwal     chmdep71
c     data for dimensionless deposition velocity vs dimensionless       chmdep72
c     particle relaxation time. The correlation is unverified for       chmdep73
c     dimensionless relaxation time lt .1 and gt 100.                   chmdep74
c                                                                       chmdep75
      relaxtim = mass(icoll)*mo                                         chmdep76
      dimrelax = relaxtim * ustar * ustar * gdensity / dynvisc          chmdep77
      if(dimrelax.eq.0.e0)then                                          chmdep78
         dimvturb=0.e0                                                  chmdep79
      else                                                              chmdep80
         dimvtur1 = 6.e-4 * dimrelax * dimrelax                         chmdep81
         dimvtur2 = 2.13e-1 * dimrelax**(-0.125e0)                      chmdep82
         dimvturb = 1.e0 /                                              chmdep83
     +   sqrt( 1.e0/(dimvtur1*dimvtur1) + 1.e0/(dimvtur2*dimvtur2) )    chmdep84
      endif                                                             chmdep85
      vturb = dimvturb * ustar                                          chmdep86
c                                                                       chmdep87
c     Thermophoresis deposition velocity                                chmdep88
c     **********************************                                chmdep89
c                                                                       chmdep90
c     The gdensity*vblthick term is missing in early versions of MAEROS.chmdep91
c                                                                       chmdep92
      brockfac = kbrock / (1.e0 + 3.e0*cmbrock/rl)                      chmdep93
     +                  / (2.e0 + 1.e0/(ratiocon*ctbrock/rl))           chmdep94
      if(vblthick.eq.0.e0)then                                          chmdep95
      vthermo = 0.e0                                                    chmdep96
      else                                                              chmdep97
      vthermo = 9.e0*pi*dynvisc*dynvisc                                 chmdep98
     +             *r*mo*brockfac/ gdensity*vblthick)                   chmdep99
      endif                                                             chmde100
c                                                                       chmde101
c     Diffusiophoresis deposition velocity                              chmde102
c     ************************************                              chmde103
c                                                                       chmde104
c     This is taken from a later version of MAEROS.                     chmde105
c                                                                       chmde106
      if(vconclng.eq.0.e0)then                                          chmde107
      vdfoclng=0.e0                                                     chmde108
      else                                                              chmde109
      vdfoclng = diffusv * vcgrclng / vconclng                          chmde110
     +   * vmfrclng / ( vmfrclng + (1.e0-vmfrclng)*sqrt(molwt/molwtv) ) chmde111
      endif                                                             chmde112
c                                                                       chmde113
      if(vconwall.eq.0.e0)then                                          chmde114
      vdfowall=0.e0                                                     chmde115
      else                                                              chmde116
      vdfowall = diffusv * vcgrwall / vconwall                          chmde117
     +   * vmfrwall / ( vmfrwall + (1.e0-vmfrwall)*sqrt(molwt/molwtv) ) chmde118
      endif                                                             chmde119
c                                                                       chmde120
      if(vconflor.eq.0.e0)then                                          chmde121
```

```
      vdfoflor=0.e0                                                          chmde122
      else                                                                   chmde123
      vdfoflor = diffusv * vcgrflor / vconflor                               chmde124
     +    * vmfrflor / ( vmfrflor + (1.e0-vmfrflor)*sqrt(molwt/molwtv) )    chmde125
      endif                                                                  chmde126
c                                                                            chmde127
c     Net deposition velocity onto ceiling, walls and floor                  chmde128
c     ************************************************                       chmde129
c                                                                            chmde130
c     Note the thermo- and diffusiophoresis terms could be negative...       chmde131
c                                                                            chmde132
      vclng = max(0.e0,vbrown+vturb+deltclng*vthermo+vdfoclng-vgravity)      chmde133
      vwall = max(0.e0,vbrown+vturb+deltwall*vthermo+vdfowall)               chmde134
      vflor = max(0.e0,vbrown+vturb+deltflor*vthermo+vdfoflor+vgravity)      chmde135
c                                                                            chmde136
c     Store deposition rates for ceiling, walls and floor                    chmde137
c     ***************************************************                    chmde138
c                                                                            chmde139
      deposrtf(icoll) = areaflor*vflor/volume                                chmde140
      deposrtw(icoll) = areawall*vwall/volume                                chmde141
      deposrtc(icoll) = areaclng*vclng/volume                                chmde142
100   continue                                                               chmde143
      return                                                                 chmde144
      end                                                                    chmde145
```

```
      subroutine charmsln                                                chmsln 2
c                                                                        chmsln 3
c     This subroutine sets-up the source number density distribution.    chmsln 4
c                                                                        chmsln 5
c     It takes the source distribution to be log-normal. The three       chmsln 6
c     input parameters are the cube root of the geometric mass           chmsln 7
c     standard deviation (in keeping with the normal convention),        chmsln 8
c     sigmasln (no units), the mass median radius, rad50sln (m),         chmsln 9
c     and the mass density generation rate, mdensln (kg m**-3 s**-1).    chmsln10
c                                                                        chmsln11
c     mdensln = the mass generation rate divided by the cell volume.     chmsln12
c                                                                        chmsln13
c     The three parameters of the log-normal distribution are the        chmsln14
c     number density generation rate, ndensln (m**-3 s**-1), the         chmsln15
c     geometric mean mass, geomsln (kg), and the logarithm of the        chmsln16
c     geometric mass standard deviation, logsigma (no units).            chmsln17
c                                                                        chmsln18
c     The discretized distribution is stored as the number density       chmsln19
c     generation rate times mass, since this is the most convenient      chmsln20
c     variable for the aerosol equation.                                 chmsln21
c                                                                        chmsln22
      common /fundcon/  pi,boltzmnn,gravitat,gasconst                    fundcon2
c                                                                        chmsln24
      common /aerslcon/ cshpfctr,dshpfctr,stickeff                       aerslco2
     +                 ,aknudweb,qknudweb,bknudweb                       aerslco3
     +                 ,pdensity,pthrmcon                                aerslco4
     +                 ,kbrock,cmbrock,ctbrock                           aerslco5
      real              kbrock                                           aerslco6
c                                                                        chmsln28
      common /collpts/  ncoll,mlower,mupper,spacing,dlogem,logem0        collpts2
     +                 ,radius(100),mass(100),mobility(100)              collpts3
      real              logem0,mass,mobility,mlower,mupper               collpts4
c                                                                        chmsln28
      common /lognorms/ sigmasln,rad50sln,mdensln                        lognorm2
     +                 ,ndensln,geomsln,mas50sln                         lognorm3
      real              mdensln,ndensln,mas50sln                         lognorm4
c                                                                        chmsln30
      common /source/   sourcert(100)                                    source 2
c                                                                        chmsln32
      real logsigma,mdensity                                             chmsln33
c                                                                        chmsln34
c     Convert the input parameters to the log-normal parameters          chmsln35
c     *********************************************************          chmsln36
c                                                                        chmsln37
      logsigma = 3.e0*alog( sigmasln )                                   chmsln38
      mas50sln=4.e0*pi*pdensity*rad50sln*rad50sln*rad50sln/3.e0          chmsln39
      geomsln=mas50sln*exp(-logsigma*logsigma)                           chmsln40
      ndensln=mdensln*exp(-logsigma*logsigma/2.e0)/geomsln               chmsln41
c                                                                        chmsln42
c     Calculate the distribution at the collocation points and the       chmsln43
c     mass density generation rate of the discretized distribution       chmsln44
c     *********************************************************          chmsln45
c                                                                        chmsln46
      mdensity=0.e0                                                      chmsln47
      const=ndensln/( sqrt(2.e0*pi) * logsigma )                         chmsln48
      do 100 icoll=1,ncoll                                               chmsln49
      exponent=alog( mass(icoll)/geomsln ) / logsigma                    chmsln50
      sourcert(icoll)=const*exp(-exponent*exponent/2.e0)                 chmsln51
      mdensity=mdensity+dlogem*sourcert(icoll)*mass(icoll)               chmsln52
100   continue                                                           chmsln53
c                                                                        chmsln54
c     Renormalize so that no mass is lost                                chmsln55
c     ***********************************                                chmsln56
c                                                                        chmsln57
      if(mdensity.ne.0.e0)then                                           chmsln58
      renorm=mdensln/mdensity                                            chmsln59
      do 200 icoll=1,ncoll                                               chmsln60
      sourcert(icoll)=sourcert(icoll)*renorm                             chmsln61
200   continue                                                           chmsln62
```

```
endif                                                          chmsln63
return                                                         chmsln64
end                                                            chmsln65
```

```
      subroutine charmdif(reset)                                        chmdif 2
c                                                                       chmdif 3
c     This subroutine sets-up the input for the CLAMS ODE solver        chmdif 4
c     DEBDF, calls DEBDF for one time step and checks that the          chmdif 5
c     integration was done correctly.                                   chmdif 6
c                                                                       chmdif 7
      common /collpts/  ncoll,mlower,mupper,spacing,dlogem,logem0       collpts2
     +                 ,radius(100),mass(100),mobility(100)            collpts3
      real              logem0,mass,mobility,mlower,mupper              collpts4
c                                                                       chmdif 9
      common /distrib/  zstore(105)                                     distrib2
c                                                                       chmdif11
      common /timings/  time,istep,thhystep                             timings2
     +                 ,itime,ntime,timestep(20),timeend(20)            timings3
     +                 ,ithhy,nthhy,timethhy(20)                        timings4
     +                 ,idata,ndata,timedata(20)                        timings5
c                                                                       chmdif13
      common /toleranc/ eps,eta,zero,maxcalls,maxtrys                   toleran2
c                                                                       chmdif15
      common /moments/  sigma,rad50,mdensity,ndensity,geommean,mass50   moments2
      real              mdensity,ndensity,mass50                        moments3
c                                                                       chmdif17
      common /lognormz/ sigmazln,rad50zln,mdenzln                       lognorm2
     +                 ,ndenzln,geomzln,mas50zln                        lognorm3
      real              mdenzln,ndenzln,mas50zln                        lognorm4
c                                                                       chmdif19
      common /lognorms/ sigmasln,rad50sln,mdensln                       lognorm2
     +                 ,ndensln,geomsln,mas50sln                        lognorm3
      real              mdensln,ndensln,mas50sln                        lognorm4
c                                                                       chmdif21
      common /iotapes/  columns,ntape4,ntape5,ntape6                    iotapes2
      integer           columns                                         iotapes3
c                                                                       chmdif23
      dimension z(105)                                                  chmdif24
      dimension rwork(12325),iwork(180)                                 chmdif25
      dimension info(15)                                                chmdif26
      dimension rtol(100),atol(100)                                     chmdif27
      logical reset,firstgo                                             chmdif28
      data firstgo/.true./                                              chmdif29
      external charmrhs                                                 chmdif30
c                                                                       chmdif31
c     Set-up input for DEBDF                                            chmdif32
c     **********************                                            chmdif33
c                                                                       chmdif34
      if(firstgo)then                                                   chmdif35
      timein=0.e0                                                       chmdif36
      else                                                              chmdif37
      if(reset)timein=timeout                                           chmdif38
      endif                                                             chmdif39
      timeout=time                                                      chmdif40
c                                                                       chmdif41
c     Initial number density distribution and deposited mass            chmdif42
c     ******************************************************            chmdif43
c                                                                       chmdif44
c     z(ncoll+1) holds the integrated mass deposited on floors          chmdif45
c     z(ncoll+2) holds the integrated mass deposited on walls           chmdif46
c     z(ncoll+3) holds the integrated mass deposited on ceilings        chmdif47
c     z(ncoll+4) holds the integrated source mass                       chmdif48
c     z(ncoll+5) holds the integrated leaked mass                       chmdif49
c                                                                       chmdif50
      if(firstgo)then                                                   chmdif51
         do 100 icoll=1,ncoll                                           chmdif52
         z(icoll)=zstore(icoll)                                         chmdif53
100      continue                                                       chmdif54
c                                                                       chmdif55
         neqns=ncoll+5                                                  chmdif56
         do 105 ieqns=ncoll+1,neqns                                     chmdif57
         z(ieqns) = 0.e0                                                chmdif58
105      continue                                                       chmdif59
```

```
      endif                                                              chmdif80
c                                                                        chmdif81
c     Information for DEBDF                                              chmdif82
c     *********************                                              chmdif83
c                                                                        chmdif84
c        Flag first/reset call or subsequent call                        chmdif85
c        Reset is set whenever the independent variables change          chmdif86
c        discontinuously so that derivatives are calculated afresh on    chmdif87
c        passing the discontinuity.                                      chmdif88
c                                                                        chmdif89
         if(firstgo .or. reset)then                                      chmdif70
         info(1)=0                                                       chmdif71
         else                                                            chmdif72
         info(1)=1                                                       chmdif73
         endif                                                           chmdif74
c                                                                        chmdif75
c        Both tolerances are vector                                      chmdif76
c                                                                        chmdif77
         if(firstgo)then                                                 chmdif78
         info(2)=1                                                       chmdif79
c                                                                        chmdif80
c        The solution is not required at intermediate times              chmdif81
c                                                                        chmdif82
         info(3)=0                                                       chmdif83
c                                                                        chmdif84
c        The integration can be done without restriction on t            chmdif85
c                                                                        chmdif86
         info(4)=0                                                       chmdif87
c                                                                        chmdif88
c        Partial derivatives should be calculated by differencing        chmdif89
c                                                                        chmdif90
         info(5)=0                                                       chmdif91
c                                                                        chmdif92
c        The Jacobian is dense                                           chmdif93
c                                                                        chmdif94
         info(6)=0                                                       chmdif95
         endif                                                           chmdif96
c                                                                        chmdif97
c        info(7) to info(15) are not used by DEBDF                       chmdif98
c                                                                        chmdif99
c     Set-up relative and absolute tolerances                            chmdi100
c     ***************************************                            chmdi101
c                                                                        chmdi102
c     atol is chosen to obtain accuracy in both the number density       chmdi103
c     and mass density distributions. rtol is used as a trap in          chmdi104
c     case ndensity or mdensity decrease by large amounts during         chmdi105
c     the time step.                                                     chmdi106
c                                                                        chmdi107
      do 150 icoll=1,ncoll                                               chmdi108
      rtol(icoll)=eps                                                    chmdi109
      if(mdensity.eq.0.e0)then                                           chmdi110
      atol(icoll)=min( mdensln*(timeout-timein)/mass(icoll)              chmdi111
     +               ,ndensln*(timeout-timein) )*eps/dlogem              chmdi112
      else                                                               chmdi113
      atol(icoll)=min(mdensity/mass(icoll),ndensity)*eps/dlogem          chmdi114
      endif                                                              chmdi115
150   continue                                                           chmdi116
c                                                                        chmdi117
      do 155 ieqns=ncoll+1,neqns                                         chmdi118
      rtol(ieqns)=eps                                                    chmdi119
      if(mdensity.eq.0.e0)then                                           chmdi120
      atol(ieqns)=mdensln*(timeout-timein)*eps                           chmdi121
      else                                                               chmdi122
      atol(ieqns)=mdensity*eps                                           chmdi123
      endif                                                              chmdi124
155   continue                                                           chmdi125
c                                                                        chmdi126
c     Dimensions of rwork and iwork arrays.                              chmdi127
c     *************************************                              chmdi128
```

```
c                                                                       chmdi129
      if(firstgo)then                                                   chmdi130
      lrwork=12325                                                      chmdi131
      liwork=160                                                        chmdi132
         if( (250 + 10*neqns + neqns*neqns) .gt. lrwork)then            chmdi133
         write(ntape5,2000)                                             chmdi134
         write(ntape6,2000)                                             chmdi135
2000     format(4x,'*** CHARMDIF fails: LRWORK is too small ***')       chmdi136
         stop                                                           chmdi137
         endif                                                          chmdi138
         if( (55 + neqns) .gt. liwork)then                              chmdi139
         write(ntape5,2001)                                             chmdi140
         write(ntape6,2001)                                             chmdi141
2001     format(4x,'*** CHARMDIF fails: LIWORK is too small ***')       chmdi142
         stop                                                           chmdi143
         endif                                                          chmdi144
      endif                                                             chmdi145
c                                                                       chmdi146
c     Call DEBDF and check that the call was O.K.                       chmdi147
c     ******************************************                        chmdi148
c                                                                       chmdi149
      icall=0                                                           chmdi150
160   call debdf(charmrhs,neqns,timein,z,timeout,info,rtol,atol         chmdi151
     +                  ,idid,rwork,lrwork,iwork,liwork,rpar,ipar,jac)  chmdi152
      icall=icall+1                                                     chmdi153
         if(idid.eq.-1 .and. icall.lt.maxtrys)then                      chmdi154
         info(1)=1                                                      chmdi155
         goto 160                                                       chmdi156
         endif                                                          chmdi157
         if(idid.lt.2)then                                              chmdi158
         write(ntape5,1000)idid                                         chmdi159
         write(ntape6,1000)idid                                         chmdi160
1000     format(4x,'*** CHARMDIF fails: IDID is ',i3,' ***')            chmdi161
         stop                                                           chmdi162
         endif                                                          chmdi163
c                                                                       chmdi164
c     Store the answer                                                  chmdi165
c     ****************                                                  chmdi166
c                                                                       chmdi167
      do 200 icoll=1,neqns                                              chmdi168
      zstore(icoll)=z(icoll)                                            chmdi169
200   continue                                                          chmdi170
c                                                                       chmdi171
c     Kill the flag which signals first call to DEBDF                   chmdi172
c     ***********************************************                   chmdi173
c                                                                       chmdi174
      firstgo=.false.                                                   chmdi175
      return                                                            chmdi176
      end                                                               chmdi177
```

```
      subroutine charmrhs(timenow,z,dzdt,rpar,ipar)                     chmrhs 2
c                                                                       chmrhs 3
c     This subroutine calculates dz/dt arising from the agglomeration,  chmrhs 4
c     deposition, source and leak terms on the rhs of the discretized   chmrhs 5
c     equations.                                                        chmrhs 6
c                                                                       chmrhs 7
c     The dependent variable of the aerosol equation is chosen          chmrhs 8
c     to be the number density times mass. This quantity is             chmrhs 9
c     better suited to numerical quadrature.                            chmrhs10
c                                                                       chmrhs11
c     The mass deposited on floors is integrated in z(ncoll+1)          chmrhs12
c     The mass deposited on walls is integrated in z(ncoll+2)           chmrhs13
c     The mass deposited on ceilings is integrated in z(ncoll+3)        chmrhs14
c     The source mass is integrated in z(ncoll+4)                       chmrhs15
c     The leaked mass is integrated in z(ncoll+5)                       chmrhs16
c                                                                       chmrhs17
      common /collpts/  ncoll,mlower,mupper,spacing,dlogem,logem0       collpts2
     +                 ,radius(100),mass(100),mobility(100)            collpts3
      real              logem0,mass,mobility,mlower,mupper              collpts4
c                                                                       chmrhs19
      common /agglom/   agglomrt(100,100),deposrtf(100)                agglom 2
     +                 ,deposrtw(100),deposrtc(100)                    agglom 3
c                                                                       chmrhs21
      common /indexcoe/ nelement,hwidth,jbarmin,index(-2:100)           indexco2
     +                 ,kbarmin(-2:100),kbarmax(-2:100),nkbar(-2:100)  indexco3
c                                                                       chmrhs23
      common /coef/     pijk(300),njk(100,100)                          coef   2
      real              njk                                             coef   3
c                                                                       chmrhs25
      common /lognorms/ sigmasln,rad50sln,mdensln                       lognorm2
     +                 ,ndensln,geomsln,mas50sln                       lognorm3
      real              mdensln,ndensln,mas50sln                        lognorm4
c                                                                       chmrhs27
      common /cell/     areaclng,areawall,areaflor                      cell   2
     +                 ,tempclng,tempwall,tempflor                     cell   3
     +                 ,volume,leakrate                                cell   4
     +                 ,hydrdiam,eqvrough                              cell   5
      real              leakrate                                        cell   6
c                                                                       chmrhs29
      common /source/   sourcert(100)                                   source 2
c                                                                       chmrhs31
      common /timings/  time,istep,thhystep                             timings2
     +                 ,itime,ntime,timestep(20),timeend(20)           timings3
     +                 ,ithhy,nthhy,timethhy(20)                       timings4
     +                 ,idata,ndata,timedata(20)                       timings5
c                                                                       chmrhs33
      dimension z(105),dzdt(105)                                        chmrhs34
c                                                                       chmrhs35
c     Update time dependent data if required                            chmrhs36
c     **************************************                            chmrhs37
c                                                                       chmrhs38
      if(thhystep.eq.0.e0)then                                          chmrhs39
      call charmuth(timenow)                                            chmrhs40
      call charmgas                                                     chmrhs41
      call charmmob                                                     chmrhs42
      call charmflo                                                     chmrhs43
      call charmagg                                                     chmrhs44
      call charmdep                                                     chmrhs45
      call charmsln                                                     chmrhs46
      endif                                                             chmrhs47
c                                                                       chmrhs48
c     Initialize the mass counters                                      chmrhs49
c     ****************************                                      chmrhs50
c                                                                       chmrhs51
      neqns=ncoll+5                                                     chmrhs52
      do 5 ieqns=ncoll+1,neqns                                          chmrhs53
      dzdt(ieqns) = 0.e0                                                chmrhs54
5     continue                                                          chmrhs55
c                                                                       chmrhs56
```

```
c     Loop over the collocation points                                  chmrhs57
c     ********************************                                  chmrhs58
c                                                                       chmrhs59
      do 100 i=1,ncoll                                                  chmrhs60
      dzdt(i)=0.                                                        chmrhs61
c                                                                       chmrhs62
c     The production terms                                              chmrhs63
c     ********************                                              chmrhs64
c                                                                       chmrhs65
      if(i-jbarmin.ge.1)then                                            chmrhs66
      do 10 j=1,min(i-jbarmin,ncoll)                                    chmrhs67
      jbar=i-j                                                          chmrhs68
      prod=0.e0                                                         chmrhs69
         if(nkbar(jbar).ne.0.e0)then                                    chmrhs70
         do 20 kbar=kbarmin(jbar),kbarmax(jbar)                         chmrhs71
         k=i-kbar                                                       chmrhs72
            if(k.ge.1 .and. k.le.ncoll)then                             chmrhs73
            prod=prod+agglomrt(j,k)*njk(j,k)*z(k)                       chmrhs74
     +                          *pijk(index(jbar)+kbar)                 chmrhs75
            endif                                                       chmrhs76
20       continue                                                       chmrhs77
         prod=prod*z(j)*dlogem                                          chmrhs78
         endif                                                          chmrhs79
      dzdt(i)=dzdt(i)+prod                                              chmrhs80
10    continue                                                          chmrhs81
      endif                                                             chmrhs82
100   continue                                                          chmrhs83
c                                                                       chmrhs84
c     The destruction terms                                             chmrhs85
c     *********************                                             chmrhs86
c                                                                       chmrhs87
      do 200 i=1,ncoll                                                  chmrhs88
      dest=0.e0                                                         chmrhs89
      do 30 j=1,ncoll                                                   chmrhs90
      if(njk(i,j).ne.0.e0)dest=dest+agglomrt(i,j)*z(j)                  chmrhs91
30    continue                                                          chmrhs92
      dest=dest*z(i)*dlogem                                             chmrhs93
      dzdt(i)=dzdt(i)-dest                                              chmrhs94
c                                                                       chmrhs95
c     Sources and sinks                                                 chmrhs96
c     *****************                                                 chmrhs97
c                                                                       chmrhs98
      dzdt(i)=dzdt(i)-z(i)*deposrtf(i)                                  chmrhs99
      dzdt(i)=dzdt(i)-z(i)*deposrtw(i)                                  chmrh100
      dzdt(i)=dzdt(i)-z(i)*deposrtc(i)                                  chmrh101
      dzdt(i)=dzdt(i)+sourcert(i)                                       chmrh102
      dzdt(i)=dzdt(i)-z(i)*leakrate                                     chmrh103
c                                                                       chmrh104
c     Update mass counters                                              chmrh105
c     ********************                                              chmrh106
c                                                                       chmrh107
      factor = dlogem*volume*mass(i)                                    chmrh108
      dzdt(ncoll+1)=dzdt(ncoll+1)+z(i)*deposrtf(i)*factor               chmrh109
      dzdt(ncoll+2)=dzdt(ncoll+2)+z(i)*deposrtw(i)*factor               chmrh110
      dzdt(ncoll+3)=dzdt(ncoll+3)+z(i)*deposrtc(i)*factor               chmrh111
      dzdt(ncoll+4)=dzdt(ncoll+4)+sourcert(i)*factor                    chmrh112
      dzdt(ncoll+5)=dzdt(ncoll+5)+z(i)*leakrate*factor                  chmrh113
200   continue                                                          chmrh114
      return                                                            chmrh115
      end                                                               chmrh116
```

```
      subroutine charmmom                                                    chmmom 2
c                                                                            chmmom 3
c     This subroutine calculates moments of the discretized number           chmmom 4
c     density distribution using the trapezium rule.                         chmmom 5
c                                                                            chmmom 6
      common /collpts/  ncoll,mlower,mupper,spacing,dlogem,logem0            collpts2
     +                 ,radius(100),mass(100),mobility(100)                  collpts3
      real              logem0,mass,mobility,mlower,mupper                   collpts4
c                                                                            chmmom 8
      common /aerslcon/ cshpfctr,dshpfctr,stickeff                           aerslco2
     +                 ,aknudweb,qknudweb,bknudweb                           aerslco3
     +                 ,pdensity,pthrmcon                                    aerslco4
     +                 ,kbrock,cmbrock,ctbrock                               aerslco5
      real              kbrock                                               aerslco8
c                                                                            chmmom10
      common /fundcon/  pi,boltzmnn,gravitat,gasconst                        fundcon2
c                                                                            chmmom12
      common /toleranc/ eps,eta,zeta,maxcalls,maxtrys                        toleran2
c                                                                            chmmom14
      common /distrib/  zstore(105)                                          distrib2
c                                                                            chmmom16
      common /moments/  sigma,rad50,mdensity,ndensity,geommean,mass50        moments2
      real              mdensity,ndensity,mass50                             moments3
c                                                                            chmmom18
      common /indexcoe/ nelement,hwidth,jbarmin,index(-2:100)                indexco2
     +                 ,kbarmin(-2:100),kbarmax(-2:100),nkbar(-2:100)        indexco3
c                                                                            chmmom20
      common /iotapes/  columns,ntape4,ntape5,ntape8                         iotapes2
      integer           columns                                              iotapes3
c                                                                            chmmom22
      external charmm50                                                      chmmom23
c                                                                            chmmom24
c     Calculate sums - x is loge(mass)                                       chmmom25
c     ********************                                                   chmmom26
c                                                                            chmmom27
      sum1 = 0.e0                                                            chmmom28
      sum2 = 0.e0                                                            chmmom29
      sum3 = 0.e0                                                            chmmom30
      sum4 = 0.e0                                                            chmmom31
      do 10 i=1,ncoll                                                        chmmom32
      x = logem0 + i*dlogem                                                  chmmom33
      sum1 = sum1 + zstore(i)*mass(i)                                        chmmom34
      sum2 = sum2 + zstore(i)                                                chmmom35
      sum3 = sum3 + zstore(i)*x                                              chmmom36
      sum4 = sum4 + zstore(i)*x*x                                            chmmom37
10    continue                                                               chmmom38
c                                                                            chmmom39
c     Calculate moments from these sums                                      chmmom40
c     *********************************                                      chmmom41
c                                                                            chmmom42
         if(sum2.eq.0.e0)then                                                chmmom43
         sigma=0.e0                                                          chmmom44
         rad50=0.e0                                                          chmmom45
         mdensity=0.e0                                                       chmmom46
         ndensity=0.e0                                                       chmmom47
         geommean=0.e0                                                       chmmom48
         mass50=0.e0                                                         chmmom49
         return                                                              chmmom50
         endif                                                               chmmom51
      mdensity = dlogem * sum1                                               chmmom52
      ndensity = dlogem * sum2                                               chmmom53
      dummy1 = dlogem * sum3                                                 chmmom54
      geommean = exp( dummy1/ndensity )                                      chmmom55
      dummy2 = dlogem * sum4                                                 chmmom56
      dummy3 = (dummy2 - dummy1*dummy1/ndensity) /ndensity                   chmmom57
      if(dummy3.lt.0.e0)dummy3=0.e0                                          chmmom58
      sigma = exp( sqrt( dummy3 )/3.e0 )                                     chmmom59
c                                                                            chmmom60
c     Calculate mass median mass                                             chmmom61
```

```
c     ***************************                                    chmmom62
c                                                                    chmmom63
c     set-up input for c05whe - ifail = 1 is the soft fail option    chmmom64
c                                                                    chmmom65
      ifail = 1                                                      chmmom66
      xlower = logem0 + dlogem                                       chmmom67
      xupper = logem0 + dlogem*ncoll                                 chmmom68
      rtol = eps                                                     chmmom69
      atol = eta                                                     chmmom70
      ztol = zeta                                                    chmmom71
      ncalls = maxcalls                                              chmmom72
      call c05whe                                                    chmmom73
     +    (xlower,xupper,rtol,atol,charmm50,ncalls,ztol,x50,ifail)   chmmom74
          if(ifail.ne.0)then                                         chmmom75
          write(ntape5,1000)ifail                                    chmmom76
          write(ntape6,1000)ifail                                    chmmom77
1000      format(4x,'*** CHARMMOM fails: IFAIL is ',i2,' ***')       chmmom78
          stop                                                       chmmom79
          endif                                                      chmmom80
      mass50 = exp( x50 )                                            chmmom81
      rad50 = (3.e0*mass50/(4.e0*pi*pdensity))**(1.e0/3.e0)          chmmom82
      return                                                         chmmom83
      end                                                            chmmom84
```

```
      function charmm50(x)                                                  chmm50 2
c                                                                           chmm50 3
c     This function subroutine calculates the mass density of the           chmm50 4
c     discretised distribution up to x and subtracts                        chmm50 5
c     half the total mass density. The result is zero when                  chmm50 6
c     x is x50: x is loge(mass).                                            chmm50 7
c                                                                           chmm50 8
      common /collpts/  ncoll,mlower,mupper,spacing,dlogem,logem0           collpts2
     +                 ,radius(100),mass(100),mobility(100)                 collpts3
      real              logem0,mass,mobility,mlower,mupper                  collpts4
c                                                                           chmm5010
      common /moments/  sigma,rad50,mdensity,ndensity,geommean,mass50       moments2
      real              mdensity,ndensity,mass50                            moments3
c                                                                           chmm5012
      common /distrib/  zstore(105)                                         distrib2
c                                                                           chmm5014
      sum = 0.e0                                                            chmm5015
      do 10 i=1,ncoll                                                       chmm5016
      sum = sum + zstore(i)*mass(i)*charmfe0(x,i)                           chmm5017
10    continue                                                              chmm5018
      charmm50 = dlogem*sum - 0.5e0*mdensity                                chmm5019
      return                                                                chmm5020
      end                                                                   chmm5021
```

```
      function charmfe0(arg,k)                                          chmfe0 2
c                                                                       chmfe0 3
c     This function subroutine calculates the integral of the k-th      chmfe0 4
c     finite element up to ARG, where ARG is loge(mass). As for         chmfe0 5
c     CHARMFE, ARG is scaled by the collocation interval and            chmfe0 6
c     translated so that the element is centred about zero.             chmfe0 7
c                                                                       chmfe0 8
c     The elements are symmetric and have total integral unity          chmfe0 9
c     so only the integral from zero up to the absolute value           chmfe010
c     of the scaled argument is calculated.                             chmfe011
c                                                                       chmfe012
c     These integrals are calculated from analytic formulae since       chmfe013
c     they may be done many times. They are used in the calculation     chmfe014
c     of the mass median mass of the discretized distribution.          chmfe015
c                                                                       chmfe016
c     The choice of element is determined by NELEMENT                   chmfe017
c                                                                       chmfe018
      common /collpts/  ncoll,mlower,mupper,spacing,dlogem,logem0       collpts2
     +                 ,radius(100),mass(100),mobility(100)            collpts3
      real              logem0,mass,mobility,mlower,mupper              collpts4
c                                                                       chmfe020
      common /indexcoe/ nelement,hwidth,jbarmin,index(-2:100)           indexco2
     +                 ,kbarmin(-2:100),kbarmax(-2:100),nkbar(-2:100)  indexco3
c                                                                       chmfe022
      real integral                                                     chmfe023
c                                                                       chmfe024
c     Transform the argument and select an element                      chmfe025
c     ********************************************                      chmfe026
c                                                                       chmfe027
      if(k.ne.0) x = (arg - logem0)/dlogem - k                          chmfe028
      if(k.eq.0) x = arg                                                chmfe029
      if(x.le.-hwidth)then                                              chmfe030
         charmfe0 = 0.e0                                                chmfe031
         return                                                         chmfe032
      endif                                                             chmfe033
      if(x.ge.hwidth)then                                               chmfe034
         charmfe0 = 1.e0                                                chmfe035
         return                                                         chmfe036
      endif                                                             chmfe037
      y = abs(x)                                                        chmfe038
      goto(1,2,3,4,5,6)nelement                                         chmfe039
c                                                                       chmfe040
c     First order element                                               chmfe041
c     *******************                                               chmfe042
c                                                                       chmfe043
1     continue                                                          chmfe044
      charmfe0 = 0.5e0 + x                                              chmfe045
      return                                                            chmfe046
c                                                                       chmfe047
c     Second order element                                              chmfe048
c     ********************                                              chmfe049
c                                                                       chmfe050
2     continue                                                          chmfe051
      integral = y - y*y/2.e0                                           chmfe052
      charmfe0 = 0.5e0 + sign(integral,x)                               chmfe053
      return                                                            chmfe054
c                                                                       chmfe055
c     Third order element                                               chmfe056
c     *******************                                               chmfe057
c                                                                       chmfe058
3     continue                                                          chmfe059
      z = y - 1.e0                                                      chmfe060
      if(y.lt.2.e0 .and. y.ge.1.e0)                                     chmfe061
     +      integral=13.e0/24.e0-z*z*z/6.e0+z*z*z*z/8.e0                chmfe062
      if(y.lt.1.e0)                                                     chmfe063
     +      integral=y-5.e0*y*y*y/6.e0+3.e0*y*y*y*y/8.e0                chmfe064
      charmfe0 = 0.5e0 + sign(integral,x)                               chmfe065
      return                                                            chmfe066
c                                                                       chmfe067
```

```
c     Fourth order element                                                    chmfe068
c     ********************                                                    chmfe069
c                                                                             chmfe070
4     continue                                                                chmfe071
      z = y - 1.e0                                                            chmfe072
      if(y.lt.2.e0 .and. y.ge.1.e0)                                           chmfe073
     +      integral=13.e0/24.e0-3.e0*z*z*z*z/8.e0+z*z*z*z*z/2.e0             chmfe074
     +      -z*z*z*z*z*z/6.e0                                                 chmfe075
      if(y.lt.1.e0)                                                           chmfe076
     +      integral=y-y*y*y/3.e0-9.e0*y*y*y*y/8.e0+3.e0*y*y*y*y*y/2.e0       chmfe077
     +      -y*y*y*y*y*y/2.e0                                                 chmfe078
      charmfe0 = 0.5e0 + sign(integral,x)                                     chmfe079
      return                                                                  chmfe080
c                                                                             chmfe081
c     Fifth order element                                                     chmfe082
c     *******************                                                     chmfe083
c                                                                             chmfe084
5     continue                                                                chmfe085
      z = y - 1.e0                                                            chmfe086
      u = y - 2.e0                                                            chmfe087
      if(y.lt.3.e0 .and. y.ge.2.e0)                                           chmfe088
     +      integral=1681.e0/3360.e0-u*u*u*u*u/60.e0+u*u*u*u*u*u/24.e0        chmfe089
     +      -u*u*u*u*u*u*u/28.e0+u*u*u*u*u*u*u*u/96.e0                        chmfe090
      if(y.lt.2.e0 .and. y.ge.1.e0)                                           chmfe091
     +      integral=303.e0/560.e0+z*z*z*z/48.e0-31.e0*z*z*z*z*z/30.e0        chmfe092
     +      +25.e0*z*z*z*z*z*z/12.e0-31.e0*z*z*z*z*z*z*z/21.e0                chmfe093
     +      +35.e0*z*z*z*z*z*z*z*z/96.e0                                      chmfe094
      if(y.lt.1.e0)                                                           chmfe095
     +      integral=y-y*y*y/3.e0-91.e0*y*y*y*y*y/30.e0                       chmfe096
     +      +19.e0*y*y*y*y*y*y/3.e0-32.e0*y*y*y*y*y*y*y/7.e0                  chmfe097
     +      +55.e0*y*y*y*y*y*y*y*y/48.e0                                      chmfe098
      charmfe0 = 0.5e0 + sign(integral,x)                                     chmfe099
      return                                                                  chmfe100
c                                                                             chmfe101
c     Sixth order element                                                     chmfe102
c     *******************                                                     chmfe103
c                                                                             chmfe104
6     continue                                                                chmfe105
      z = y - 1.e0                                                            chmfe106
      u = y - 2.e0                                                            chmfe107
      if(y.lt.3.e0 .and. y.ge.2.e0)                                           chmfe108
     +      integral=1081.e0/2160.e0-11.e0*u*u*u*u*u*u/144.e0                 chmfe109
     +      +u*u*u*u*u*u*u/4.e0-5.e0*u*u*u*u*u*u*u/16.e0                      chmfe110
     +      +19.e0*u*u*u*u*u*u*u*u/108.e0                                     chmfe111
     +      -3.e0*u*u*u*u*u*u*u*u*u/80.e0                                     chmfe112
      if(y.lt.2.e0 .and. y.ge.1.e0)                                           chmfe113
     +      integral=73.e0/135.e0+z*z*z*z*z/48.e0+z*z*z*z*z*z/120.e0          chmfe114
     +      -145.e0*z*z*z*z*z*z/48.e0+103.e0*z*z*z*z*z*z*z/12.e0              chmfe115
     +      -115.e0*z*z*z*z*z*z*z*z/12.e0+355.e0*z*z*z*z*z*z*z*z*z/72.e0      chmfe116
     +      -47.e0*z*z*z*z*z*z*z*z*z*z/48.e0                                  chmfe117
      if(y.lt.1.e0)                                                           chmfe118
     +      integral=y-y*y*y/3.e0+y*y*y*y*y/20.e0                             chmfe119
     +      -659.e0*y*y*y*y*y*y/72.e0+371.e0*y*y*y*y*y*y*y/14.e0              chmfe120
     +      -30.e0*y*y*y*y*y*y*y*y+1685.e0*y*y*y*y*y*y*y*y/108.e0             chmfe121
     +      -25.e0*y*y*y*y*y*y*y*y*y/8.e0                                     chmfe122
      charmfe0 = 0.5e0 + sign(integral,x)                                     chmfe123
      return                                                                  chmfe124
      end                                                                     chmfe125
```

```
      subroutine charmout                                                    chmout 2
c                                                                            chmout 3
c     This subroutine writes to tapes 5 & 6 (tty & output file resp.).       chmout 4
c     No form control characters are printed.                                chmout 5
c                                                                            chmout 6
      common /collpts/  ncoll,mlower,mupper,spacing,dlogem,logem0            collpts2
     +                 ,radius(100),mass(100),mobility(100)                  collpts3
      real              logem0,mass,mobility,mlower,mupper                   collpts4
c                                                                            chmout 8
      common /aerslcon/ cshpfctr,dshpfctr,stickeff                           aerslco2
     +                 ,aknudweb,qknudweb,bknudweb                         aerslco3
     +                 ,pdensity,pthrmcon                                  aerslco4
     +                 ,kbrock,cmbrock,ctbrock                             aerslco5
      real              kbrock                                               aerslco6
c                                                                            chmout10
      common /gasprops/ temp,press,gdensity,dynvisc,molwt,mnfrpath           gasprop2
     +                 ,gthrmcon,velocity,molwtv,diffusv                   gasprop3
     +                 ,vmfrclng,vmfrwall,vmfrflor                         gasprop4
     +                 ,vcgrclng,vcgrwall,vcgrflor                         gasprop5
     +                 ,vconclng,vconwall,vconflor                         gasprop6
      real              molwt,mnfrpath,molwtv                                gasprop7
c                                                                            chmout12
      common /agglom/   agglomrt(100,100),deposrtf(100)                     agglom 2
     +                 ,deposrtw(100),deposrtc(100)                        agglom 3
c                                                                            chmout14
      common /indexcoe/ nelement,hwidth,jbarmin,index(-2:100)                indexco2
     +                 ,kbarmin(-2:100),kbarmax(-2:100),nkbar(-2:100)      indexco3
c                                                                            chmout16
      common /coef/     pijk(300),njk(100,100)                               coef   2
      real              njk                                                  coef   3
c                                                                            chmout18
      common /distrib/  zstore(105)                                          distrib2
c                                                                            chmout20
      common /lognorms/ sigmasln,rad50sln,mdensln                            lognorm2
     +                 ,ndensln,geomsln,mas50sln                           lognorm3
      real              mdensln,ndensln,mas50sln                             lognorm4
c                                                                            chmout22
      common /source/   sourcert(100)                                        source 2
c                                                                            chmout24
      common /timings/  time,istep,thhystep                                  timings2
     +                 ,itime,ntime,timestep(20),timeend(20)                timings3
     +                 ,ithhy,nthhy,timethhy(20)                            timings4
     +                 ,idata,ndata,timedata(20)                            timings5
c                                                                            chmout26
      common /toleranc/ eps,eta,zeta,maxcalls,maxtrys                        toleran2
c                                                                            chmout28
      common /moments/  sigma,rad50,mdensity,ndensity,geommean,mass50        moments2
      real              mdensity,ndensity,mass50                             moments3
c                                                                            chmout30
      common /fundcon/  pi,boltzmnn,gravitat,gasconst                        fundcon2
c                                                                            chmout32
      common /ioflags/  iondis,iozmom,iocoef,ionorm                          ioflags2
     +                 ,iodepo,iomass,ioradi,iomobi                        ioflags3
     +                 ,ioaggl,iomdis,iosdis,iombal                        ioflags4
     +                 ,ioindx,iomom,iocell,iogasp                         ioflags5
     +                 ,iotole,ioacon,ioflow                               ioflags6
c                                                                            chmout34
      common /iotapes/  columns,ntape4,ntape5,ntape6                         iotapes2
      integer           columns                                              iotapes3
c                                                                            chmout36
      common /cell/     areaclng,areawall,areaflor                           cell   2
     +                 ,tempclng,tempwall,tempflor                          cell   3
     +                 ,volume,leakrate                                     cell   4
     +                 ,hydrdiam,eqvrough                                   cell   5
      real              leakrate                                             cell   6
c                                                                            chmout38
      common /flow/     blflag,vblthick,dblthick(100)                        flow   2
     +                 ,eddydiss,ustar,reynolds                             flow   3
      integer           blflag                                               flow   4
```

```
c                                                                       chmout40
      logical firstgo                                                   chmout41
      data firstgo/.true./                                              chmout42
c                                                                       chmout43
c     Rule-off page first time round                                    chmout44
c     ******************************                                    chmout45
c                                                                       chmout46
      if(firstgo)then                                                   chmout47
      if(columns.eq. 80)write(ntape6,9999)                              chmout48
      if(columns.eq.132)write(ntape6,8999)                              chmout49
      endif                                                             chmout50
c                                                                       chmout51
c     Current time and mass balance                                     chmout52
c     *****************************                                     chmout53
c                                                                       chmout54
      if(firstgo)balance0 = mdensity*volume                             chmout55
      balance = mdensity*volume                                         chmout56
     +        + zstore(ncoll+1)                                         chmout57
     +        + zstore(ncoll+2)                                         chmout58
     +        + zstore(ncoll+3)                                         chmout59
     +        - zstore(ncoll+4)                                         chmout60
     +        + zstore(ncoll+5)                                         chmout61
     +        - balance0                                                chmout62
      write(ntape5,9009)istep,time,balance                              chmout63
      write(ntape6,9038)istep,time,balance                              chmout64
c                                                                       chmout65
c     Mass balances                                                     chmout66
c     *************                                                     chmout67
c                                                                       chmout68
      if(iombal.ne.0)then                                               chmout69
      if(mod(istep,iombal).eq.0)then                                    chmout70
      write(ntape6,9010)                                                chmout71
      write(ntape6,9001)mdensity*volume,zstore(ncoll+1)                 chmout72
     +             ,zstore(ncoll+2),zstore(ncoll+3)                     chmout73
     +             ,zstore(ncoll+4),zstore(ncoll+5)                     chmout74
      endif                                                             chmout75
      endif                                                             chmout76
c                                                                       chmout77
c     Sound alarm if too much mass at end collocation points            chmout78
c     ******************************************************            chmout79
c                                                                       chmout80
      if(mdensity.ne.0.e0)then                                          chmout81
      fraction = zstore(ncoll)*mass(ncoll)*dlogem/mdensity              chmout82
      if(fraction.gt.1.e-1)then                                         chmout83
      write(ntape5,9015)fraction                                        chmout84
      write(ntape6,9015)fraction                                        chmout85
      endif                                                             chmout86
      fraction = zstore(1)*mass(1)*dlogem/mdensity                      chmout87
      if(fraction.gt.1.e-1)then                                         chmout88
      write(ntape5,9053)fraction                                        chmout89
      write(ntape6,9053)fraction                                        chmout90
      endif                                                             chmout91
      endif                                                             chmout92
c                                                                       chmout93
c     Sound alarm if too many particles at end collocation points       chmout94
c     ***********************************************************       chmout95
c                                                                       chmout96
      if(ndensity.ne.0.e0)then                                          chmout97
      fraction = zstore(ncoll)*dlogem/ndensity                          chmout98
      if(fraction.gt.1.e-1)then                                         chmout99
      write(ntape5,9054)fraction                                        chmou100
      write(ntape6,9054)fraction                                        chmou101
      endif                                                             chmou102
      fraction = zstore(1)*dlogem/ndensity                              chmou103
      if(fraction.gt.1.e-1)then                                         chmou104
      write(ntape5,9055)fraction                                        chmou105
      write(ntape6,9055)fraction                                        chmou106
      endif                                                             chmou107
      endif                                                             chmou108
```

```
c                                                                       chmou109
c     Moments                                                           chmou110
c     *******                                                           chmou111
c                                                                       chmou112
      if(iozmom.ne.0)then                                               chmou113
      if(mod(istep,iozmom).eq.0)then                                    chmou114
      write(ntape6,9002)                                                chmou115
      write(ntape6,9001)sigma,rad50,mdensity,ndensity,geommean,mass50   chmou116
      endif                                                             chmou117
      endif                                                             chmou118
c                                                                       chmou119
c     Source moments                                                    chmou120
c     **************                                                    chmou121
c                                                                       chmou122
      if(iosmom.ne.0)then                                               chmou123
      if(mod(istep,iosmom).eq.0)then                                    chmou124
      write(ntape6,9021)                                                chmou125
      write(ntape6,9001)                                                chmou126
     +   sigmasln,rad50sln,mdensln,ndensln,geomsln,mas50sln             chmou127
      endif                                                             chmou128
      endif                                                             chmou129
c                                                                       chmou130
c     Mass density distribution                                         chmou131
c     *************************                                         chmou132
c                                                                       chmou133
      if(iomdis.ne.0)then                                               chmou134
      if(mod(istep,iomdis).eq.0)then                                    chmou135
      write(ntape6,9044)                                                chmou136
      if(columns.eq. 80)write(ntape6,9018)                              chmou137
      if(columns.eq.132)write(ntape6,8018)                              chmou138
      if(columns.eq. 80)                                                chmou139
     +                  write(ntape6,9003)(i,zstore(i)*mass(i),i=1,ncoll)chmou140
      if(columns.eq.132)                                                chmou141
     +                  write(ntape6,8003)(i,zstore(i)*mass(i),i=1,ncoll)chmou142
      endif                                                             chmou143
      endif                                                             chmou144
c                                                                       chmou145
c     Number density distribution                                       chmou146
c     ***************************                                       chmou147
c                                                                       chmou148
      if(iondis.ne.0)then                                               chmou149
      if(mod(istep,iondis).eq.0)then                                    chmou150
      write(ntape6,9045)                                                chmou151
      if(columns.eq. 80)write(ntape6,9017)                              chmou152
      if(columns.eq.132)write(ntape6,8017)                              chmou153
      if(columns.eq. 80)write(ntape6,9003)(i,zstore(i),i=1,ncoll)       chmou154
      if(columns.eq.132)write(ntape6,8003)(i,zstore(i),i=1,ncoll)       chmou155
      endif                                                             chmou156
      endif                                                             chmou157
c                                                                       chmou158
c     Source mass and number density distributions                      chmou159
c     ********************************************                      chmou160
c                                                                       chmou161
      if(iosdis.ne.0)then                                               chmou162
      if(mod(istep,iosdis).eq.0)then                                    chmou163
      write(ntape6,9022)                                                chmou164
      if(columns.eq. 80)write(ntape6,9018)                              chmou165
      if(columns.eq.132)write(ntape6,8018)                              chmou166
      if(columns.eq. 80)                                                chmou167
     +                  write(ntape6,9003)(i,sourcert(i)*mass(i),i=1,ncoll)chmou168
      if(columns.eq.132)                                                chmou169
     +                  write(ntape6,8003)(i,sourcert(i)*mass(i),i=1,ncoll)chmou170
      write(ntape6,9039)                                                chmou171
      if(columns.eq. 80)write(ntape6,9017)                              chmou172
      if(columns.eq.132)write(ntape6,8017)                              chmou173
      if(columns.eq. 80)write(ntape6,9003)(i,sourcert(i),i=1,ncoll)     chmou174
      if(columns.eq.132)write(ntape6,8003)(i,sourcert(i),i=1,ncoll)     chmou175
      endif                                                             chmou176
      endif                                                             chmou177
```

```
c                                                                       chmou178
c     Element, number of collocation points etc.                        chmou179
c     ******************************************                        chmou180
c                                                                       chmou181
      if(firstgo)then                                                   chmou182
      range = mupper/mlower                                             chmou183
      write(ntape6,9004)                                                chmou184
      write(ntape6,9005)nelement,ncoll,hwidth,spacing,range             chmou185
      endif                                                             chmou186
c                                                                       chmou187
c     Indexing for production coefficients                              chmou188
c     ************************************                              chmou189
c                                                                       chmou190
      if(firstgo .and. ioindx.ne.0)then                                 chmou191
      write(ntape6,9023)                                                chmou192
      write(ntape6,9024)(jbar,kbarmin(jbar),kbarmax(jbar),nkbar(jbar)   chmou193
     +                  ,index(jbar),jbar=jbarmin,ncoll-1)              chmou194
      endif                                                             chmou195
c                                                                       chmou196
c     Production coefficients                                           chmou197
c     ***********************                                           chmou198
c                                                                       chmou199
      if(firstgo .and. iocoef.ne.0)then                                 chmou200
      if(columns.eq. 80)write(ntape6,9007)                              chmou201
      if(columns.eq.132)write(ntape6,8007)                              chmou202
      if(columns.eq. 80)write(ntape6,9001)(pijk(ind),ind=1,index(ncoll))chmou203
      if(columns.eq.132)write(ntape6,8001)(pijk(ind),ind=1,index(ncoll))chmou204
      endif                                                             chmou205
c                                                                       chmou206
c     Normalization                                                     chmou207
c     *************                                                     chmou208
c                                                                       chmou209
      if(firstgo .and. ionorm.ne.0)then                                 chmou210
      if(columns.eq. 80)write(ntape6,9008)                              chmou211
      if(columns.eq.132)write(ntape6,8008)                              chmou212
      do 10 j=1,ncoll                                                   chmou213
      if(columns.eq. 80)write(ntape6,9003)(k,njk(j,k),k=1,ncoll)        chmou214
      if(columns.eq.132)write(ntape6,8003)(k,njk(j,k),k=1,ncoll)        chmou215
10    continue                                                          chmou216
      endif                                                             chmou217
c                                                                       chmou218
c     Collocation points                                                chmou219
c     ******************                                                chmou220
c                                                                       chmou221
      if(firstgo .and. iomass.ne.0)then                                 chmou222
      if(columns.eq. 80)write(ntape6,9006)                              chmou223
      if(columns.eq.132)write(ntape6,8006)                              chmou224
      if(columns.eq. 80)write(ntape6,9003)(i,mass(i),i=1,ncoll)         chmou225
      if(columns.eq.132)write(ntape6,8003)(i,mass(i),i=1,ncoll)         chmou226
      endif                                                             chmou227
c                                                                       chmou228
c     Radii at the collocation points                                   chmou229
c     *******************************                                   chmou230
c                                                                       chmou231
      if(firstgo .and. ioradi.ne.0)then                                 chmou232
      if(columns.eq. 80)write(ntape6,9011)                              chmou233
      if(columns.eq.132)write(ntape6,8011)                              chmou234
      if(columns.eq. 80)write(ntape6,9003)(i,radius(i),i=1,ncoll)       chmou235
      if(columns.eq.132)write(ntape6,8003)(i,radius(i),i=1,ncoll)       chmou236
      endif                                                             chmou237
c                                                                       chmou238
c     Tolerances                                                        chmou239
c     **********                                                        chmou240
c                                                                       chmou241
      if(firstgo .and. iotole.ne.0)then                                 chmou242
      write(ntape6,9025)                                                chmou243
      write(ntape6,9026)eps,eta,zeta,maxcalls,maxtrys                   chmou244
      endif                                                             chmou245
c                                                                       chmou246
```

```
c     Aerosol physics data                                                    chmou247
c     ********************                                                    chmou248
c                                                                             chmou249
      if(firstgo .and. ioacon.ne.0)then                                       chmou250
      write(ntape6,9040)                                                      chmou251
      write(ntape6,9041)cshpfctr,dshpfctr,stickeff                            chmou252
      write(ntape6,9027)aknudweb,qknudweb,bknudweb                            chmou253
      write(ntape6,9028)pdensity,pthrmcon                                     chmou254
      write(ntape6,9029)kbrock,cmbrock,ctbrock                                chmou255
      endif                                                                   chmou256
c                                                                             chmou257
c     Cell data                                                               chmou258
c     *********                                                               chmou259
c                                                                             chmou260
      if(iocell.ne.0)then                                                     chmou261
      if(mod(istep,iocell).eq.0)then                                          chmou262
      write(ntape6,9042)                                                      chmou263
      endif                                                                   chmou264
      endif                                                                   chmou265
      if(firstgo .and. iocell.ne.0)then                                       chmou266
      write(ntape6,9030)volume,leak .te                                       chmou267
      write(ntape6,9048)hydrdiam,eqvrough                                     chmou268
      write(ntape6,9031)areaclng,areawall,areaflor                            chmou269
      endif                                                                   chmou270
      if(iocell.ne.0)then                                                     chmou271
      if(mod(istep,iocell).eq.0)then                                          chmou272
      write(ntape6,9032)tempclng,tempwall,tempflor                            chmou273
      endif                                                                   chmou274
      endif                                                                   chmou275
c                                                                             chmou276
c     Gas data                                                                chmou277
c     ********                                                                chmou278
c                                                                             chmou279
      if(iogasp.ne.0)then                                                     chmou280
      if(mod(istep,iogasp).eq.0)then                                          chmou281
      write(ntape6,9043)                                                      chmou282
      write(ntape6,9033)temp,press,velocity                                   chmou283
      write(ntape6,9034)gdensity,dynvisc,mnfrpath                             chmou284
      if(firstgo)then                                                         chmou285
      write(ntape6,9038)molwt,gthrmcon                                        chmou286
      write(ntape6,9041)molwtv,diffusv                                        chmou287
      write(ntape6,9049)vmfrclng,vmfrwall,vmfrflor                            chmou288
      write(ntape6,9050)vcgrclng,vcgrwall,vcgrflor                            chmou289
      write(ntape6,9052)vconclng,vconwall,vconflor                            chmou290
      endif                                                                   chmou291
      endif                                                                   chmou292
      endif                                                                   chmou293
c                                                                             chmou294
c     Flow data                                                               chmou295
c     *********                                                               chmou296
c                                                                             chmou297
      if(ioflow.ne.0)then                                                     chmou298
      if(mod(istep,ioflow).eq.0)then                                          chmou299
      write(ntape6,9046)                                                      chmou300
      write(ntape6,9035)eddydiss,ustar,vblthick                               chmou301
      write(ntape6,9047)                                                      chmou302
      if(columns.eq. 80)write(ntape6,9003)(i,dblthick(i),i=1,ncell)           chmou303
      if(columns.eq.132)write(ntape6,8003)(i,dblthick(i),i=1,ncell)           chmou304
      endif                                                                   chmou305
      endif                                                                   chmou306
c                                                                             chmou307
c     Mobilities at the collocation points                                    chmou308
c     ************************************                                    chmou309
c                                                                             chmou310
      if(iomobi.ne.0)then                                                     chmou311
      if(mod(istep,iomobi).eq.0)then                                          chmou312
      if(columns.eq. 80)write(ntape6,9012)                                    chmou313
      if(columns.eq.132)write(ntape6,8012)                                    chmou314
      if(columns.eq. 80)write(ntape6,9003)(i,mobility(i),i=1,ncell)           chmou315
```

```
      if(columns.eq.132)write(ntape6,8003)(i,mobility(i),i=1,ncoll)      chmou316
      endif                                                              chmou317
      endif                                                              chmou318
c                                                                        chmou319
c     Deposition rates at the collocation points                         chmou320
c     ******************************************                         chmou321
c                                                                        chmou322
      if(iodepo.ne.0)then                                                chmou323
      if(mod(istep,iodepo).eq.0)then                                     chmou324
      write(ntape6,9018)                                                 chmou325
      if(columns.eq. 80)write(ntape6,9013)                               chmou326
      if(columns.eq.132)write(ntape6,8013)                               chmou327
      if(columns.eq. 80)write(ntape6,9003)(i,deposrtf(i),i=1,ncoll)      chmou328
      if(columns.eq.132)write(ntape6,8003)(i,deposrtf(i),i=1,ncoll)      chmou329
      write(ntape6,9019)                                                 chmou330
      if(columns.eq. 80)write(ntape6,9013)                               chmou331
      if(columns.eq.132)write(ntape6,8013)                               chmou332
      if(columns.eq. 80)write(ntape6,9003)(i,deposrtw(i),i=1,ncoll)      chmou333
      if(columns.eq.132)write(ntape6,8003)(i,deposrtw(i),i=1,ncoll)      chmou334
      write(ntape6,9020)                                                 chmou335
      if(columns.eq. 80)write(ntape6,9013)                               chmou336
      if(columns.eq.132)write(ntape6,8013)                               chmou337
      if(columns.eq. 80)write(ntape6,9003)(i,deposrtc(i),i=1,ncoll)      chmou338
      if(columns.eq.132)write(ntape6,8003)(i,deposrtc(i),i=1,ncoll)      chmou339
      endif                                                              chmou340
      endif                                                              chmou341
c                                                                        chmou342
c     Agglomeration rates at the collocation points                      chmou343
c     *********************************************                      chmou344
c                                                                        chmou345
      if(ioaggl.ne.0)then                                                chmou346
      if(mod(istep,ioaggl).eq.0)then                                     chmou347
      if(columns.eq. 80)write(ntape6,9014)                               chmou348
      if(columns.eq.132)write(ntape6,8014)                               chmou349
      do 20 j=1,ncoll                                                    chmou350
      if(columns.eq. 80)write(ntape6,9003)(k,agglomrt(j,k),k=1,ncoll)    chmou351
      if(columns.eq.132)write(ntape6,8003)(k,agglomrt(j,k),k=1,ncoll)    chmou352
20    continue                                                           chmou353
      endif                                                              chmou354
      endif                                                              chmou355
c                                                                        chmou356
c     Kill FIRSTGO flag & rule-off page                                  chmou357
c     *********************************                                  chmou358
c                                                                        chmou359
      firstgo=.false.                                                    chmou360
      if(columns.eq. 80)write(ntape6,9999)                               chmou361
      if(columns.eq.132)write(ntape6,8999)                               chmou362
      return                                                             chmou363
c                                                                        chmou364
c     Format statements                                                  chmou365
c     *****************                                                  chmou366
c                                                                        chmou367
9001  format(1p8e12.4)                                                   chmou368
9002  format('Airborne aerosol moments...',/                            chmou369
     +'      sigma  ',                                                   chmou370
     +'      rad50  ',                                                   chmou371
     +'   mdensity ',                                                    chmou372
     +'   ndensity ',                                                    chmou373
     +'   geommean ',                                                    chmou374
     +'    mass50  ')                                                    chmou375
9003  format(5(i4,1p1e12.4))                                             chmou376
9004  format('Collocation information...',/                              chmou377
     +' nelement ',                                                      chmou378
     +'  ncoll   ',                                                      chmou379
     +'    hwidth  ',4x,                                                 chmou380
     +'    spacing ',4x,                                                 chmou381
     +'     range  ')                                                    chmou382
9005  format(2(i6,4x),3(1p1e12.4,4x))                                    chmou383
9006  format('Collocation points...',/                                   chmou384
```

```
     +5(4x,'     mass    '))                                                    chmou385
9007  format(                                                                   chmou386
     +'Production coefficients in indexing order...',/                         chmou387
     +6('     pijk    '))                                                      chmou388
9008  format('Normalization factors...',/                                       chmou389
     +5(4x,'      njk    '))                                                   chmou390
9009  format(                                                                   chmou391
     +'step no. = ',i3,'   time =',1p1e12.4,'   mass check =',1p1e12.4)        chmou392
9010  format('Mass budget...',/                                                 chmou393
     +'   air-borne',                                                          chmou394
     +'   flor dep.',                                                          chmou395
     +'   wall dep.',                                                          chmou396
     +'   clng dep.',                                                          chmou397
     +'    source  ',                                                          chmou398
     +'    leaked  ')                                                          chmou399
9011  format('Radii at the collocation points...',/                             chmou400
     +5(4x,'    radius  '))                                                   chmou401
9012  format('Mobilities at the collocation points...',/                        chmou402
     +5(4x,'   mobility '))                                                    chmou403
9013  format(5(4x,'   dep.rate '))                                            chmou404
9014  format('Agglomeration kernel...',/                                        chmou405
     +5(4x,'   agg.rate '))                                                   chmou406
9015  format(4x,'*** CHARMOUT warning: the mass fraction in the',              chmou407
     +' top bin is ',1p1e12.4,' ***')                                          chmou408
9016  format(5(4x,'    m-distr '))                                            chmou409
9017  format(5(4x,'    n-distr '))                                            chmou410
9018  format('Rate of deposition onto floors...')                              chmou411
9019  format('Rate of deposition onto walls...')                               chmou412
9020  format('Rate of deposition onto ceilings...')                            chmou413
9021  format('Source moments...',/                                             chmou414
     +'     sigma  ',                                                         chmou415
     +'     rad50  ',                                                         chmou416
     +'   mdensity ',                                                         chmou417
     +'   ndensity ',                                                         chmou418
     +'   geommean ',                                                         chmou419
     +'    mass50  ')                                                         chmou420
9022  format('Source mass distribution...')                                    chmou421
9023  format('Indexing for production coefficients...',/                       chmou422
     +'  jbar  ',                                                            chmou423
     +' kbarmin',                                                            chmou424
     +' kbarmax',                                                            chmou425
     +'  nkbar ',                                                            chmou426
     +'  index ',                                                            chmou427
     +'  jbar  ',                                                            chmou428
     +' kbarmin',                                                            chmou429
     +' kbarmax',                                                            chmou430
     +'  nkbar ',                                                            chmou431
     +'  index ')                                                            chmou432
9024  format(i6,9i8)                                                           chmou433
9025  format('Tolerance information...',/                                      chmou434
     +'      eps   ',                                                         chmou435
     +'      eta   ',                                                         chmou436
     +'     zeta   ',                                                         chmou437
     +'   maxcalls ',                                                         chmou438
     +'    maxtrys ')                                                         chmou439
9026  format(1p3e12.4,i8,i12)                                                  chmou440
9027  format(                                                                  chmou441
     +'  aknudweb =',1p1e12.4,                                                chmou442
     +'  qknudweb =',1p1e12.4,                                                chmou443
     +'  bknudweb =',1p1e12.4)                                                chmou444
9028  format(                                                                  chmou445
     +'  pdensity =',1p1e12.4,                                                chmou446
     +'  pthrmcon =',1p1e12.4)                                                chmou447
9029  format(                                                                  chmou448
     +'    kbrock =',1p1e12.4,                                                chmou449
     +'   cmbrock =',1p1e12.4,                                                chmou450
     +'   ctbrock =',1p1e12.4)                                                chmou451
9030  format(                                                                  chmou452
     +'    volume =',1p1e12.4,                                                chmou453
```

```
     +'  leakrate =',1p1e12.4)                                                    chmou454
9031  format(                                                                    chmou455
     +'  areaclng =',1p1e12.4,                                                   chmou456
     +'  areawall =',1p1e12.4,                                                   chmou457
     +'  areaflor =',1p1e12.4)                                                   chmou458
9032  format(                                                                    chmou459
     +'  tempclng =',1p1e12.4,                                                   chmou460
     +'  tempwall =',1p1e12.4,                                                   chmou461
     +'  tempflor =',1p1e12.4)                                                   chmou462
9033  format(                                                                    chmou463
     +'       temp =',1p1e12.4,                                                  chmou464
     +'      press =',1p1e12.4,                                                  chmou465
     +'  velocity =',1p1e12.4)                                                   chmou466
9034  format(                                                                    chmou467
     +'  gdensity =',1p1e12.4,                                                   chmou468
     +'   dynvisc =',1p1e12.4,                                                   chmou469
     +'  mnfrpath =',1p1e12.4)                                                   chmou470
9035  format(                                                                    chmou471
     +'  eddydiss =',1p1e12.4,                                                   chmou472
     +'     ustar =',1p1e12.4,                                                   chmou473
     +'  vblthick =',1p1e12.4)                                                   chmou474
9036  format(                                                                    chmou475
     +'      molwt =',1p1e12.4,                                                  chmou476
     +'  gthrmcon =',1p1e12.4)                                                   chmou477
9038  format(                                                                    chmou478
     +62('*'),/                                                                 chmou479
     +'step no. = ',i3,'   time =',1p1e12.4,'   mass check =',1p1e12.4,/chmou480
     +62('*')/)                                                                 chmou481
9039  format('Source number distribution...')                                    chmou482
9040  format('Aerosol physics data...')                                          chmou483
9041  format(                                                                    chmou484
     +'  cshpfctr =',1p1e12.4,                                                   chmou485
     +'  dshpfctr =',1p1e12.4,                                                   chmou486
     +'  stickeff =',1p1e12.4)                                                   chmou487
9042  format('Cell data...')                                                     chmou488
9043  format('Gas data...')                                                      chmou489
9044  format('Airborne mass distribution...')                                    chmou490
9045  format('Airborne number distribution...')                                  chmou491
9046  format('Flow data...')                                                     chmou492
9047  format('Diffusion boundary layer thickness...',/                          chmou493
     +5(4x,'   dblthick '))                                                      chmou494
9048  format(                                                                    chmou495
     +'  hydrdiam =',1p1e12.4,                                                   chmou496
     +'  eqvrough =',1p1e12.4)                                                   chmou497
9049  format(                                                                    chmou498
     +'  vmfrclng =',1p1e12.4,                                                   chmou499
     +'  vmfrwall =',1p1e12.4,                                                   chmou500
     +'  vmfrflor =',1p1e12.4)                                                   chmou501
9050  format(                                                                    chmou502
     +'  vcgrclng =',1p1e12.4,                                                   chmou503
     +'  vcgrwall =',1p1e12.4,                                                   chmou504
     +'  vcgrflor =',1p1e12.4)                                                   chmou505
9051  format(                                                                    chmou506
     +'    molwtv =',1p1e12.4,                                                   chmou507
     +'   diffusv =',1p1e12.4)                                                   chmou508
9052  format(                                                                    chmou509
     +'  vconclng =',1p1e12.4,                                                   chmou510
     +'  vconwall =',1p1e12.4,                                                   chmou511
     +'  vconflor =',1p1e12.4)                                                   chmou512
9053  format(4x,'*** CHARMOUT warning: the mass fraction in the',               chmou513
     +' bottom bin is ',1p1e12.4,' ***')                                        chmou514
9054  format(4x,'*** CHARMOUT warning: the number fraction in the',             chmou515
     +' top bin is ',1p1e12.4,' ***')                                           chmou516
9055  format(4x,'*** CHARMOUT warning: the number fraction in the',             chmou517
     +' bottom bin is ',1p1e12.4,' ***')                                        chmou518
9999  format(/,80('*'),/)                                                        chmou519
8001  format(1p11e12.4)                                                          chmou520
8003  format(8(i4,1p1e12.4))                                                     chmou521
8006  format('Collocation points...',/                                           chmou522
```

```
     +8(4x,'     mass    '))                                          chmou523
8007  format(                                                         chmou524
     +'Production coefficients in indexing order...',/                chmou525
     +11('      pijk    '))                                          chmou526
8008  format('Normalization factors...',/                            chmou527
     +8(4x,'       njk    '))                                        chmou528
8011  format('Radii at the collocation points...',/                  chmou529
     +8(4x,'     radius   '))                                        chmou530
8012  format('Mobilities at the collocation points...',/             chmou531
     +8(4x,'    mobility '))                                         chmou532
8013  format(8(4x,'   dep.rate '))                                   chmou533
8014  format('Agglomeration kernel...',/                             chmou534
     +8(4x,'    agg.rate '))                                        chmou535
8016  format(8(4x,'     m-distr '))                                  chmou536
8017  format(8(4x,'     n-distr '))                                  chmou537
8047  format('Diffusion boundary layer thickness...',/               chmou538
     +8(4x,'   dblthick '))                                         chmou539
8999  format(/,132('*'),/)                                          chmou540
      end                                                           chmou541
```

APPENDIX E - NOMENCLATURE

a, $\tilde{a}$, $\tilde{a}_j$, $\tilde{a}_k$	Parameters in Sitarski and Seinfeld's correction to the Brownian agglomeration rate.
A	The cell cross-sectional area perpendicular to the flow.
A_c, A_w, A_f, A_s	The area exposed to aerosol of , respectively, the ceiling, wall, floor and any one of these surfaces.
b_k, b_m, b_t	Dimensionless constants appearing in the Brock factor, Br, in the expression for the thermophoresis deposition velocity.
B, B_i, B_j, B_k	The mobility of particles of any mass and those of mass m_i, m_j and m_k respectively.
Br	The Brock factor in the expression for the thermophoretic deposition velocity.
B_{St}	The particle mobility according to Stoke's law.
c_c, c_w, c_f, c_s	The vapor concentration adjacent to, respectively, the ceiling, wall, floor and any one of the surfaces.
$\frac{dc_c}{dx}$, $\frac{dc_w}{dx}$, $\frac{dc_f}{dx}$, $\frac{dc_s}{dx}$	The vapor concentration gradient adjacent to, respectively, the ceiling, wall, floor and any one of these surfaces.
$C(m,t)$	The number density distribution.
Cu	The Cunningham correction factor in the expression for the particle mobility.
d_h	The hydraulic diameter of the flow path - see Eq. 9.
D_j^i	The integral which appears in the destruction term in the discretized aerosol equation.
D_r	The diffusivity of steam in air at T_{r1} and P_r.
D_v	The vapor diffusivity in the gas.
f_*	The Fanning friction factor.
f_c, f_w, f_f, f_s	The molar fraction of vapor in the gas/vapor mixture adjacent to, respectively, the ceiling, wall, floor and any one of these surfaces.
Fu	Fuchs' correction factor to the Brownian agglomeration rate.
Fu_1, Fu_2	The terms due to Fuchs, and Sitarski and Seinfeld which appear in Fu.
g	The acceleration due to gravity.
$g(y)$	The basic finite element.
g_i, g_j, g_k	The i^{th}, j^{th} and k^{th} finite elements based on $g(y)$. e.g. $g_i(x) = g((x-x_i)/h)$.
h	The logarithmic spacing between successive collocation points.
i, j, k	Indices with values in the range 1 to n. See under subscripts below.
$\bar{j}$, $\bar{k}$	$\bar{j} = i - j$ and similarly for $\bar{k}$.
k	Boltzmann's constant.

k_a, k_b, k_q	Dimensionless constants in the Cunningham correction factor, Cu.
$K(\mu,m,t)$, K_{jk}	The agglomeration kernel. K_{jk} is $K(m_j,m_k,t)$.
Kn	The particle Knudsen number based on the radius of the equivalent spherical particle.
l	The mean free path of gas molecules in the bulk gas.
m, m_i, m_j, m_k	The particle mass in general and at the i^{th}, j^{th} and k^{th} collocation points respectively.
$m_{5\emptyset}$	The mass median mass of the airborne and source distributions.
m_g	The geometric mean mass of the airborne and source number distributions.
n	The number of collocation points.
n_{jk}	The normalization factor appearing in the production term of the discretized aerosol equation.
N	The number density of the airborne distribution.
$\frac{dN}{dt}$	The number release rate of source particles per unit cell volume.
p	The "wetted" perimeter of the cell perpendicular to the flow field.
P	The mean pressure in the cell.
P^i_{jk}	The integral which appears in the production term of the discretized aerosol equation.
P_r	A reference pressure which appears in the correlation for the diffusivity of steam in air.
r, r_i, r_j, r_k	The radius of the equivalent spherical particle in general and at the i^{th}, j^{th} and k^{th} collocation points respectively.
$r_{5\emptyset}$	The mass median particle radius of the airborne and source distributions.
r, s	Indices with values in the range 1 to n.
R	The universal gas constant.
R(m,t), R_i, R_j	The removal rate of particles of mass m per unit cell volume. R_i is $R(m_i,t)$ and similarly for j.
Re	The Reynolds number based on the hydrodynamic diameter and the mean flow speed.
S(m,t), S_i, S_j	The source number distribution. S_i is $S(m_i,t)$ and similarly for j.
Sc	The particle Schmidt number.
t	Time.
T	The mean gas temperature in the cell.
T_c, T_w, T_f, T_s	The mean gas temperature adjacent to, respectively, the ceiling, wall, floor and any one of these surfaces.

T_{r1}, T_{r2}	Reference temperatures appearing in correlations for the dynamic viscosity of air and the diffusivity of steam in air respectively.
U	The mean flow speed in the cell.
u_*	The friction velocity.
u_G	The particle gravitational terminal velocity.
v_*	The deposition velocity due to turbulence.
$\tilde{v}_*$, $\tilde{v}_{*1}$, $\tilde{v}_{*2}$	Non-dimensional turbulent deposition velocities appearing in the correlation based on the Lui and Agarwal data.
v_c, v_w, v_f, v_s	Net deposition velocities to, respectively, the ceiling, floor, wall and any one of these surfaces.
v_B	The deposition velocity due to Brownian diffusion across the laminar sublayer in a turbul nt flow.
v_{Dc}, v_{Dw}, v_{Df}, v_{Ds}	Diffusiophoresis deposition velocities to, respectively, the ceiling, wall, floor and any one of these surfaces.
v_G	The gravitational deposition velocity.
v_{Tc}, v_{Tw}, v_{Tf}, v_{Ts}	Thermophoresis deposition velocities to, respectively, the ceiling, wall, floor and any one of these surfaces.
V	The cell volume.
W_g, W_v	The average molecular weight of the gas in the bulk of the cell and the molecular weight of the vapor promoting diffusiophoresis.
x, x_i	x is $\log_e(m)$ and x_i is $\log_e(m_i)$.
y	The independent variable in the simplified integral for P^i_{jk} (Eq. (56)).
$Y(m,t)$, Y_i, Y_j, Y_k	The density distribution of the airborne aerosol. It is occasionally referred to as the mass distribution. Y_i is $Y(m_i,t)$ and similarly for j and k.
z_s	The equivalent sand roughness of surfaces in the cell.
α_g, α_p	The thermal conductivities of the gas and the particle material respectively.
χ_c, χ_d	The collision and dynamic shape factors.
χ_{Fu}, χ_s	The Fuchs collision efficiency and the particle-particle sticking efficiency.
δ_*	The viscous boundary layer thickness.
δ_{ij}	The Kronecker delta.
δ_D, δ_{Di}	The particle diffusion boundary layer thickness. δ_{Di} is δ_D evaluated for particles of mass m_i.
ϵ	The relative tolerance parameter.
ϵ_*	The average energy dissipation rate per unit mass due to turbulence in the bulk gas.
ϕ_{*I}, ϕ_{*S}, ϕ_B, ϕ_G	Agglomeration rates due to: particle inertia in a turbulent flow, turbulent shear, Brownian motion and gravitational settling respectively.

η	An absolute tolerance parameter required by the method for finding zeros of functions.
η_g	The dynamic viscosity of the bulk gas.
η_r	The dynamic viscosity of air at T_{r1}.
λ_l	The removal rate due to leakage.
λ_c, λ_w, λ_f, λ_s	The removal rates due to deposition to, respectively, the ceiling, wall, floor and any one of these surfaces.
λ_{ci}, λ_{wi}, λ_{fi}, λ_{si}	As above evaluated at m_i.
μ, ν	Particle masses.
π	Pi.
ρ	The density of the airborne aerosol.
$\frac{d\rho}{dt}$	The mass generation rate of the source per unit cell volume.
ρ_g, ρ_p	The density of the bulk gas and the particle material respectively.
σ	The cube root of the geometric standard deviation with respect to mass of the airborne and source number concentration distributions. It is also called the geometric standard deviation.
τ	The particle relaxation time.
$\bar{\tau}$	The dimensionless particle relaxation time.
ζ	A parameter required by the method for finding zeros of functions.

Subscripts

*	Denotes associated with turbulence.
c, w, f, s	Denotes, respectively, the ceiling, wall, floor and any one of these surfaces.
g, p	Denotes the bulk gas (except when used in m_g) and particles respectively.
i, j, k	Indices with values in the range 1 to n. They denote that the associated variable is to be evaluated at the corresponding collocation point.
r	Denotes a reference value. It is used for parameters appearing in correlations of physical properties.

APPENDIX F - INDEX OF SUBROUTINE AND FILE NAMES

[d] subroutine description

[l] subroutine listing

U.S. Government Printing Office
Receiving Branch (Attn: NRC Stock)
8610 Cherry Lane
Laurel, MD 20707
200 copies for R7

U.S. Nuclear Regulatory Commission (23)
Office of Nuclear Regulatory Research
Division of Reactor System Safety
Washington, DC 20555
Attn: D. Ross
M. Silberberg
G. Marino
C. Kelber
J. Mitchell
R. Meyer
L. Chan
J. Han (5)
P. Worthington
R. Wright (5)
R. VanHouten
B. Burson
T. Walker
P. Wood
T. Lee

U.S. Nuclear Regulatory Commission
Office of Nuclear Reactor Regulation
Washington, DC 20555
Attn: R. Palla

U.S. Department of Energy
Office of Nuclear Safety Coordination
Washington, DC 20545
Attn: R. W. Barber

U.S. Department of Energy (2)
Albuquerque Operations Office
P. O. Box 5400
Albuquerque, NM 87185
Attn: J. R. Roeder, Director
Transportation Safeguards Division
J. A. Morley, Director
Energy Research Technology Division

Argonne National Laboratory
9700 South Cass Avenue
Argonne, IL 60439
Attn: W. Sha

Battelle Columbus Laboratory (2)
505 King Avenue
Columbus, OH 43201
Attn: R. Denning
P. Cybulskis

Battelle Northwest (2)
P. O. Box 999
Richland, WA 99352
Attn: F. Panisko
D. Lanning

Brookhaven National Laboratory (3)
Upton, NY 11973
Attn: R. A. Bari
T. Ginsberg
M. Khatib-Rhabar

Electric Power Research Institute
P. O. Box 10412
Palo Alto, CA 94303
Attn: R. Sehgal (2)

Los Alamos National Laboratory (9)
P. O. Box 1663
Los Alamos, NM 87545
Attn: J. Dearing (5)
E. Fugelso
D. Liles
M. Sahota
J. Spore

Oak Ridge National Laboratory (2)
P. O. Box Y
Oak Ridge, TN 37830
Attn: T. Kress
S. Hodge
T. Nakamura

EG&G Idaho (4)
P. O. Box 1625
Idaho Falls, ID 83415
Attn: C. Allison
P. Bayless
J. Dahlman
D. Hagrman

University of Wisconsin
Nuclear Engineering Department
1500 Johnson Drive
Madison, WI 53706
Attn: M. L. Corradini

H. Bairiot
Belgonucleaire S. A.
Rue de Champ de Mars 25
B-1050 Brussels
BELGIUM

D. Haas
Belgonucleaire S. A.
Rue de Champ de Mars 25
B-1050 Brussels
BELGIUM

Atomic Energy Canada Ltd. (4)
Chalk River, Ontario
CANADA K0J 1J0
Attn: P. Fehrenbach
Wren
M. Notely
D. Nishimura

H. Rosinger
Atomic Energy Canada Ltd.
Pinawa, Manitoba
CANADA ROE 1LO

J. P. Longworth
Central Electric Generating Board
Berkeley Nuclear Labs
Berkeley, Gloucester GL13 9PB
ENGLAND

UK Atomic Energy Authority (6)
SRD
Culcheth, Warrington WA3 4VE
ENGLAND
Attn: M. Hayns
C. J. Wheatley (5)

UK Atomic Energy Authority (6)
NEEW
Winfrith, Dorchester
Dorset DT2 8DH
ENGLAND
Attn: R. Potter
S. Kinnersley
J. Lillington
D. Williams
N. Johns
N. Chown

B. Turland
UK Atomic Energy Authority
Culham Laboratory
Oxon 0X14 30B
Abingdon
ENGLAND

UK Atomic Energy Authority (6)
SRD
Culcheth, Warrington WA3 4VE
ENGLAND
Attn: J. Stephenson
F. Allen
P. Clough
I. Dunbar
S. Hall
F. Abbey

Kernforschungszentrum (4)
Postfach 3640
7500 Karlsruhe
FEDERAL REPUBLIC OF GERMANY
Attn: A. Fiege
S. Hagen
P. Hoffman
H. Rininsland

G. Petrangeli, Dire R. D. S.
Enea Nucl Energ Alt Disp
Via V Brancati 48
00144 Rome
ITALY

G. Saponaro
Enea Nucl Energ Alt Disp
Via Regina Marg-125
00144 Rome
ITALY

Japan Atomic Energy Res. Inst. (3)
Tokai-Mura Naka-Gun
Ibaraki-Ken 319-11
JAPAN
Attn: K. Hirano
S. Saito
K. Soda

J. H. Cha
Korea Adv. Energy Research Inst.
P. O. Box 7
Cheung-Ryang
Seoul
SOUTH KOREA

Korea Adv. Energy Research Inst. (2)
P. O. Box 7
Daeduk Danji
Choongnam
SOUTH KOREA
Attn: S. K. Chae
H. R. Jun

K. J. Brinkmann
Netherlands Energy Res. Fdtn.
P. O. Box 1
1/55ZG Petten NH
NETHERLANDS

Chao-Chin Tung
Atomic Energy Council
67 Lane 144 Sec 4
Keelung Rd.
Taipei, Taiwan
REPUBLIC OF CHINA

Sen-I Chang
Institute of Nuclear Energy Res.
P. O. Box 3
Lungtan
Taiwan 325
REPUBLIC OF CHINA

K. Johansson
Studsvik Energiteknik AB
S-611 82 Nykoping
SWEDEN

J. Bagues
Consejo de Seguridad Nuclear
Sor Angela de la Cruz 3
Madrid 28046
SPAIN

C. Graeslund
Statens Karnkraftinspektion
P. O. Box 27106
S-10252 Stockholm
SWEDEN

Sandia Distribution:
3141 S. A. Landenberger (5)
3151 W. L. Garner
6400 D. J. McCloskey
6415 R. M. Cranwell
6418 J. E. Kelly
6420 J. V. Walker
6422 D. A. Powers
6422 J. E. Brockmann
6422 W. W. Tarbell
6422 N. Yamano
6423 K. O. Reil
6423 R. O. Gauntt
6423 B. W. Marshall, Jr.
6425 W. J. Camp (18)
6425 S. S. Dosanjh
6425 R. D. Gasser
6425 A. J. Grimley, III
6425 T. J. Heames
6425 A. W. Reed
6425 R. C. Schmidt
6425 R. C. Smith
6425 J. L. Torkins
6425 K. A. Williams
6425 M. F. Young
6427 M. Berman
6429 K. D. Bergeron
6429 F. Gelbard
6429 K. K. Murata
8524 P. W. Dean

NRC FORM 335 (2-84) NRCM 1107, 3201, 3202

U.S. NUCLEAR REGULATORY COMMISSION

BIBLIOGRAPHIC DATA SHEET

SEE INSTRUCTIONS ON THE REVERSE

1. REPORT NUMBER (Assigned by TIDC, add Vol. No., if any)

NUREG/CR-5162
SAND88-0745

2. TITLE AND SUBTITLE

CHARM: A MODEL FOR AEROSOL BEHAVIOR IN TIME VARYING THERMAL-HYDRAULIC CONDITIONS

3. LEAVE BLANK

4. DATE REPORT COMPLETED

MONTH: April — YEAR: 1988

5. AUTHOR(S)

C. J. Wheatley

6. DATE REPORT ISSUED

MONTH: August — YEAR: 1988

7. PERFORMING ORGANIZATION NAME AND MAILING ADDRESS (Include Zip Code)

Sandia National Laboratories
P. O. Box 5800
Albuquerque, NM 87185

8. PROJECT/TASK/WORK UNIT NUMBER

9. FIN OR GRANT NUMBER

A1342

10. SPONSORING ORGANIZATION NAME AND MAILING ADDRESS (Include Zip Code)

Division of Systems Research
Office of Nuclear Regulatory Research
U.S. Nuclear Regulatory Commission
Washington, DC 20555

11a. TYPE OF REPORT

b. PERIOD COVERED (Inclusive dates)

12. SUPPLEMENTARY NOTES

*This work was done while the author was a UKAEA attache at Sandia National Laboratories.

13. ABSTRACT (200 words or less)

CHARM is a computer model for the behavior of a one component aerosol in a single region with time-varying external conditions. It treats particle agglomeration due to Brownian motion, gravity and turbulence, and particle deposition due to Brownian motion, gravity, turbulence, thermophoresis and diffusiophoresis. Turbulence properties are estimated for flow through a region of arbitrary cross-sectional shape, with aerodynamically rough or smooth walls at any Reynolds number. The gas can be of any composition. The time-varying external conditions allowed for are: the temperature, pressure and velocity of the gas, wall temperatures, and the rate, mass median radius and geometric standard deviation of the source. The model is simple, modified to enable this list to be extended if needed. A new method of solving the governing equations, based on the finite element collocation method, enables the time-varying conditions to be treated accurately and economically. We describe in detail: the models, the numerical methods, the execution of the computer code (including how to write the input data file and interpret results), and how to make simple modifications to the model. We discuss how the model could be implemented as a sub-model of a larger one and what further work needs to be done to enable it to efficiently treat multicomponent aerosols, and condensation onto and evaporation from particles.

14. DOCUMENT ANALYSIS - a. KEYWORDS/DESCRIPTORS

CHARM, aerosol computer model, aerosol coupling to thermal-hydraulics, finite-element method, collocation, CHARM user's manual

b. IDENTIFIERS/OPEN ENDED TERMS

15. AVAILABILITY STATEMENT

Unlimited

16. SECURITY CLASSIFICATION

(This page) Unclassified

(This report) Unclassified

17. NUMBER OF PAGES

18. PRICE

UNITED STATES
NUCLEAR REGULATORY COMMISSION
WASHINGTON, D.C. 20555

OFFICIAL BUSINESS
PENALTY FOR PRIVATE USE, $300

SPECIAL FOURTH-CLASS RATE
POSTAGE & FEES PAID
USNRC
PERMIT No. G-67

120555139217 1 1AN1R3
US NRC-OARM-ADM
DIV FOIA & PUBLICATIONS SVCS
RRES-PDR NUREG
P-210
WASHINGTON DC 20555